

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Emulace činnosti pozemních stanic s pasivní lokalizací polohy letadel
Bc. David Mlada

Diplomová práce
2019

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2017/2018

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David Mlada**
Osobní číslo: **I16234**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Emulace činnosti pozemních stanic s pasivní lokalizací polohy letadel**
Zadávající katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem diplomové práce bude návrh a realizace tří typů emulátorů pozemních stanic. Požaduje se, aby jako zdroj informací o pohybu leteckých cílů a vysílání ADS-B zpráv transpondérem z palub letadel, byl použit emulátor letadlového odpovídače SSR. První stanice bude emulovat chování přijímače ADS-B se směrovými anténami. Stanice bude zpracovávat zprávy ADS-B od simulovaných leteckých cílů. Zprávy budou filtrovány podle směrových charakteristik antén a pravděpodobnosti příjmu. Druhá stanice bude emulovat příjem signálu se zprávami ADS-B směrovým zaměřovačem. Měřené hodnoty azimutu a elevace leteckých cílů budou opatřeny chybami měření. Rozsah a rozložení chyb musí odpovídat reálným hodnotám skutečných zaměřovačů. Třetím emulátorem bude pasivní multilaterační systém. Tento emulátor nebude řešit multilaterační princip, ale bude pouze vydávat polohy leteckých cílů, která budou zatížena chybou. Poznámka: Tyto emulátory společně s emulátorem letadlového odpovídače SSR budou tvořit prostředek pro ověřování algoritmů validace ADS-B zpráv. Přičemž tyto algoritmy nejsou součástí tohoto zadání diplomové práce.

Rozsah grafických prací:

Rozsah pracovní zprávy: **minimálně 60 stran**

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

ARLOW, Jim a NEUSTADT, Ila. 2007. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. [překl.] Bogdan Kiszka. Brno : Computer Press, 2007. str. 567. ISBN 978-80-251-1503-9.
BEZOUŠEK, Pavel a ŠEDIVÝ, Pavel. 2004. Radarová technika. Praha : Vydavatelství ČVUT, 2004. ISBN 978-80-01-03036-3.

Vedoucí diplomové práce:

Ing. Karel Šimerda

Katedra softwarových technologií

Datum zadání diplomové práce: **30. října 2017**

Termín odevzdání diplomové práce: **18. května 2018**



Ing. Zdeněk Němec, Ph.D.
děkan



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2017

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 23. 8. 2019

Bc. David Mlada

PODĚKOVÁNÍ

Tímto bych chtěl velmi poděkovat panu Ing. Karlu Šimerdovi za jeho čas, který mi věnoval. Za jeho pomoc, připomínky a rady. Dále bych rád poděkoval své rodině, která mě celou dobu studia velmi podporovala a vždy se mi snažila co nejvíce pomoci.

ANOTACE

Cílem této diplomové práce je návrh a realizace tří typů emulátoru pozemních stanic. Využije se zdroj informací v ADS-B zprávě. Bude se emulovat stanice ADS-B se směrovými anténami. Dále se bude emulovat příjem signálu směrovým zaměřovačem. Třetím typem emulátoru bude pasivní multilaterační systém. Bude řešit umělou chybu daných hodnot.

KLÍČOVÁ SLOVA

emulátor, pozemní stanice, ADS-B, multilaterace, radiolokace, lokátor

TITLE

Emulation of ground stations with passive location of aircraft position.

ANNOTATION

The aim of this thesis is to design and implement three types of ground station emulator. The source of information in the ADS-B message is used. An ADS-B station with directional antennas will be emulated. Next, signal reception will be emulated by the direction finder. The third type of emulator will be a passive multilateral system. It will solve an artificial error of given values.

KEYWORDS

Emulator, ground station, ADS-B, multilateration, radiolocation, locator

OBSAH

Seznam obrázků	10
Seznam tabulek	12
Seznam zkratek	13
Úvod	14
1 Radiolokace	15
1.1 Metody snímání prostoru	15
1.2 Metody měření vzdálenosti.....	16
1.3 Metody určování úhlových souřadnic.....	17
1.4 Minimální dosah radaru	18
1.5 Maximální dosah radaru	18
1.6 Aktivní radar	19
1.6.1 Chyba náhodného měření	21
1.6.2 Systematické chyby měření	22
1.6.3 Přesnost měření úhlu.....	23
1.7 ADS-B přijímač	23
1.7.1 Princip.....	23
1.7.2 Bezpečnost.....	24
1.8 Pasivní radar	24
1.8.1 Princip.....	25
1.8.2 Postup zpracování	25
1.8.3 Porovnání	26
1.9 Multilaterační systém.....	26
1.10 Některé další radary v letectví	28
1.10.1 Meteorologické radary	28
1.10.2 Letecký palubní povětrnostní radar	28
2 Primární radar	29
2.1 Měření souřadnic cíle.....	29
2.2 Vzdálenost objektu	30

2.3	Režimy primárního radaru	32
3	Sekundární radar.....	33
3.1	Měření souřadnic cíle.....	34
3.2	Radiální vzdálenost.....	34
3.3	Azimut cíle.....	34
3.4	Výška cíle	35
4	Chybovost	37
4.1	Uměle vytvořené chyby	37
4.2	Typy chyb	38
4.2.1	Instrumentální omezení.....	38
4.2.2	Systematické chyby	38
4.2.3	Náhodné chyby	38
4.3	Přesnost a správnost.....	38
4.3.1	Malá přesnost, malá správnost.....	39
4.3.2	Malá přesnost, velká správnost.....	39
4.3.3	Velká přesnost, malá správnost	39
4.3.4	Velká přesnost, velká správnost.....	40
4.4	Markovův rozhodovací proces.....	40
4.5	Markovův řetězec	41
5	Umělé zatížení chybou	42
5.1	Výpadky.....	42
5.2	Korelovaná chyba pomocí Gauss-Markova procesu	42
5.3	Pevná chyba	43
6	Návrh.....	47
6.1	Funkční požadavky	47
6.2	Nefunkční požadavky	47
6.3	Analýza	48
6.4	Metamodel	48
6.5	Model	49

7	Implementace	51
7.1	Spring Framework	51
7.2	XML.....	52
7.3	Schéma.....	52
7.3.1	Zdroje.....	54
7.3.2	Pozice.....	55
7.3.3	Paprsky.....	56
7.3.4	Antény.....	56
7.3.5	Stanice.....	58
7.4	Aplikace	58
7.4.1	Balíček coordinates	59
7.4.2	Balíček jaxb	60
7.4.3	Balíček utils	60
7.4.4	Balíček locator	61
7.4.5	Balíček jama	61
7.4.6	Zkreslení chybou.....	61
7.4.7	Instalace aplikace	62
7.4.8	Spuštění aplikace	63
7.5	Vstupní data	63
7.6	Výstupní zprávy	65
7.6.1	ADS-B se směrovým zaměřovačem	66
7.6.2	ADS-B se směrovými anténami	67
7.6.3	Pasivní multilaterační systém	68
	Závěr	69
	Použitá literatura	70
	Přílohy	71

SEZNAM OBRÁZKŮ

Obrázek 1: Šroubovicové a řádkové snímání [9]	16
Obrázek 2: Určení polohy pomocí amplitudy odrazu [9]	17
Obrázek 3: Princip vysílání radaru [9]	21
Obrázek 4: Vliv šumu [1]	22
Obrázek 5: Systém multilaterace [2]	27
Obrázek 6: Schéma primárního radaru [10]	29
Obrázek 7: Sférické souřadnice [10]	30
Obrázek 8: Časová osa radaru [10]	31
Obrázek 9: Schéma radaru se spojitým provozem [10]	32
Obrázek 10: SSR princip [10]	33
Obrázek 11: Hlavní a potlačovací anténa [10]	35
Obrázek 12: Malá přesnost, malá správnost [7]	39
Obrázek 13: Malá přesnost, velká správnost [7]	39
Obrázek 14: Velká přesnost, malá správnost [7]	39
Obrázek 15: Velká přesnost, velká správnost [7]	40
Obrázek 16: Stavový diagram řetězce [6]	41
Obrázek 17: Korelovaná a nekorelovaná X souřadnice [8]	43
Obrázek 18: Korelovaná a nekorelovaná Y souřadnice [8]	43
Obrázek 19: Kružnicové označení vzdáleností od antény	44
Obrázek 20: Zdeformované označení vzdáleností od antény	45
Obrázek 21: Označení vzdáleností s rozdělením mřížkou	45
Obrázek 22: Metamodel návrhového modelu [11]	49
Obrázek 23: Model aplikace	50
Obrázek 24: Grafické schéma XSD	53
Obrázek 25: Zdroje aplikace	54
Obrázek 26: Frekvence a pozice	55
Obrázek 27: Beams	56
Obrázek 28: Antény	57
Obrázek 29: Lokátor	58
Obrázek 30: ARTAS transformace [6]	59
Obrázek 31: NLR transformace [6]	60
Obrázek 32: Aplikace SSR Transponder Emulator [12]	64

Obrázek 33: Výstup zdrojových dat [12]	64
Obrázek 34: Stanice ADS-B se směrovým zaměřovačem	66
Obrázek 35: Stanice ADS-B se směrovými anténami	67
Obrázek 36: Stanice s pasivním multilateračním systémem	68

SEZNAM TABULEK

Tabulka 1: Funkční požadavky	47
Tabulka 2: Nefunkční požadavky	48

SEZNAM ZKRATEK

CFAR	Constant False Alarm Rate
DAB	Digital Audio Broadcasting
DTD	Document Type Definition
DVB-T	Digital Video Broadcasting - Terrestrial
EPU	Estimated Position Uncertainty
HOFM	Horizontal Figure of Merit
HVE	Horizontal Velocity Error
GSM	Global System for Mobile
JSON	JavaScript Object Notation
LTE	Long Term Evolution
NM	Nautical Mile
ORM	Object Relation Mapping
POJO	Plain Old Java Object
XML	Extensible Markup Language
XSD	XML Schema Definition
YAML	Ain't Markup Language

ÚVOD

Tato diplomová práce se zabývá návrhem a následně realizací emulátoru pozemních stanic, které mají důležitou roli u radiolokace v letecké dopravě. Výstup této práce společně s dalšími projekty bude tvořit jeden velký projekt na odhalování falešných cílů v letecké dopravě. Tedy bude sloužit pro vývoj daných algoritmů bez potřebného hardwaru.

Nejprve je nutné si přiblížit informace o radarech a principech jejich chování. Budou představeny jednotlivé možnosti radiolokace. Rozdíly mezi jednotlivými typy stanic. Popis daných vlastností. Využití v řízení letového provozu. Nedostatky a chyby měření. Jedná se o obecné informace k radarům.

Následovat budou informace o aktivním a pasivním radaru. Popis jednotlivých vlastností s odkazem na dříve vysvětlené principy. Rozdíl mezi danými radary a využití v praxi. Zavedou se do praxe obecné informace fungování.

Dále je nutné specifikovat co je to chyba. Její vlastnosti a kdy vzniká a za jakých okolností. Popíší se jednotlivé typy chyb. Proběhne definice přesnosti a správnosti. Následovat taktéž budou informace o tom, jak uměle vytvářet chybu. Tak aby nebyla příliš akademická a odpovídala co nejvíce reálnému světu. To je velice důležité pro další využití této práce a akademický přínos.

V druhé části práce proběhne návrh a následně i implementace a realizace jednoduchého emulátoru, který půjde nakonfigurovat na jednotlivé typy. Možnosti konfigurace budou veliké. Bude se jednat o konzolovou aplikaci. Grafické prostředí není pro využití nutné a zbytečně by celou aplikaci zkomplikovalo s minimálním přínosem.

Společně s emulátorem letadlového odpovídače SSR bude tato aplikace tvořit prostředek pro ověřování algoritmů ADS-B zpráv.

Pro lepší čitelnost textu jsou v celé práci dodrženy jednotné konvence textu. Důležité pojmy jsou vyznačeny **tučně** a zdrojový kód je psán **neproporcionálním písmem**. Veškeré použité zkratky jsou vysvětleny v seznamu zkratek na začátku práce. Názvy tříd a atributů jsou *zkosené*.

1 RADIOLOKACE

Základy radiolokace byly určeny již ve druhé polovině 19. století, použití v praxi se ovšem radar dočkal až ve 2. světové válce. Začal být využíván jako varování před napadením nepřátelskými letouny. První radiolokátor byl sestrojen v Anglii roku 1939. Kvůli vysoké ceně bylo jeho použití do 70. let 19. století omezeno na velmi náročné aplikace. V nynější době se radary používají v široké škále odvětví a jsou významným prvkem letecké dopravy.

Radiolokace je činnost, při které je využíván radar pro zjišťování polohy objektů. Ve službách řízení letového provozu jsou dané objekty letadla. Radiolokací se tedy myslí zjišťování objektů v prostoru a určování jejich parametrů pohybu.

Podle způsobu, jakým radar plní funkci, rozlišujeme několik druhů: aktivní, aktivní s odpovídačem a pasivní. V případě aktivní radiolokace vyzařuje vysílač elektromagnetickou energii na objekty a přijímač zachycuje odražený signál od objektu. Sekundární radar pracuje s aktivním odpovídačem, který je umístěn na daném objektu. Díky tomu můžeme zjistit další informace o objektu, podle typu odpovědi.

Pasivní systémy byly vojenskou záležitostí, avšak v poslední době úspěšně pronikají do civilního letectví. Toto zařízení pouze přijímá elektromagnetickou energii vyzařovanou letadly.

Pro řízení letového provozu se nejčastěji používají přehledové radary. Ty snímají okolní prostor kruhovým pohybem antény ve směru hodinových ručiček. Přehledové radary při použití se liší podle role, na kterou jsou určeny. Důležitým faktorem je potom dosah maximální vzdálenosti, na kterou jsou schopny detekovat objekty a rychlost obnovení informace. Ta je dána rychlostí otáčení antény. Takto se dají radary rozdělit na traťové a okřskové. Dalším typem je radar pro detekci pohybů na ploše letiště. Přibližovací radary se dnes již téměř v civilním letectví nepoužívají. [9]

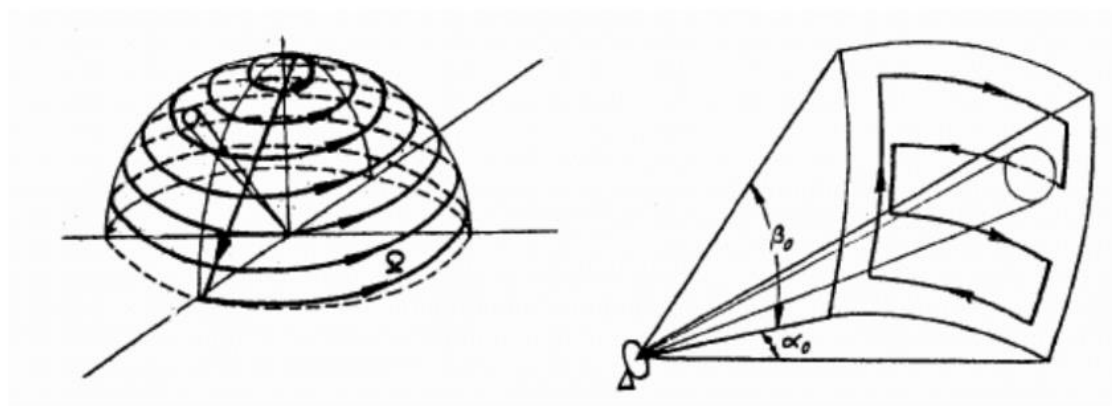
1.1 Metody snímání prostoru

Moderní radary mají velice úzký tok vyzařovací charakteristiky. Proto je nutné, aby anténa prohledávala prostor krytí. To se realizuje mechanickým vychylováním a natáčením antény. Další způsob je elektronické snímání, kdy je anténní systém nepohyblivý a svazek se vychyluje pomocí fázování signálů do jednotlivých dílčích zářičů. Snímání prostoru lze rozdělit na jednoduché a složené.

Při jednoduchém snímání provádí anténa pohyb kolem jedné osy. Takový systém zjišťuje jednu nebo dvě souřadnice většinou šikmou vzdálenost a azimut. Nejběžnější typ jednoduchého snímání prostoru je přehledové snímání, kdy se anténa otáčí v horizontální rovině o 360°.

Počet otáček přehledového radaru se pohybuje od 4 do 30 otáček za minutu. Rychlejší otáčení se využívá pro letištní pojezdové radary. Další možností je kuželové snímání, kdy hlavní anténní paraboloid je v klidu a pohybuje se pouze jen zářič.

U složeného snímání provádí anténa i vyzařovací diagram pohyb okolo dvou až tří os. To umožňuje zjištění více informací o cíli: šikmá vzdálenost, azimut, elevace a výška. Řadíme sem snímání spirálové, šroubovicové, pilovité a někdy i snímání, které kombinuje několik uvedených typů snímání dohromady. Šroubovicové snímání je způsobení kruhovým pohybem antény v azimutu a pomalým natáčením antény v elevačním úhlu. [9]



Obrázek 1: Šroubovicové a řádkové snímání [9]

1.2 Metody měření vzdálenosti

Základ všech metod pro měření vzdálenosti je v principu stejný. Jde o přesné určení časového rozdílu mezi odeslaným impulsem a přijatým odrazem. Z toho rozdílu pak lze vypočítat vzdálenost. Tedy pokud víme, jak signál rychle putuje, můžeme úplně jednoduše určit i vzdálenost pomocí upraveného vzorečku pro výpočet rychlosti.

$$v = \frac{s}{t}$$

Čili rychlost se rovná podílu dráhy a času. Po úpravě tedy dostaneme, že pro výpočet dráhy musíme vynásobit čas a rychlost.

$$s = v * t$$

Rychlost musí být v kilometrech za hodinu, nebo v metrech za sekundu. Čas potom v odpovídajících jednotkách.

Nesmíme zapomenout použít polovinu času, jelikož signál musel cestovat tam a zpátky.

1.3 Metody určování úhlových souřadnic

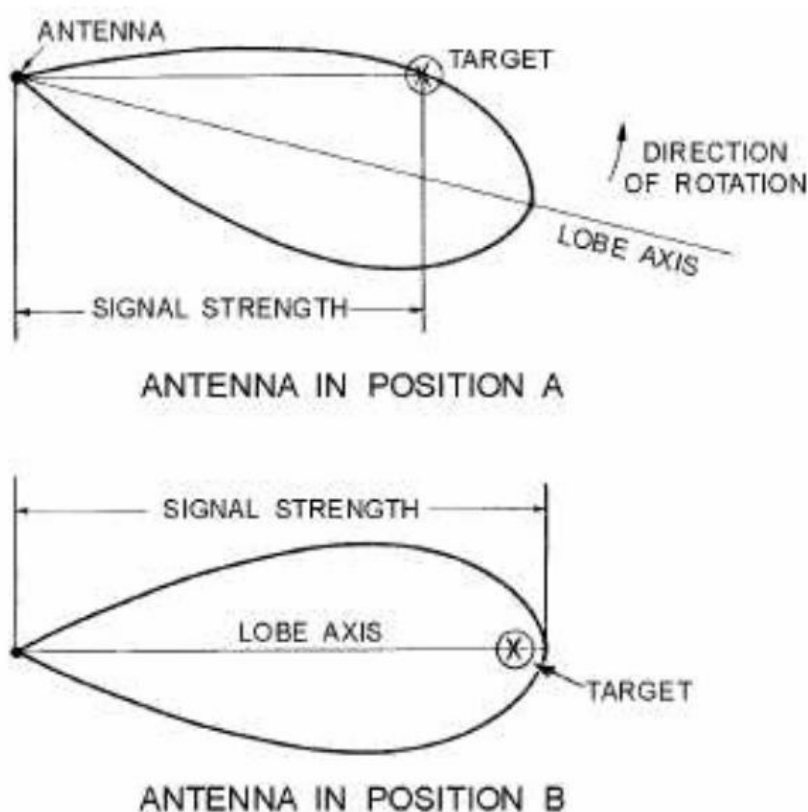
Toto měření využívá principu přímočarosti šíření elektromagnetických vln a vlastností antén. V praxi a reálných podmínkách bývá trajektorie zakřivená z důvodu nehomogenity prostředí. To nám způsobí hlavně nepřesnost při určení elevačního úhlu. Přesnost při měření v horizontální rovině je potom ovlivněna jen nepatrně.

Další příčinou je to, že reálný objekt je tvořen větším množstvím bodových objektů. Záření takovým objektem odražené je nesférické a v závislosti na směru šíření se jeho výkonová hustota prudce mění. Tím je deformovaná směrová charakteristika antény a vznikají tak chyby při určování úhlových souřadnic, kterým se říká **úhlový šum**.

Metody, které se používají pro zjišťování úhlových informací o objektech se standardně označují podle toho, který parametr přijatého signálu je dále použit pro vyhodnocení informací.

Na základě toho lze metody rozdělit na:

- Amplitudové,
- Fázové,
- Amplitudovo-fázové.



Obrázek 2: Určení polohy pomocí amplitudy odrazu [9]

1.4 Minimální dosah radaru

Radarová anténa slouží především k vysílání a zároveň i příjmu signálu. To je technicky možné díky **duplexeru**, který má na starosti přepínání funkce antény mezi příjmem a vysíláním. Je to tedy právě frekvence přepínání duplexeru, která je velmi kritická pro celý radarový systém. Ovlivňuje totiž právě minimální vzdálenost, na kterou je radarový systém schopen snímat objekt.

Musíme brát v úvahu dvě nejdůležitější hodnoty. Jsou to šířka pulsu a doba obnovy. Při vysílání impulsu je zapojen vysílač. Po vyslání daného pulsu, je anténa duplexem přepojena na přijímač. Přepnutí na vysílač se děje okamžitě. Přepnutí na přijímač už ovšem okamžité není a toto zpoždění nazýváme **doba obnovy**. Celkový čas, po který anténa nemůže přijímat signál, je tedy dán součtem času vysílání pulsu a onou dobou obnovy. Veškeré signály, které dorazí k přijímači dříve, než je přepnuto na příjem, nejsou detekovány a dochází ke ztrátě.

Z tohoto principu lze tedy odvodit výpočet minimální vzdálenosti. Například pro šířku pulsu 25 μ s a dobu obnovy 0,1 μ s je to:

$$((25 + 0,1))/2 * 984 \text{ (ft)} = 12.349,2 \text{ feet} = 2,03 \text{ NM}$$

Číslo 984 ft je vyzařovaná energie za mikrosekundu. Jednotka NM značí námořní míli. V dnešní době už má většina moderních systémů tak malou dobu obnovy, že může být zcela zanedbána ve výpočtu. [9]

1.5 Maximální dosah radaru

Maximální dosah impulsního radarového systému určuje hlavně nosná frekvence, maximální výkon pulsu, frekvence opakování pulsu a citlivost přijímače. Jelikož rádiové vlny s frekvencí nad 3 tisíce MHz jsou výrazně pohlcovány atmosférou, bývá nosný kmitočet vysílaný radarem nejčastěji limitujícím faktorem maximálního dosahu daného radaru. Jelikož při zvyšování frekvence nosné vlny musí dojít i ke zvýšení vyzářeného výkonu pro udržení stejného maximálního dosahu. Z tohoto důvodu radarové systémy velkého dosahu používají nižší frekvence než radary krátkého dosahu. Impulsní výkon ovlivňuje maximální dosah radaru právě z důvodu ztrát spojeným s průchodem atmosférou.

Radarové systémy vysílají každý puls během doby, kdy je anténa nastavená pro vysílání a poté čekají na návrat odraženého signálu po dobu zbývajících času určeného pro příjem. Frekvence opakování pulsu nám pak udává množství pulsů vyslaných během jedné vteřiny. Čas mezi začátkem jednoho pulsu a začátkem druhého pulsu se nazývá **čas opakování pulsu**.

Po vyslání pulsu je vždy resetován vnitřní čas radarového systému. To zajistí správné změření doby od vyslání signálu a přijetí odraženého signálu. Čas mezi začátkem prvního a druhého pulsu je důležitý při určení maximálního dosahu radaru. Jelikož signál, jehož doba návratu je větší než tento čas, radar následně vyhodnotí jako v nesprávné vzdálenosti. Odrazy se v takové nesprávné vzdálenosti označují jako **sporné odrazy**. Posledním prvek je citlivost přijímače, která ovlivňuje nejmenší možnost energie odraženého signálu, kterou je přijímač schopen zachytit a systém zpracovat. [9]

1.6 Aktivní radar

Základní princip je vcelku jednoduchý, ovšem teorie se může zdát být velmi komplexní. Zavedení a provoz systému zahrnuje velkou škálu oborů. Přes stavební práce, těžkou mechanickou a elektronickou výrobu přes mikrovlnné inženýrství s vysokým výkonem a dále pokročilé vysokorychlostní signály a techniky zpracování dat.

Měření rozsahu či vzdálenosti je možné díky vlastnostem vyzařované elektromagnetické energie. Elektromagnetické vlny se odrážejí, pokud narazí na elektricky vodivý povrch. Pokud se tyto odražené vlny znovu dostanou na místo jejich původu, pak to znamená, že překážka je ve směru šíření.

Elektromagnetická energie prochází vzduchem konstantní rychlostí přibližně rychlosti světla ve vakuu, což je přibližně necelých 300 000 kilometrů za sekundu. Tato rychlost umožňuje určení vzdálenosti mezi objekty (letadla, lodě, automobily a další) a radarovým stanovištěm měřením doby chodu přenášených impulzů. Energie za normálních podmínek cestuje prostorem v přímém směru a mění se jen nepatrně kvůli atmosférickým a klimatickým podmínkám. Použitím speciálních radarových antén může být tato energie zaměřena do požadovaného směru. Tak lze měřit například směr odražených objektů v azimutu a výšce.

Zaměřovač je schopen pracovat ve dne, v noci i v jakémkoliv světle. Je schopen pracovat v mlze či dešti a za všech okolností. Může dokonce proniknout i do stěn nebo vrstev sněhu. Má velmi široké pokrytí, je možné pozorovat celou hemisféru. Detekuje a sleduje pohyblivé objekty. Je možné zobrazení s vysokým rozlišením, což vede k rozpoznání objektu. Může pracovat bez posádky a bez přestávky.

Měření vzdálenosti funguje na principu krátkého impulsu s velmi vysokým výkonem. Tento impuls je zaměřen pouze jedním směrem na směrové anténě a šíří se v tomto směru rychlostí světla. Pokud je v tomto směru překážka, například letadlo, pak je část energie z impulsu rozptýlená ve všech směrech. Velmi malá část se odráží zpět do bodu vyslání. Anténa přijme tuto

energii a dochází k vyhodnocování informací. Vzhledem k tomu, že šíření vln probíhá v konstantních rychlostech, je tato vzdálenost určena z doby trvání mezi vysláním a následným přijmutím odraženého impulsu. Skutečný rozsah cíle z radaru se nazývá **šikmý rozsah**. Šikmá vzdálenost je přímka mezi radarem a objektem. Přičemž rozmezí země je horizontální vzdálenost mezi vysílačem a cílem. Výpočet vyžaduje znalost výšky cíle. Vzhledem k tomu, že impuls cestuje tam a zpět, doba cesty se dělí 2, aby se získal čas pouze na vzdálenost k objektu.

Následující vzorec tedy vypadá takto. Kde c je rychlost světla, symbol t znamená naměřený čas a R naměřená vzdálenost v kilometrech.

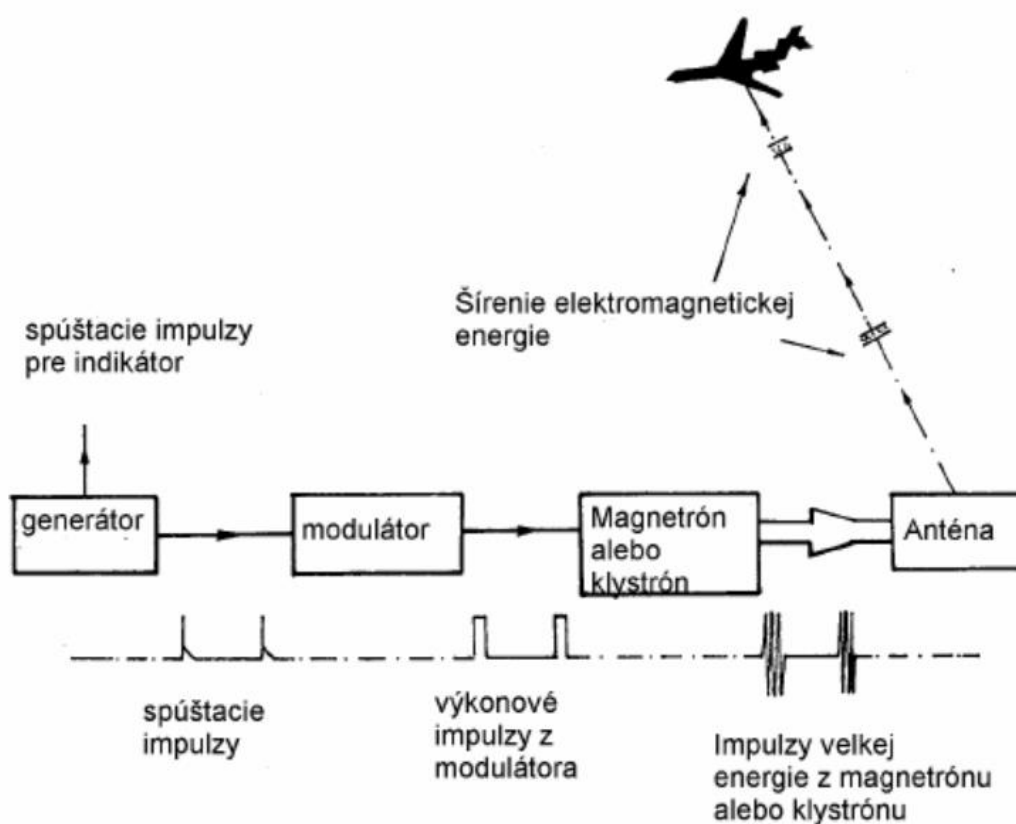
$$R = \frac{c * t}{2}$$

Přesnost je velikost shody mezi odhadovanou (měřenou) a její skutečnou polohou a rychlostí. Rádiová navigační přesnost je obvykle prezentována jako statistická míra systémové chyby a je určena jako:

- předvídatelná – přesnost polohy ve vztahu ke geografickým a geodetickým souřadnicím země,
- opakovatelná – přesnost, ve které se uživatel může vrátit do pozice, jejíž souřadnice byly v předchozím čase měřeny stejným systémem,
- relativní – přesnost, kterou uživatel může určit jednu pozici vůči druhé tím, že neznečistí všechny možné chyby.

Hodnota požadované přesnosti představuje nejistotu vykazované hodnoty s ohledem na skutečnou hodnotu a udává interval, ve kterém skutečná hodnota leží s uvedenou pravděpodobností. Nejčastější a doporučovaná pravděpodobnost je 95 %, která představuje cca 2 standardní odchylky průměru pro **normální (Gaussovu) distribuci** proměnné. Předpokládáme, že se bere v úvahu veškerá známa korekce. Znamená to tedy, že chyby v hodnotách, které vykazujeme, budou mít střední hodnotu či zkreslení téměř nulové. Jakékoliv zbytkové zkreslení by mělo být malé ve srovnání s požadovaným požadavkem na přesnost. Skutečná hodnota, která za provozních podmínek dokonale popisuje proměnnou, je hodnota, která se má měřit či pozorovat v požadovaném čase, oblasti a intervalu s přihlédnutím k umístění.

Teoretická největší přesnost, se kterou může být vzdálenost měřena, závisí na přesnosti měření doby chodu.

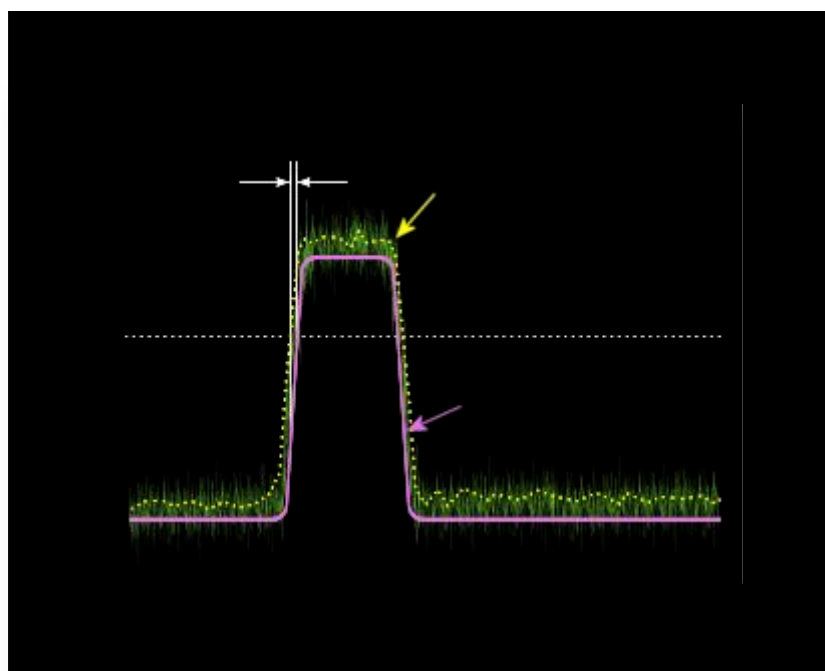


Obrázek 3: Princip vysílání radaru [9]

1.6.1 Chyba náhodného měření

Při vysílání impulsu může nastat a často nastává náhodná chyba, když je vzrůstající okraj odraženého signálu zkreslený šumem. Impuls je při měření vždy rušen šumem. K měření šumu a impulsu se používá **amplituda**, tudíž se posunuje okraj a tím způsobuje chybu v měření.

Následující obrázek ukazuje vliv šumu:



Obrázek 4: Vliv šumu [1]

Pevná růžová čára představuje téměř ideální lichoběžníkový impuls se strmými hranami. Impuls nemůže být zcela pravoúhlý, protože by to vyžadovalo nekonečnou šířku pásma. Čas se měří v místě určeném prahovou hodnotou, obvykle na úrovni 0,7V maximálního napětí. Impuls se překrývá s hladinou šumu, označeného zeleně. Může se tedy měřit pouze napětí, které je tvořené napětím impulsu a šumu, vyznačené žlutou tečkovanou čarou. Změřené napětí překračuje prahovou hodnotu dříve než impuls. Rozdíl je náhodná chyba měření způsobená šumem. Pokud známe trvání impulsu, může být chyba matematicky snížena současným vyhodnocením předního a zadního okraje impulsu.

1.6.2 Systematické chyby měření

Doba běhu se standardně měří od stoupající hrany vysílacího impulsu ke stoupajícímu okraji signálu ozvěny. Přesnost tohoto měření však závisí na velikosti frekvence hodin pro měření času. Výsledky měření mezi jednotlivými cykly není možné změřit a generují systematickou chybu měření. Přesnost tedy závisí na velikosti jednotlivých buněk rozsahu v procesech zpracování signálu.

1.6.3 Přesnost měření úhlu

Přesnost při měření úhlu závisí na interních způsobech zpracování dat, a na vnějších podmínkách. Podmínky šíření anomálie, která se často vyskytuje v důsledku změn tlaku vzduchu při měření výšky. V principu se mohou také vyskytovat v bočních úhlech a vytvářet náhodnou chybu.

Většinou je ale větší problém v interním zpracování dat. Určení úhlu posuvným oknem je poněkud nepřesný postup. Poloviční šířka antény je dělena pouze počtem pulsových period a výsledkem je systematická chyba řádově až o jeden stupeň. Korelační metody mohou také interpolovat mezilehlé hodnoty, a proto jsou mnohem přesnější. Nejvyšší přesnost dosáhne metoda kónického skenování. [1]

1.7 ADS-B přijímač

Automatic Dependent Surveillance-Broadcast systém se dělí na dvě základní skupiny. Na vysílací část **ADS-B OUT** a na přijímací část **ADS-B IN**. Jedná se o systém plně automatické obsluhy. Systém slouží převážně ke sledování pohybu a polohy. Stroj, nejčastěji letadlo, vybavené tímto systémem, v intervalech zjišťuje svoji polohu, například pomocí GPS a jiných globálních navigačních systémů. Tyto a další informace o sobě vysílá podle stanovených norem všemi směry. Pro vysílání tedy slouží ADS-B OUT.

1.7.1 Princip

Stroje vybavené ADS-B IN, tyto získané informace odchyť a zpracují. Nejčastěji používanými pásmy jsou 978MHz a 1090MHz. Kvůli velkému rádiovému provozu a využití pásma 1090MHz dojde brzy pravděpodobně k rozdělení do pásem. Rozdělení bude podle hranice 18 tisíc stop. Vzniknou dvě pásma: **LOW (978 UAT)** a **HIGH (1090 ES)**. Toto rozdělení bude příčinou následné úpravy starších systému, aby byly taktéž schopné pracovat v obou režimech a kmitočtech.

Pro příjem je ovšem potřeba určité vybavení. Hlavním prvek je anténa, která je schopná přijímat danou frekvenci. Při výběru antény se musí dbát na to, že vysílání probíhá s vertikální polarizací. Anténa se umístí co nejvýše nad okolní terén, což zajistí signál z co největší možné vzdálenosti.

Okolní blízké silné vysílače mohou příjem zahltit. Nejčastěji se setkáme s problémem GSM, DVB-T a nově i LTE. Na tento problém zatím neexistuje efektivní řešení. Jedno z možných řešení je ovšem použít **filtr**, který má danou pásmovou propustnost.

Přijímání informací probíhá v následujících krocích. Při přijmutí a dekodování zpráv se data ukládají do paměti a skládají se celé zprávy. Tato celá kompletní zpráva je potom předána dané aplikaci. [3]

1.7.2 Bezpečnost

Komunikace probíhá přes otevřený a známý protokol. To bohužel otevírá možnost útokům na měření jako například **spoofing attacks**, kdy útočník je schopen zablokovat nebo zfalšovat signál. To má za následek ztrátu přesnosti polohy a další neočekávané chování v řízení letecké dopravy. Útok může být pasivní nebo aktivní.

Pasivní útok je spíše příprava na aktivní útok. Jedná se o sběr dat, odposlouchávají se pravidelné zprávy, získávají se jedinečné identifikátory a trajektorie komunikujících letadel. Naopak aktivní útoky jsou prováděné za účelem narušení sledování letového provozu. Jedná se o chyby v aplikaci během přenosu, které mohou způsobit falešnou chybu. Spuštění příkazů, které vysílají nesprávné či falešné zprávy o umístění letadel. Během rušení mohou spustit systémy řízení letového provozu, když zaznamenají selhání ADS-B, záložní systémy s nižším výkonem a kapacitou. [1]

Tyto útoky mohou způsobit letecké nehody, konflikty v trasách způsobené vysíláním falešného umístění a záměru letadla. Přesnost sledování může být snížena a vznikají bezpečnostní obavy pro cestující. Proto se musí zprávy ADS-B verifikovat.

Na základě času vyslání zprávy, času užitečného zatížení dat a času přijetí zprávy lze teoreticky vypočítat rozsah do vysílače a ověřit rozsah do polohy ADS-B. Bohužel je tato metoda zranitelná, pokud někdo zfalšuje zprávu v souladu s falešným rozsahem času. [2]

1.8 Pasivní radar

Konvenční radarové systémy obsahují jednak přijímač a jednak i vysílač. Oba dva systémy obvykle vysílají a přijímají společnou anténou, což šetří náklady. Vyslaný pulzní signál a následná doba potřebná k tomu, aby se tento signál vrátil od odraženého objektu zpět do antény, odkud byl vyslaný, slouží k určení rozsahu určeného objektu. [1]

1.8.1 Princip

V pasivním radarovém systému ovšem žádný takový vysílač není. Namísto něj přijímač využívá vysílače třetích stran. Měří tedy časový rozdíl příchodu mezi signálem přicházejícím přímo z cizího vysílače a signálem přicházejícím prostřednictvím odrazu od objektu. Toto nám umožňuje stanovení takzvané **bistatického rozsahu** objektu.

Kromě tohoto bistatického rozsahu měří pasivní zaměřovač i Dopplerův posun ozvěny a její směr. Umožňuje to vypočítat polohu, směr a rychlost sledovaného objektu. U některých případů se využívá více vysílačů a přijímačů pro provedení více nezávislých měření. Termín pasivní radar je někdy nesprávně používán jako pro pasivní senzory, které detekují a sledují objekt pomocí RF emisí. Tyto systém ovšem nevyužívají odraženou energii vysílaného pulsu, a proto jsou přesnějším názvem popsány jako systémy **ESM**. Mezi nejznámější příklady patří české systémy TAMARA a VERA. [4]

1.8.2 Postup zpracování

V konvenčním radarovém systému jsou přesně známy všechny časy přenosu pulsu a dané přenášené křivky. To umožňuje jednoduše vypočítat rozsah objektů a využít přizpůsobený filtr pro dosažení optimálního poměru šumu k signálu v daném přijímači. Pasivní systém ovšem nemá tyto informace, a proto musí využít vyhrazený přijímací kanál, kterému se říká referenční kanál. Pro sledování a monitorování každého vysílače, který je využíván. Sleduje se přenášený průběh. Pasivní zaměřovač obvykle používá následující algoritmus zpracování:

- 1) Příjem přímého signálu z vysílačů a z oblasti vyhrazených lineárních, digitálních přijímačích.
- 2) Digitální zpracování a tvarování paprsku pro určení směru příchodu signálů a prostorové vyhodnocení odmítnutí silného rušení v daném pásmu.
- 3) Dynamické adaptivní filtrování pro vyrušení nežádoucích přímých zpětných signálů ve sledovaném pásmu.
- 4) Úprava signálů dle daných požadavků pro vysílač.
- 5) Křížová korelace mezi referenčním kanálem a sledovaným pásmem pro určení bistatického rozsahu a Dopplerova posunu.
- 6) Detekce pomocí schématu CFAR.
- 7) Sledování objektů v rozsahu a asociace trasy.
- 8) Sdružení a spojení traťových linek z každého dostupného vysílače, jenž tvoří konečný odhad umístění objektu, jeho směru a rychlosti. [5]

1.8.3 Porovnání

Pasivní systém je výkonnostně srovnatelný s konvenčními aktivními systémy krátkého a středního dosahu. Detekční rozsah lze určit pomocí standardní radarové rovnice, kromě zajištění řádného zesílení zpracování a omezení externího šumu. Na rozdíl od aktivního zaměřovače je rozsah dán také funkcí geometrického rozmístění. Jelikož vzdálenost přijímače od vysílače určuje úroveň vnějšího šumu, proti kterému musí být cíle detekovány. Je ovšem nutné očekávat, že pasivní systém, který využívá rádiové frekvence FM, dosáhne detekčních dosahů až 150 km. Pro vysoce výkonné analogové televizory a stanice až 300 km. Pro nižší napájení digitálních signálů, jako je například DAB či DVB-T až několik desítek kilometrů. [4, 5]

Pasivní přesnost radaru je silně ovlivněna geometrickou funkcí na základě počtu přijímačů a vysílačů, které se využívají. Dá se říci, že čím více je přijímačů a vysílačů, tím je větší přesnost. Systém, který využívá pouze jednoho vysílače a jednoho přijímače, je tedy mnohem méně přesný než systém pro běžné sledování aktivními radary. Zatímco při využití více přijímačů a vysílačů je schopen dosáhnout daleko větší přesnosti. Většina pasivních systému detekuje polohu pouze dvourozměrně, ovšem měření výšky je možné. [4, 5]

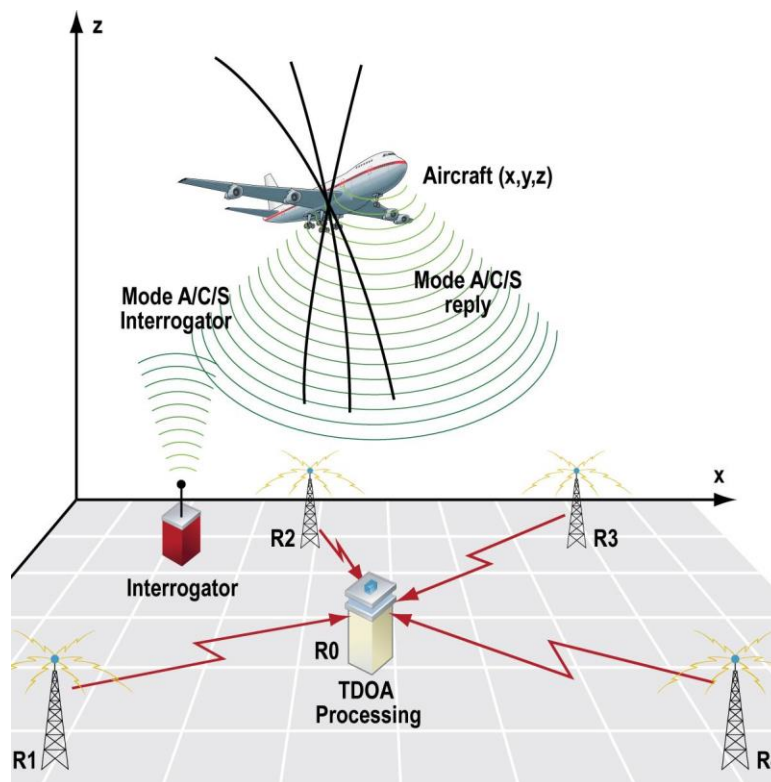
Výhody oproti aktivním systémům jsou například nižší pořizovací náklady či náklady na provoz a údržbu, a to v důsledku chybějícího vysílače a pohyblivých částí. Jedná se o zakrytou operaci, není nutné přidělovat frekvenci. Fyzicky jsou malé a snadno se rozmístí v místech, kde aktivní radary být nemohou. Poskytují velmi rychlé aktualizace, nejčastěji do vteřiny. Jsou odolné vůči protiradiačním raketám a celkově je obtížné jejich rušení. [4, 5]

Na druhou stranu se dá tato technologie považovat za nevyzrálou. Je zde evidentní závislost na dalších systémech. Nasazení je poněkud složité a poskytne nám jen dvourozměrný provoz. Pro trojrozměrný by se muselo využít dvou různých systémů. [4, 5]

1.9 Multilaterační systém

Pasivní sledovací systémy fungují na principu **multilaterace**. Přesnou polohu objektu se určí vyhodnocením rozdílu času příchodu signálu z cíle minimálně na tři přijímací stanice umístěné v určité potřebné vzdálenosti.

Pojem multilaterace je překládán jako hyperbolické polohování. Systém slouží k určování polohy v rovině či prostoru pomocí několika pozemních stanovišť. Celý systém však závisí na datech vysílaných objektem, nejčastěji ADS-B. [1]



Obrázek 5: Systém multilaterace [2]

Základní princip je obdobný principu fungování u aktivního radaru. Zachycuje data vysílané objektem (například letadlem). Na rozdíl od aktivního radaru však nevysílá požadavek na odpověď s údaji. Systém tedy pouze jenom čeká na signál. Celý systém je složen z X pozemních stanic. Pro určování polohy cílů v rovině stačí pouze tři stanice. Pro určování v prostoru už jsou zapotřebí čtyři stanice. Systém závisí na tvorbě hyperboloidů, kterých vzniká vždy $X-1$, kde X je počet pozemních stanic. Z toho vyplývá, že pro určení polohy v rovině nám stačí dva hyperboloidy (tři stanice) a pro určení polohy v prostoru tedy tři hyperboloidy (čtyři stanice). [3]

Přesnost určení pozice je závislá pouze na přesnosti přijímačů **MLAT**. Výška objektu na přesnost nemá vliv. Na druhou stranu při použití více přijímačů, než je potřebné minimum, zase nedochází ke zpřesnění polohy. Určování polohy není závislé na čase vysílání, ale pouze na časovém rozdílu přijímaného impulsu. [2]

Dochází zde k časové odchylce zpracovaného času příchodu signálu a skutečného času, kdy signál dorazil k přijímači. Z důvodů vícecestného šíření, nepřesného zachycení začátku signálu a nepřesné vzájemné synchronizaci času mezi stanicemi. Z těchto důvodů je nutné vytvořit směrodatnou odchylku a s tou počítat ve výpočtech. [4]

1.10 Některé další radary v letectví

1.10.1 Meteorologické radary

Radiolokátory sloužící ke zjišťování rozložení okamžitých intenzit atmosférických srážek a výskytu jevů spojených s oblačností. Jsou schopné pracovat do vzdáleností až 200 km. Funkce je založena na schopnosti vodních kapek, sněhových vloček a ledových krup odrážet vlny. Vysílač generuje krátké vysokoenergetické pulsy elektromagnetického vlnění, které anténa vyzařuje v podobě úzkého svazku do atmosféry.

1.10.2 Letecký palubní povětrnostní radar

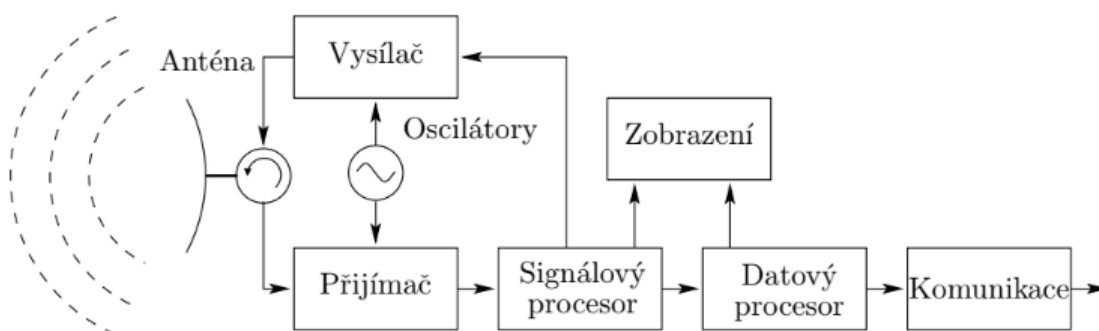
Detekce velikosti odrazů mraků a srážek. Dopplerovo měření pohybu povětrnostních částic uprostřed meteorologických situací. Spolehlivost určení palubního radaru představuje v současnosti velmi významnou a důležitou oblast zkoumání. Hlavní důvod je existence a krutost jevu známého jako **větrný stříh**.

2 PRIMÁRNÍ RADAR

Primární radar se řadí mezi aktivní radary. Využívá tedy odrazu vlastní elektromagnetické vlny. Elektromagnetická vlna se odráží od okolních předmětů a radar je přijímá a zpracovává.

Pokud se objekt, od kterého se vlna odrazila, pohybuje, dochází k posuvu frekvence odraženého signálu oproti frekvenci vyslaného signálu. Tento jev nazýváme **Dopplerův jev**.

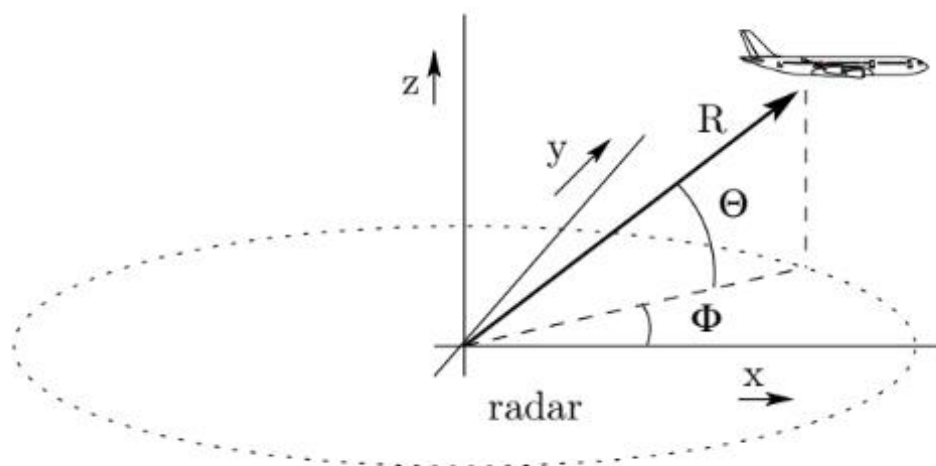
Schéma primárního radaru je zobrazeno níže. Nejprve se vygeneruje vysílaný signál, anténa vyšle elektromagnetické vlny a následně přijme odražené. Přijímač signál zesílí a vyfiltruje šum. Signálový procesor zpracuje naměřené signály a poskytne data o naměřených objektech a jednotlivé parametry, jako jsou poloha, rychlost a podobné, které se pak zobrazí nebo jsou dále využity.



Obrázek 6: Schéma primárního radaru [10]

2.1 Měření souřadnic cíle

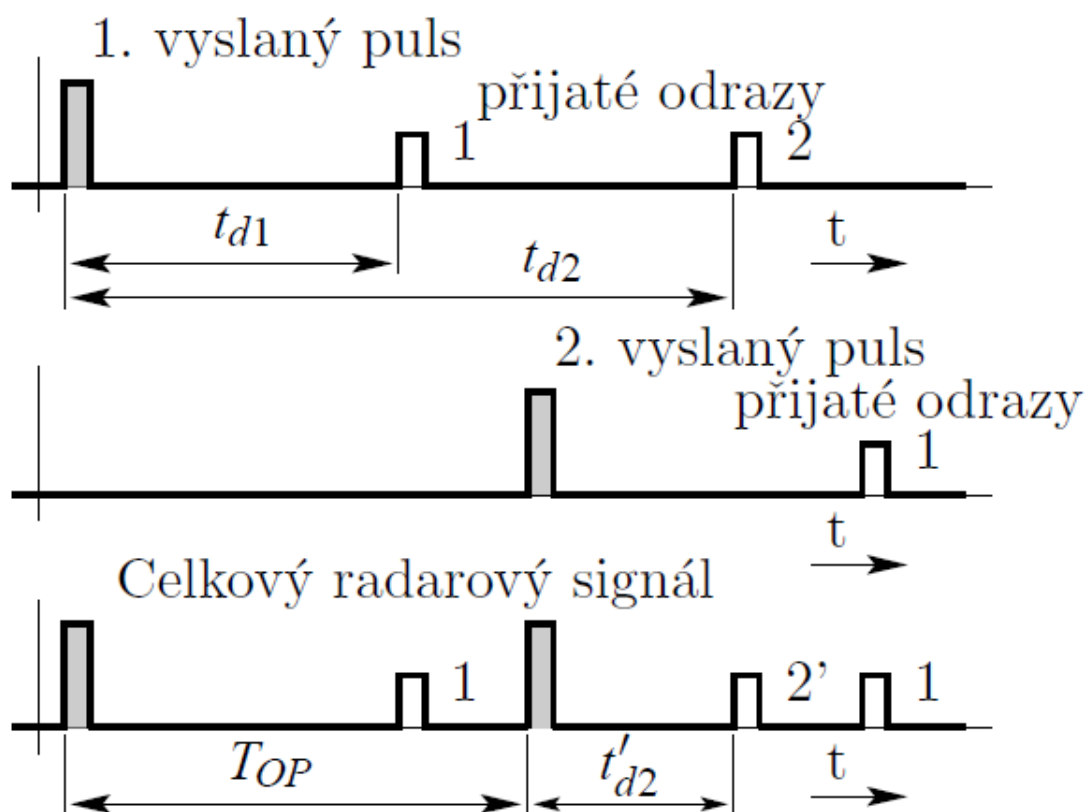
Polohu cíle určuje primární radar standardně ve sférických souřadnicích s počátkem v místě radaru. Jednotlivé souřadnice polohy se označují: šikmá vzdálenost R , úhel Φ , elevace Θ a někdy i radiální rychlost.



Obrázek 7: Sférické souřadnice [10]

2.2 Vzdálenost objektu

Jak již bylo zmíněno, vzdálenost se u primárních radarů určuje měřením rozdílu času mezi přijatým (odraženým) signálem a vyslaným signálem. Radary delšího dosahu, řádově stovky metrů, pracují v impulzním režimu. To znamená, že radar v pravidelných intervalech vysílá krátké elektromagnetické impulzy energie. Pak se přepne a jistou dobu nevysílá, ale pouze přijímá signály. Toto chování je zobrazeno na následujícím obrázku.



Obrázek 8: Časová osa radaru [10]

Jelikož poměr mezi výkonem vysílaného a výkonem přijímaného signálu u radarů, které mají větší dosah, je velmi veliký, kolem 200 dB. A kmitočty jsou velmi podobné až shodné, je velmi těžké oddělit od sebe současně pracující vysílač a přijímač. Z těchto technologických důvodů se tedy volí impulzní režim. Při impulzním provozu funguje vždy jen buď přijímač nebo vysílač. Z toho lze odvodit, že objekty, od nichž se vlna odrazí a stihne vrátit dříve, než skončí vysílání, nebudou radarem detekovány. Tuto minimální vzdálenost nazýváme jako **slepá zóna** a značíme ji R_b . [10]

$$R_b \geq \frac{c * \tau_p}{2}$$

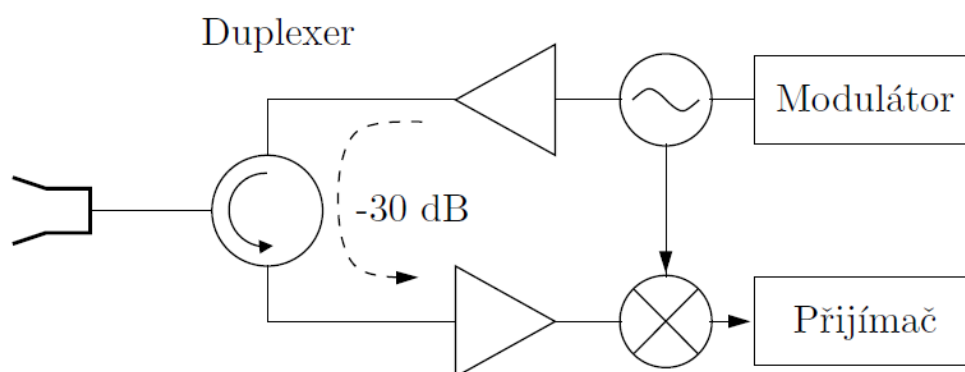
Znak c značí rychlost světla a τ_p označuje délku vysílaného signálu. Jelikož po skončení vysílání je potřeba ještě přepnout do režimu přijímače, a to zabere nějaký čas. Je zde znaménko nerovnosti. Bývá standardem, že tato slepá zóna je 0,1 % až 1 % z maximálního dosahu. Tedy poměr mezi délkou vysílaného signálu a dobou opakování je 1:100 až 1:1000. [10]

2.3 Režimy primárního radaru

Podle charakteristiky signálu vysílaného k cíli, lze rozdělit primární radary na spojitý a impulzní. Radary s impulzním režimem pravidelně střídají vysílání a přijímání odraženého signálu. Radary se spojitým režimem tyto činnosti dělají současně.

Impulzní radary umožňují provozem dosažení vyššího středního výkonu a spojitý radary zase mají jednodušší obvody pro zpracování. Jelikož pro radary se středním a velkým dosahem je zapotřebí vysoký střední výkon, je většina radarů v impulzním režimu. Na druhou stranu pro radary s krátkým dosahem lze využít radarů se spojitým režimem a jelikož jsou jednodušší, dává se jim přednost.

Rušení, které má za následek omezení dosahu, je způsobeno třemi na sobě nezávislými vlivy. Konverzí FM šumu oscilátoru radaru na šum přijímače. Přenosem AM šumu oscilátor do přijímače a šumem samotného přijímače.



Obrázek 9: Schéma radaru se spojitým provozem [10]

Aby byl přijímaný signál detekován, požaduje se minimální výkon 15 dB a část vysílaného výkonu, která projde k přijímači nesmí překročit hranici, se kterou je daný přijímač schopen pracovat.

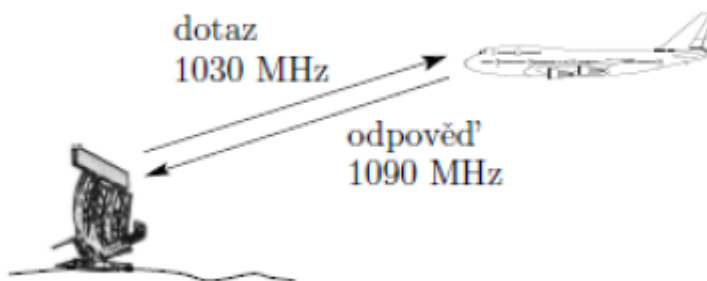
3 SEKUNDÁRNÍ RADAR

Sekundární radar je zařízení sloužící pro radiolokaci, které vysílá dotazy a zpracovává odpovědi zaslané odpovídači, které dané dotazy zachytili. Dokáže tedy komunikovat pouze s objekty vybavené nějakým aktivním odpovídačem. Tento aktivní odpovídač musí být schopen dotaz zachytit a správným způsobem na něj nějakým způsobem reagovat.

Celý systém je tedy složen ze dvou důležitých částí. Tou první je takzvaný dotazovač, který je umístěný na stanici. Druhou částí je odpovídač, který je umístěný na vzdáleném objektu. Sekundární radary se velmi často využívají spolu s primárními radary. Jejich anténa bývá umístěná nad anténou primárního radaru.

Velmi zásadní výhodou těchto radarů je fakt, že jsou daleko méně náročnější na energii. Výkonová bilance je tedy příznivější. Výkon nezávisí na odrazové ploše cíle. Velkou výhodou je pak možnost přenášet různé informace o sledovaném objektu včetně jeho unikátní identifikaci.

Celý systém pracuje na frekvenci 1 GHz – 2 GHz. Dotazovač vysílá na frekvenci 1030 MHz. Odpovídač potom využívá frekvence 1090 MHz. Dané signály jsou vysílány s vertikální polarizací.



Obrázek 10: SSR princip [10]

Systém nemění v cyklu své nastavení antény. Počítač přijímá na frekvenci a pokud přijde dotaz, zareaguje na něj odpovědí s požadovanými údaji. Formát daného dotazu, odpovědi, frekvence, kódování a všechny další parametry jsou jednoznačně definovány normou SSR. Díky tomu jsou jednotlivé systémy mezi sebou vzájemně kompatibilní.

Bohužel toto přináší i několik problémů. Můžeme například zachytit jinou odpověď. Některé problémy lze řešit celkem jednoduše. Například potlačení odpovědí vyvolaných dotazem jiného radaru, tedy filtrací příchozích zpráv. Někdy dojde i k překrytí dvou odpovědí. To už vyžaduje náročnější řešení, které ne vždy je úplně spolehlivé. [10]

3.1 Měření souřadnic cíle

Pro zjištění polohy cíle je třeba znalost dalších informací: radiální vzdálenost, azimut, nadmořská výška, geografická poloha stanice.

3.2 Radiální vzdálenost

Při výpočtu vzdálenosti se opět bere v potaz doba mezi odesláním dotazu a přijmutím odpovědi. Tento čas obsahuje dobu zpoždění odpovídače, kterou bere v potaz norma a definuje ji jako 3 μ s. Vzdálenost se tedy určí následující rovnicí.

$$s = \frac{(t_p - t_v) - 3}{2} * c$$

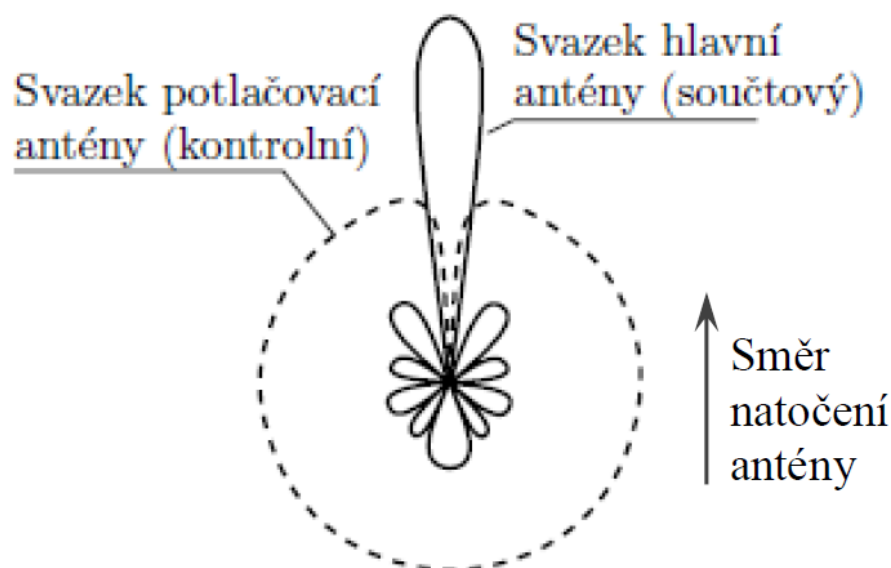
Kde t_p je čas příjmu a t_v je čas vyslání signálu. C je potom konstanta (rychlost světla). Chyba závisí především na toleranci zpoždění odpovědi odpovídače a chybné určení času příchodu odpovědi. Toto zpoždění a její přesnost u odpovídače je dána normou. Tolerance je zhruba $\pm 0,5 \mu$ s a představuje zhruba ± 75 m. [10]

V reálném využití není rozlišovací schopnost tak významná jako u primárních radarů. Avšak dva cíle lze rozlišit pouze tehdy, jestliže se dané odpovědi nepřekrývají. Jestliže odpověď trvá zhruba 20 μ s, je rozlišovací schopnost určená 3 km. Pokud je daná vzdálenost menší dochází k překrývání odpovědí.

3.3 Azimut cíle

Pro určení úhlu se využívá směrová anténa a její charakteristika, tedy úhel natočení. Jelikož odpovídač v hustém provozu nezvládá zpracovat všechny dotazy a odpovídá náhodně na několik dotazovačů, je nutné využít jistých metod, které tento problém řeší.

Velmi problematický je taktéž příjem postranními svazky antény. Díky tomu se může úhel určit špatně. Dále to může způsobit vyslání signálu na špatný prvek, který v té dané chvíli je mimo oblast zájmu, tedy hlavního svazku antény. Tento problém se však dá řešit konstrukcí potlačovací antény.



Obrázek 11: Hlavní a potlačovací anténa [10]

Vyslání signálu na dotaz je pak rozděleno na část pulsů vyslána hlavní anténou a na část vyslanou potlačovací anténou. Odpovídač na sledovaném objektu potom identifikuje, z jakého svazku byl signál odeslán a na ty z postranního svazku nereaguje.

Následně dle amplitudy přijatých signálů na hlavní a potlačovací anténě lze ignorovat odpovědi zachycené mimo hlavní svazek. Tím dojde k zahození odpovědí na dotazy jiných dotazovačů. [10]

Jelikož poměr amplitud signálu nezávisí na výkonu odpovídače, je určení úhlu nezávislé na výkonu odpovědi. Ze dvou odpovědí lze zjistit polohu odpovídače. U jedné odpovědi bohužel ne. Pro určení úhlu lze využít **monopuls**, kdy je signál přijat současně více stejnými anténami a úhel se stanoví porovnáním přijatých signálů. Tím lze získat úhel i z pouhé jedné odpovědi. [10]

3.4 Výška cíle

Není možné určit výšku cíle pomocí sekundárního radaru. K určení výšky se tedy využívá údaj z barometrického výškoměru, který se nachází na palubě letadla. Údaj o výšce je zaslán v odpovědi na dotaz. To má výhodu v tom, že je údaj měřen přímo v letadle a nezávisí tedy jeho přesnost na vzdálenosti letadla od stanice.

Výška se udává ve stopách, jsou k tomu historické důvody. Výška se udává nad hladinou moře. Pro označení výšky se užívá pojem **letová hladina**. Jedna letová hladina představuje zhruba 100 stop.

Výškoměr zjistí výšku podle tlaku. Slouží k tomu následující výpočetní rovnice.

$$H = (18400 + 67t_m) * \ln\left(\frac{p_0}{p}\right)$$

Symbol H je výsledná nadmořská výška v metrech, t_m je střední teplota vzduchu. Symbol p je tlak vzduchu v době měření a p_0 je tlak vzduchu přepočítaný na hladinu moře.

4 CHYBOVOST

Data získaná ze systému určování polohy, například GPS, nejsou v reálném provozu 100% přesná. Vždy je tu jistá chyba, kterou jsou data zatížena. Největší vliv na chybu má například počet satelitů, používaných k určení polohy. Tento počet může snížit nepřímá viditelnost, atmosférické podmínky, pohyb v rušném prostředí. Hodně závisí na použitém systému – GPS, Galileo, GLONASS apod. Tato nepřesnost ovlivní jednak rychlost a jednak polohu sledovaného objektu

Při určení rychlosti se udává velikost chyby parametrem HVE. Ten se měří v jednotkách metr za vteřinu. Reálná rychlost v horizontální rovině se udává intervalem s 95% pravděpodobností. Interval je ohraničen naměřenou rychlostí s rozsahem od mínus chyby do plus chyby. Matematicky vyjádřeno: $\langle V - \text{chyba}, V + \text{chyba} \rangle$, kde V je naměřená rychlost. Z toho vyplývá, že hodnota parametru chyby je vždy kladná.

Chybu určení polohy udává parametr HOFM. Tento parametr je taky někdy označován jako EPU. Parametr je definován poloměrem kruhu v metrech, ve kterém se nachází reálná poloha s danou 95% pravděpodobností. Hodnota tohoto parametru je opět vždy kladná.

4.1 Uměle vytvořené chyby

Jak již bylo vysvětleno, nepřesnost je například určena intervalem spolehlivosti s 95% pravděpodobností, že se reálná hodnota nachází v daném intervalu. Tento interval lze aplikovat na rychlost, definovanou jedním rozměrem. A lze to aplikovat i na polohu, která je určena minimálně dvourozměrnou veličinou. Při dvourozměrné veličině je interval definován kruhem. Při trojrozměrné veličině by byl definován koulí.

Při zatížení přesné hodnoty chybou, bychom tedy mohli vygenerovat náhodné číslo v rozsahu od -1 do 1 pomocí Gaussova rozdělení. Nastavili bychom střední hodnotu na 0 a směrodatnou odchylku na 0,5. Získali bychom tedy hodnotu v intervalu 95% pravděpodobností, kterou bychom vynásobili veličinu a tím ji zatížili chybou. To má jednu velkou chybu. Způsobilo by to velké skoky v dané poloze. V reálu je poloha hodně blízko té původní. Musíme se tedy zajistit určitá korelace dané chyby, tato korelace bude ve vztahu času a předchozí polohy. Následující poloha, či chyba, musí záviset na předešlé chybě a poloze. Dále je nutné vydefinovat typy chyb.

4.2 Typy chyb

Chyby lze rozdělit na tři kategorie.

4.2.1 Instrumentální omezení

Měřicí nástroj může být použit pouze pro měření s daným stupněm jemnosti. Samotné měření je tedy maximálně přesné tak, jako jsou přesné použité nástroje.

4.2.2 Systematické chyby

Systematická chyba se během měření nemění. Například váha, která nebyla správně nastavena na nulu při nulové zátěži. Tato váha potom při každém vážení změří špatnou hodnotu se stejnou chybou. Systematické chyby jsou buď odhalené a následně potom odstraněné, nebo jsou schované v pozadí a produkují posun od skutečné hodnoty.

4.2.3 Náhodné chyby

Ty vznikají z nekontrolovaných změn v měřící technice. Nebo v malých změnách v experimentálním prostředí. Tyto chyby ovlivňují přesnost. Jsou skutečně náhodné, pokud jsou výsledky velkého počtu pokusů kombinovány.

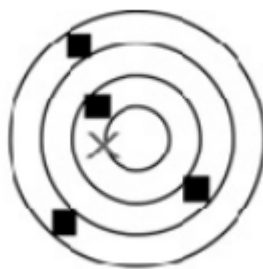
4.3 Přesnost a správnost

Přesné měření je takové, kde nezávislé měření stejné veličiny nám udává stejné nebo velmi podobné výsledky, ovšem není zaručená správnost výsledků.

Správné měření je takové, kde po vyhodnocení naměřených výsledků dojdeme ke správné hodnotě. [7]

4.3.1 Malá přesnost, malá správnost

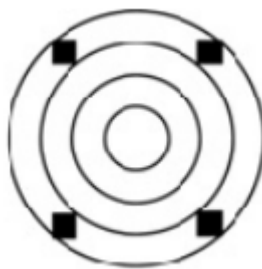
Průměr rozdílných výsledků neodpovídá správné hodnotě.



Obrázek 12: Malá přesnost, malá správnost [7]

4.3.2 Malá přesnost, velká správnost

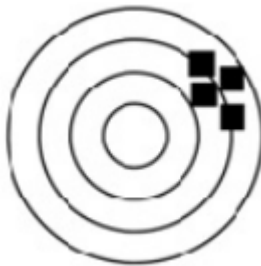
Průměr rozdílných výsledků odpovídá správné hodnotě.



Obrázek 13: Malá přesnost, velká správnost [7]

4.3.3 Velká přesnost, malá správnost

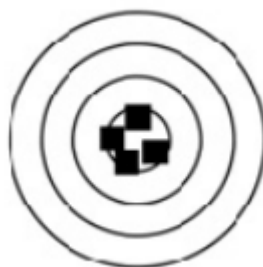
Průměr velmi podobných výsledků neodpovídá správně hodnotě.



Obrázek 14: Velká přesnost, malá správnost [7]

4.3.4 Velká přesnost, velká správnost

Průměr velmi podobných výsledků odpovídá správně hodnotě.



Obrázek 15: Velká přesnost, velká správnost [7]

4.4 Markovův rozhodovací proces

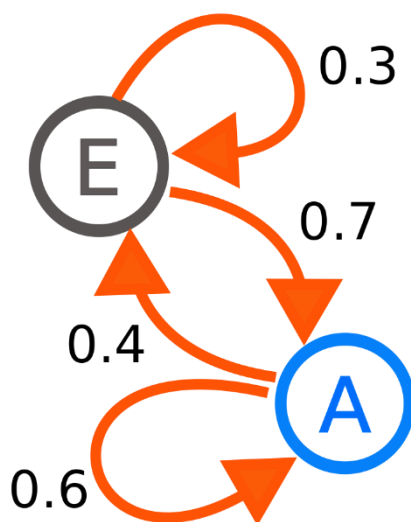
Autor Markovských rozhodovacích procesů je ruský matematik Andrej Markovov. Tyto procesy jsou známy od padesátých let minulého století. V případech, kdy máme k dispozici výsledky, které jsou zčásti náhodné a zčásti pod vlivem uživatele, nám poskytují matematický rámec pro modelaci rozhodování. Slouží a využívají se pro potřeby optimalizací a řešení problému prostřednictvím zpětnovazebního učení a dynamického programování. V současné době je jejich uplatnění největší v průmyslové výrobě, ekonomii, robotice a automatizovaného řízení. [5]

V každém časovém okamžiku má proces určitý **stav** s a je dostupná jakákoli **akce** a , která je dostupná ve stavu s . Systém v dalším časovém okamžiku reaguje náhodným přesunutím do nového stavu s_2 a vrací odpovídající užitek $R_a(s, s_2)$. Proces je tedy diskrétní, stochastický a kontrolovaný. Šance, že proces vybere zrovna s_2 jako nový stav je ovlivněna vybranou akcí. Tuto pravděpodobnost můžeme vyjádřit funkcí na přechod stavu $P_a(s, s_2)$. Z toho vyplývá, že další stav s_2 závisí na současném stavu s a na uživatelské akci a . Daný stav a akce jsou závislé na všech předchozích stavech a akcích. Tomuto stavu se odborně říká Markovská vlastnost.

Markovské rozhodovací procesy jsou rozšířením Markovských řetězců. Toto rozšíření je v přidání akcí a užiteků. Umožňují tedy výběr a poskytují motivaci. Kdyby existovala pouze jedna akce či pokud by byla stejná akce pro všechny stavy, znamenalo by to, že Markovský rozhodovací proces se zredukoval na Markovův řetězec. [5]

4.5 Markovův řetězec

Popisuje diskretní náhodný proces, pro který platí, že pravděpodobnost přechodu do dalšího stavu závisí pouze na současném stavu a nikoli na stavech předchozích. Tato vlastnost umožňuje proces znázornit stavovým diagramem, kde z každého uzlu grafu, reprezentující stav, vycházejí hrany možných přechodu do stavu jiných s udanou pravděpodobností.



Obrázek 16: Stavový diagram řetězce [6]

Na obrázku vidíme, že pokud je systém ve stavu A, přejde s pravděpodobností 0,6 zpět do stavu A. Nebo s pravděpodobností 0,4 do stavu E, ze kterého se s 0,7 dostane zpátky do A, či zůstane s 0,3 šancí ve stavu E. Chování systému je tedy bezpaměťové a není potřeba si pamatovat historii. Stačí vždy aktuální stav. Vhodný reprezentant takového systému je například konečný automat. Řetězce mají velké praktické využití v informatice a dalších společenských vědách.

5 UMĚLÉ ZATÍŽENÍ CHYBOU

Pro naše potřeby bude nutné vydefinovat možné způsoby, jak data zatížit chybou. Tyto chyby se musí jevit jako co nejvíce podobné realitě a musíme se vyvarovat akademickým a umělým vzorcům.

Jedna z možností je například vliv chyb při určení poloh stanic. Taková chyba zaměření polohy se projeví v chybě měření časů příchodů signálu na stanici. Můžeme v závislosti vzdálenosti stanice od určené polohy dostat standardní odchylku určení souřadnice. Například tedy, pokud bude poloha posunuta zhruba o padesát centimetrů, nastane chyba měření času asi o necelé dvě nanosekundy. [7]

5.1 Výpadky

Jako velmi účinný nástroj pro generování chyb se jeví modelovat výpadky a zcela tak ignorovat některé zprávy. To se v reálu děje často. Zprávy chodí necelé, nebo zcela vůbec. Tato chyba se jednoduše namodeluje, když se nastaví pravděpodobnost, se kterou daná zpráva vůbec nepřijde nebo naopak, že přijde.

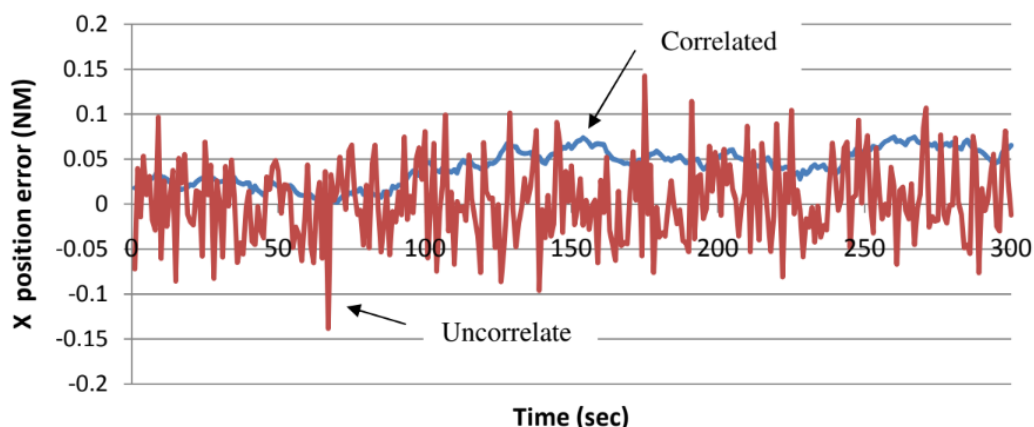
5.2 Korelovaná chyba pomocí Gauss-Markova procesu

Chyby při měření polohy letadla jsou obvykle modelovány pomocí **Gaussova rozdělení** náhodných čísel, kde časová korelace mezi časem chyby měření není modelována. K modelování je tedy třeba použít **Gauss-Markovův proces**, který koreluje chyby měření polohy pro použití v popsané situaci.

Proces prvního řádu je náhodný proces, který popisuje chování náhodné proměnné jako funkce času. Jelikož simulace **Monte Carlo** je diskrétní časová simulace, využije se **diskrétní Gauss-Markovův proces**.

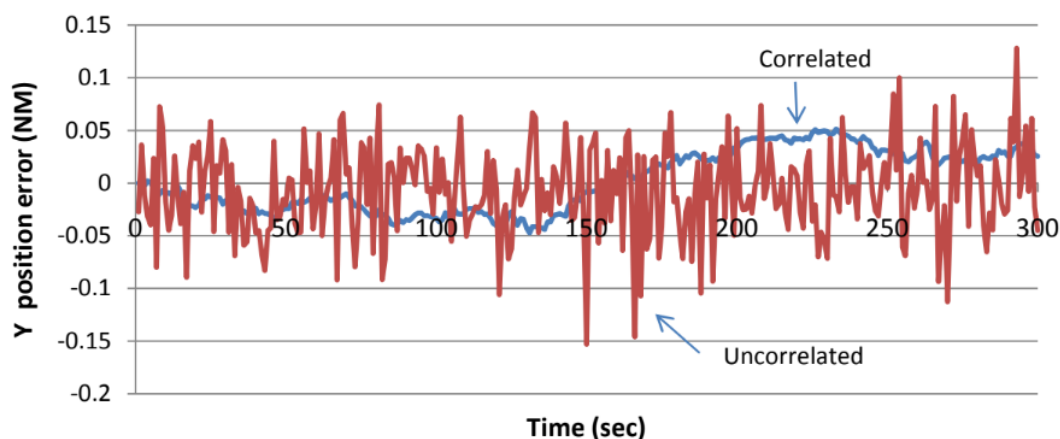
U ADS-B je chyba měření horizontální polohy GPS obecně modelována jako náhodná veličina **Rayleighova rozdělení**, kde chyby měření X a Y polohy letadla jsou modelovány jako Gaussovy náhodné distribuční rozdělení s 95% přesností horizontální chyby. Výslednou chybu korelujeme a můžeme získat následnou chybovost.

První graf ukazuje chybovost souřadnice X.



Obrázek 17: Korelovaná a nekorelovaná X souřadnice [8]

A velmi podobně získáme i Y souřadnici.



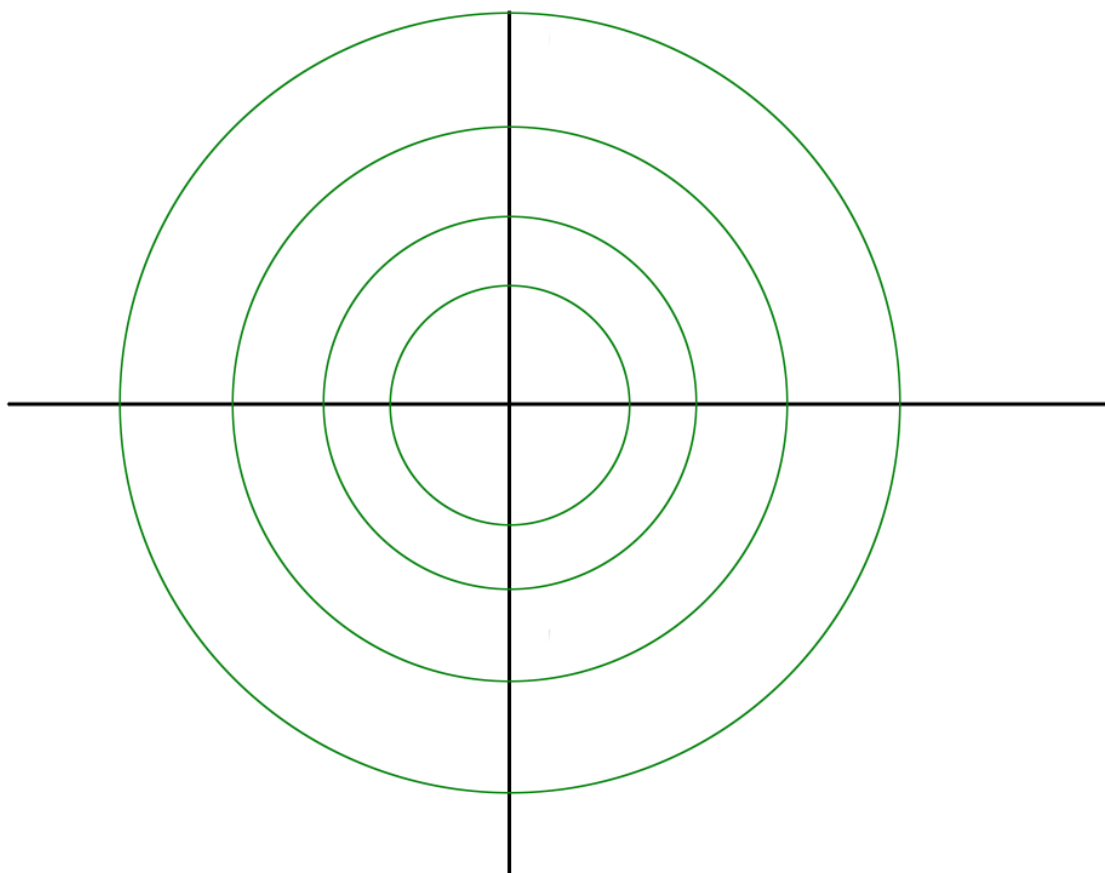
Obrázek 18: Korelovaná a nekorelovaná Y souřadnice [8]

5.3 Pevná chyba

Další možností, jak zkreslit původní souřadnice, je přidělit jim pevnou chybu. V závislosti na vzájemné poloze antény přijímače a daného objektu, můžeme zatížit daná data pevnou chybou. Tato chyba může být vypočítaná a určena právě ze vzájemné polohy. V reálném provozu to funguje podobně, anténa na různé vzdálenosti zachytí signál vždycky poněkud jinak odražený a jinak zkreslený, to může dávat nepřesné výsledky.

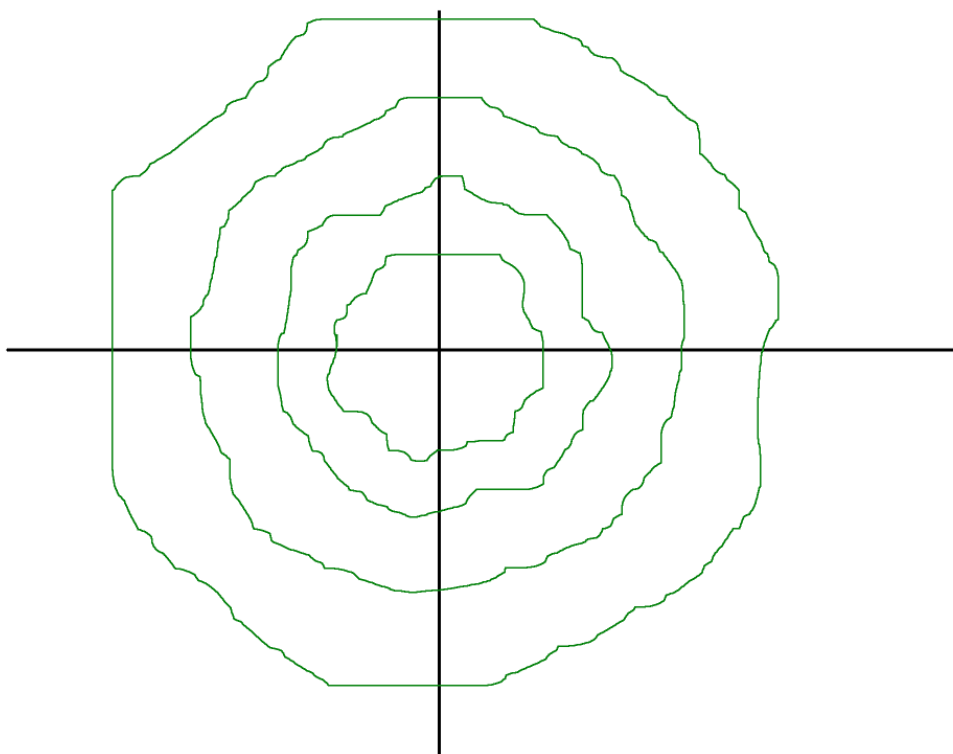
V principu dojde k rozdělení okolí antény na minimálně dvě osy: **X** a **Y**. Mohli bychom využít tři os, abychom mohli polohu určit i s výškou, ale pro zjednodušení použijeme pouze dvě. V různých vzdálenostech od antény kreslíme kružnice, které nám rozdělí danou oblast na

různé intervaly vzdáleností od antény a na základě této vzdálenosti dojde k vypočítání chyby.



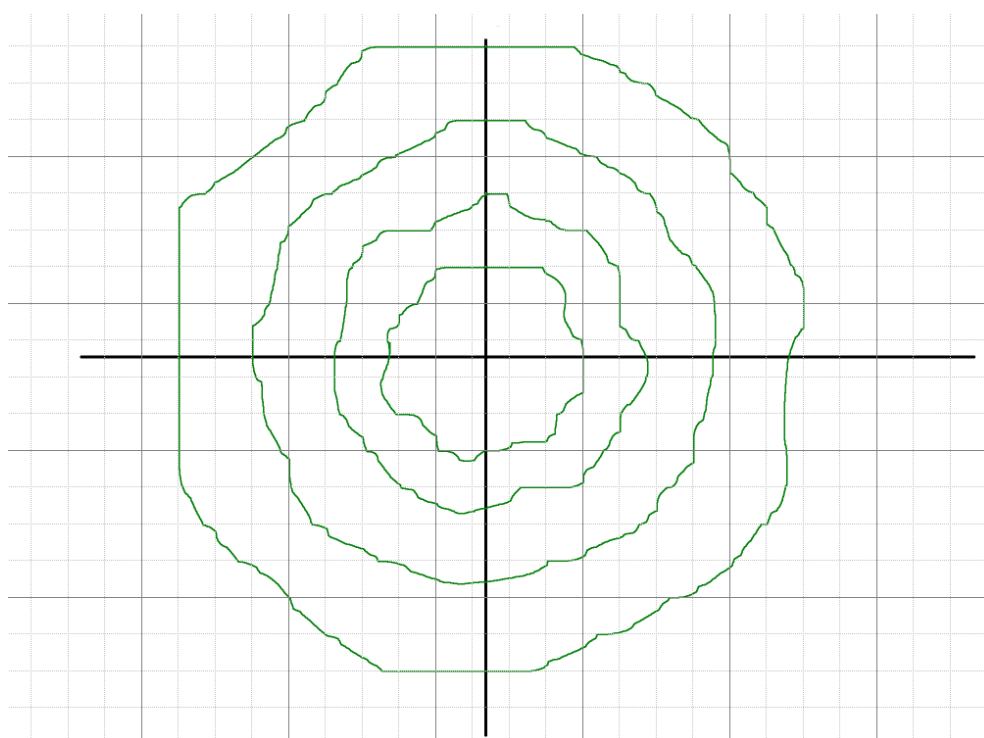
Obrázek 19: Kružnicové označení vzdáleností od antény

Tento příklad je ovšem moc přesný a kružnice dokonalá. Proto se jednotlivé kružnice zdeformují. Dostaneme tedy víc reálnější rozdělení, které způsobí například různé nerovnosti terénu, stavby a další odrazové plochy, nedokonalost antény nebo další rušení.



Obrázek 20: Zdeformované označení vzdáleností od antény

Nyní stačí tuto plochu rozdělit na jednotlivé části. Jelikož jsme využili pouze dvou os, rozdělíme danou plochu pouze dvourozměrně. Označíme jednotlivé souřadnice po vždycky stejném kroku. Dostaneme tedy mřížku, kde každá buňka má svoje unikátní X a Y souřadnice.



Obrázek 21: Označení vzdáleností s rozdělením mřížkou

V jednotlivých buňkách bude vždycky konkrétní chyba. Chyba by měla být rozdělena právě podle deformovaných elips. Při využití tabulky pak jednoduše stačí zjistit relativní polohu objektu vůči poloze stanice. Vypočítat vzdálenost v X a Y souřadnicích. Určit dané místo a buňku s chybou a tuto chybu pak aplikovat na danou polohu objektu.

6 NÁVRH

Nejprve je nutné si ujasnit požadavky na aplikaci, která bude emulovat činnost pozemních stanic. V rámci průběhu analýzy je nutné vydefinovat, co vlastně musí daný emulátor umět a těmto požadavkům je nutné nastavit prioritu.

Při stanovení požadavků v první fázi se vyhneme pozdějším komplikacím při vývoji. Jelikož požadavky popisují, co bude program dělat. Je zvykem označit každý požadavek unikátním identifikátorem s prioritou a odhadovanou cenou.

Cílem této praktické části je vytvořit konzolovou aplikaci. Aplikace bude využita spolu s dalšími projekty na vývoj algoritmů pro odhalování falešných cílů. Jako zdroj vstupních dat bude použit jeden z těchto projektů.

6.1 Funkční požadavky

Funkční požadavky určují, co musí aplikace dělat a identifikují nutné procesy a akce, které musí být vykonány. Analýza funkčních požadavků je součástí takzvané funkční analýzy. Tvoří tedy byznysové cíle aplikace.

Tabulka 1: Funkční požadavky

Identifikátor	Popis požadavku
FR001	Aplikace bude konfigurovatelná XML souborem.
FR002	Aplikace bude jednoduchého konzolového typu.
FR003	Aplikace bude mít vstup ze souboru nebo z TCP serveru.
FR004	Aplikace vyfiltruje vstupní zprávy podle konfigurace.
FR005	Aplikace zkreslí chybou vstupní zprávy podle konfigurace.
FR006	Aplikace umožní odesílat výstup do souboru nebo přes TCP.
FR007	Aplikace bude emulovat činnost pozemních stanic třech typů.
FR008	Aplikace bude validovat příchozí zprávy pomocí XSD.
FR009	Aplikace poskytne výstupní souřadnice ve více formátech.
FR010	Aplikace bude validovat konfigurační soubor.

6.2 Nefunkční požadavky

Nefunkční požadavky jsou součástí takzvané **Service-level requirements**. Tyto požadavky mají mnohdy kritický vliv na aplikaci, ale jejich účelem není podpoření byznysového cíle, ale vyvinout stabilní a kvalitní aplikaci. Patří sem spolehlivost, rozšiřitelnost, udržitelnost, škálovatelnost a bezpečnost.

Tabulka 2: Nefunkční požadavky

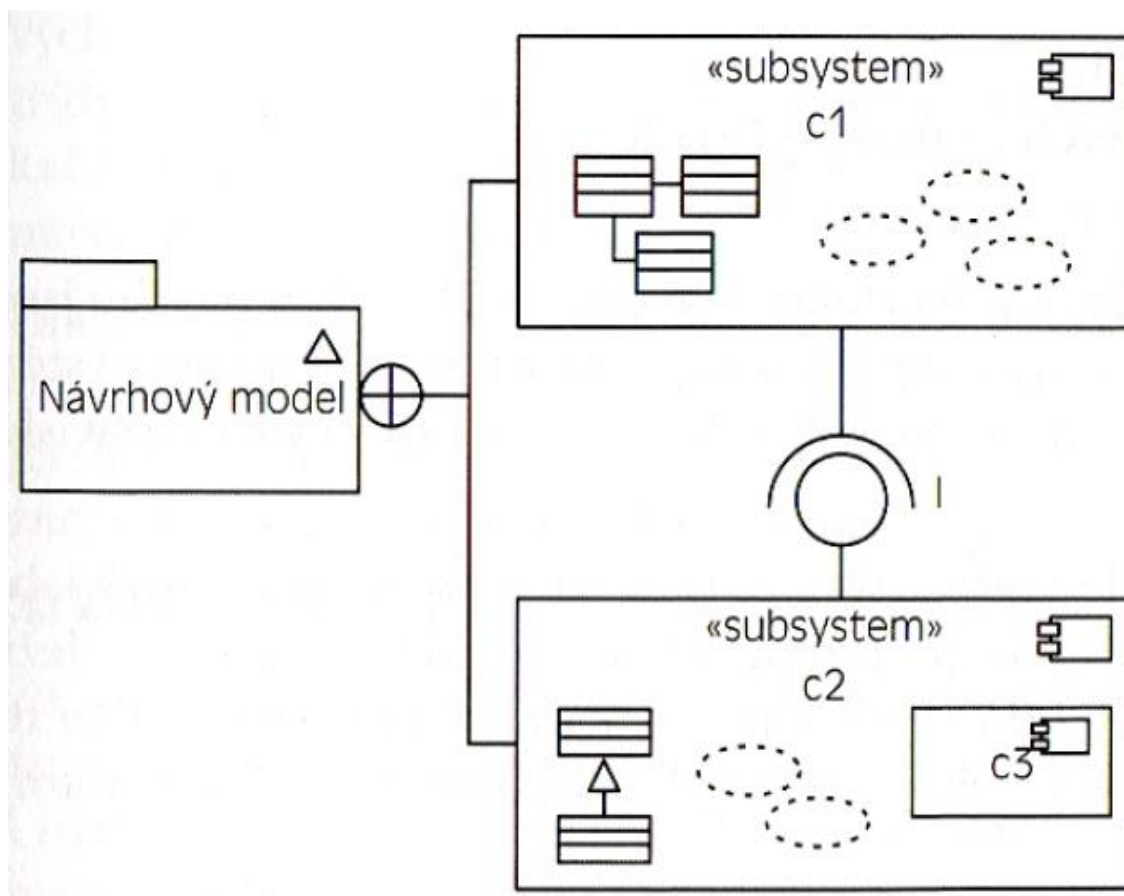
Identifikátor	Popis požadavku
NR001	Aplikace bude vytvořena v programovacím jazyce Java a frameworku Spring.
NR002	Při návrhu budou dodržena pravidla objektově orientovaného programování.

6.3 Analýza

Analýza je především zaměřená na tvorbu logického modelu připravovaného systému. Zachycuje funkce, jež tento systém má poskytovat, aby byly splněny požadavky. Smyslem návrhu je tedy přesně specifikovat, jak daný systém implementovat. To má dva úhly pohledu. Jeden z problémové domény a druhý z pohledu domény řešení. Požadavky vycházejí z problémové domény. Při analýze je zkoumána tato doména z pohledu budoucích uživatelů systému. Výsledný návrh tvoří sloučení technických řešení z domény řešení. Cílem je vytvořit model systému, který lze implementovat.

6.4 Metamodel

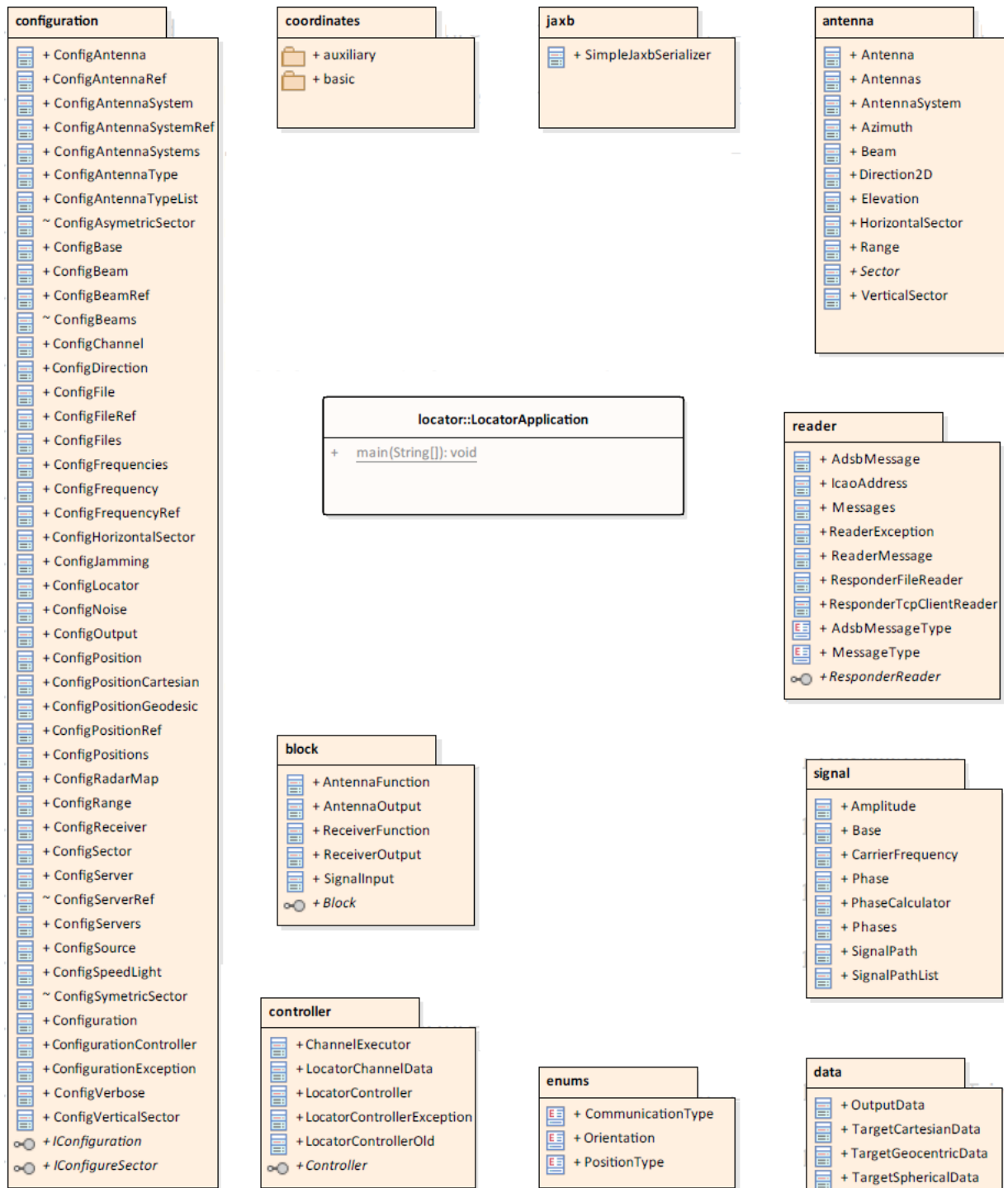
V návrhovém modelu se vyskytuje právě jeden návrhový systém. Návrhový systém obsahuje více návrhových podsystémů. Během návrhu se používá termín **podsystém** nebo taky **subsystém** častěji než pojem **balíček**. Vyjadřuje to fakt, že již není modelováno čistě koncepčně, ale už i z pohledu fyzického modelu.



Obrázek 22: Metamodel návrhového modelu [11]

6.5 Model

Pro danou aplikaci by se dal vytvořit podobný model, jako je na následujícím obrázku. Tento model zobrazuje možné balíčky a třídy aplikace.



Obrázek 23: Model aplikace

7 IMPLEMENTACE

Pro vývoj aplikace byl zvolen programovací jazyk **Java** a **Spring Framework**. Veškeré použité technologie jsou open source, nebude tak problém s jejich využitím. Není nutné zakoupit žádné placené knihovny.

7.1 Spring Framework

Populární open source aplikační rámec, který lze jej označit také jako **kontejner**. Primárně slouží pro vývoj **J2EE** aplikací. První verze byla napsána a vydána panem Rodem Johnsonem v říjnu 2002. Ten ji vydal spolu s publikací své knihy Expert One-on-One J2EE Design and Development. V knize se zabývá vývojem právě J2EE aplikací a věnuje zvláštní pozornost problémům, se kterými se programátoři setkávají. V knize popisuje kód Springu a nazývá jej **Interface21** a jeho výhody pro usnadnění J2EE aplikací. Za pomoci jeho kamaráda Juergena Hoellera je později Spring rozšířen a pod oficiálním názvem Spring Framework uvolněn jako open source.

Spring Framework byl poprvé uvolněn veřejnosti pod licenci Apache 2.0 a to v roce 2003. První verze pak byla vydána v březnu následujícího roku a další potom v září téhož roku a následně v březnu 2005. Ve verzi 1.2.6 získal Spring Framework v roce 2006 ocenění JAX Innovation Award a Jolt Productivity Award. V současnosti je dostupná verze 5.0.0.

Hlavním důvodem vzniku Springu bylo usnadnění vývoje především **enterprise aplikací**. Důraz se dal na odstranění těsných programových vazeb jednotlivých POJO objektů a vrstev pomocí návrhového vzoru **Inversion of Control**. Možnost volby implementace business vrstvy pro aplikační architekturu, a ne aby architektura předepisovala implementaci. Podpora implementace komponent pro přístup k datům v databázi, ať už formou ORM či přímého JDBC. To vedlo k vytvoření nástrojů jako je například Hibernate. Důraz se tady dal na usnadnění psaní a používání JUnit testů.

Celé jádro frameworku je postaveno na využití návrhového vzoru Inversion of Control. Tento vzor funguje na principu přesunutí zodpovědnosti za vytvoření a provázání objektů z aplikace na Framework. Jednotlivé instance objektů lze získat za pomoci **dependency injection**, neboli dosazování závislostí. Způsob dosazování objektů je například **Setter Injection**, kdy se daný objekt nastaví přes nastavovací metodu. Dále například **Constructor Injection** a **Interface Injection**. Instance objektů vytvořené frameworkem se odborně nazývají **JavaBeans** a jsou vytvořeny na základně konfiguračního souboru ve formátu XML.

Framework se nezabývá řešením nebo vysvětlením či vyřešením problémů. Místo toho poskytuje systém prověřených a dobře fungujících principů a open source nástrojů. Slouží tedy k zjednodušení návrhu, jednoduché testování, modulárnost a neinvazivnost a přepoužitelnost.

7.2 XML

Pro nastavení aplikace je obecně vhodné použít konfigurační soubor ve formátu XML. Českým názvem rozšiřitelný značkovací jazyk je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Jedná se o zjednodušenou verzi staršího jazyka SGML. XML umožňuje jednoduché vytváření značkovacích aplikací (jazyků). Používá se pro serializaci dat pro různé účely a různé typy dat. Zpracování je podporováno širokou řadou nástrojů a programovacích jazyků. Za zmínku stojí dva stejně populární, ovšem poněkud zjednodušené nástroje, a to JSON a YAML.

Praxe ukázala, že je velmi nevhodné zasílat dokumenty v uzavřených formátech, na které je potřeba specifický software. Stejně tak bylo potřeba najít univerzální prostředek na přenos dat z různých systémů. Vznikla tak potřeba vytvořit jednoduchý otevřený formát, který by nebyl pevně spjat s žádnou platformou nebo technologií. V tomto ohledu se nyní XML jeví jako skvělý nástroj. Jelikož se ve výsledku jedná o pouhý text a může být tak zpracován libovolným textovým editorem.

Od počátku vývoje se počítalo i s jiným jazykem, než je angličtina. Jazyková sada je implicitně využívána ISO 10646 (nebo Unicode). Můžeme však využít i libovolného jiného kódování, například ISO 8859-2 nebo CP-1250. Musí však být v každém dokumentu uvedeno, o jaké kódování jde.

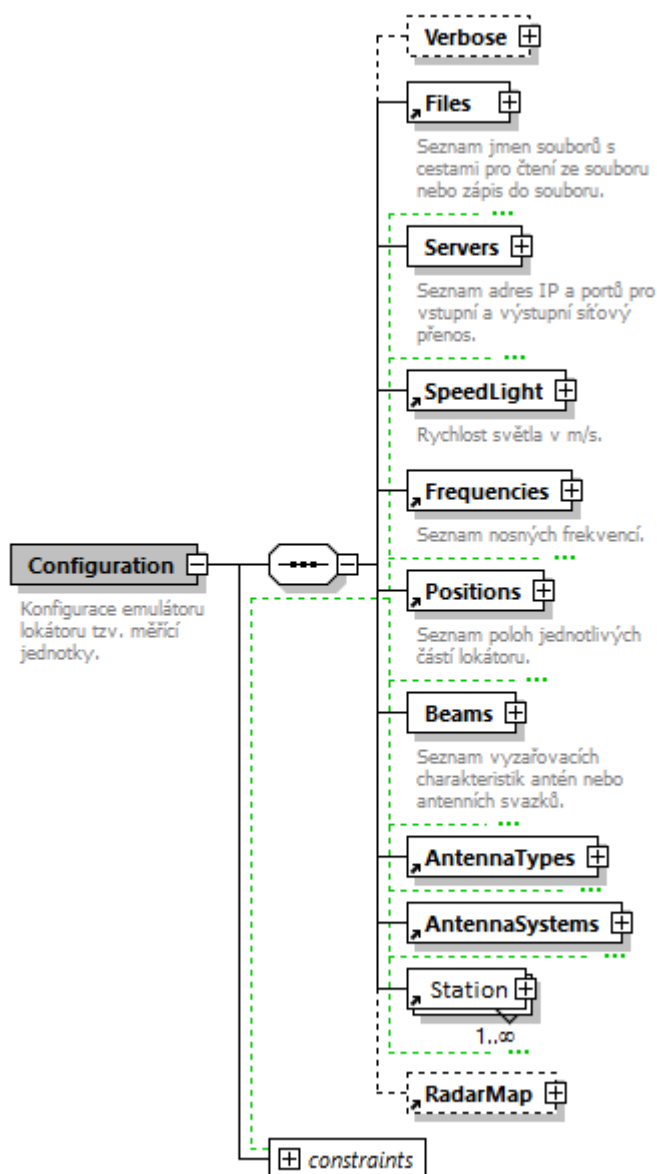
XML, na rozdíl od JSON či YAML, obsahuje tagy, což jsou značky, které vyznačují v dokumentu význam jednotlivých částí. V důsledku toho obsahuje dokument víc informací, než kdyby se využilo značkování pomocí vzhledu. Podobně jako v editoru dokumentů: styl písma, odsazení, číslování a podobně. XML dokumenty jsou tedy informačně hustější, avšak to lze s výhodou použít.

7.3 Schéma

Jelikož XML nemá předdefinované žádné tagy (značky), je zapotřebí definovat pro každé použití vlastní. Tyto tagy je možné definovat v souboru DTD nebo XSD. Po propojení doku-

mentu se schématem je možné zkontrolovat validaci daného dokumentu. Jednoduše tím zjistíme, zda daný XML soubor odpovídá definici či ne. Program, který danou kontrolu provádí, nazýváme **parser**.

Pro naši aplikaci tedy musíme vytvořit schéma konfiguračního souboru a následně dle zadaného schématu vytvořit i samotný dokument, který použijeme. Na následujícím obrázku vidíme grafické schéma našeho XSD souboru.

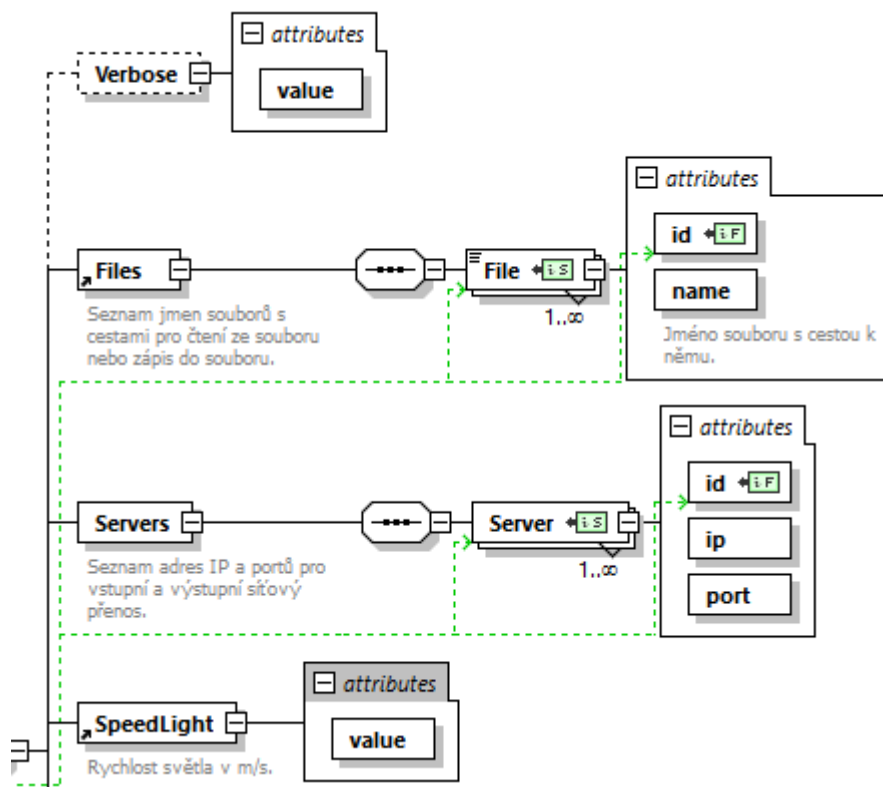


Obrázek 24: Grafické schéma XSD

Můžeme vidět hlavní značku **configuration**, která zapouzdřuje několik značek, každá má svůj význam.

7.3.1 Zdroje

V následujících elementech specifikujeme, ze kterých zdrojů bude aplikace čerpat data. Blíže nám to specifikuje obrázek níže.



Obrázek 25: Zdroje aplikace

První tag se jmenuje **verbose** a má v sobě pouze atribut *value*. Tento tag slouží pro zapnutí nebo vypnutí detailnějšího výpisu.

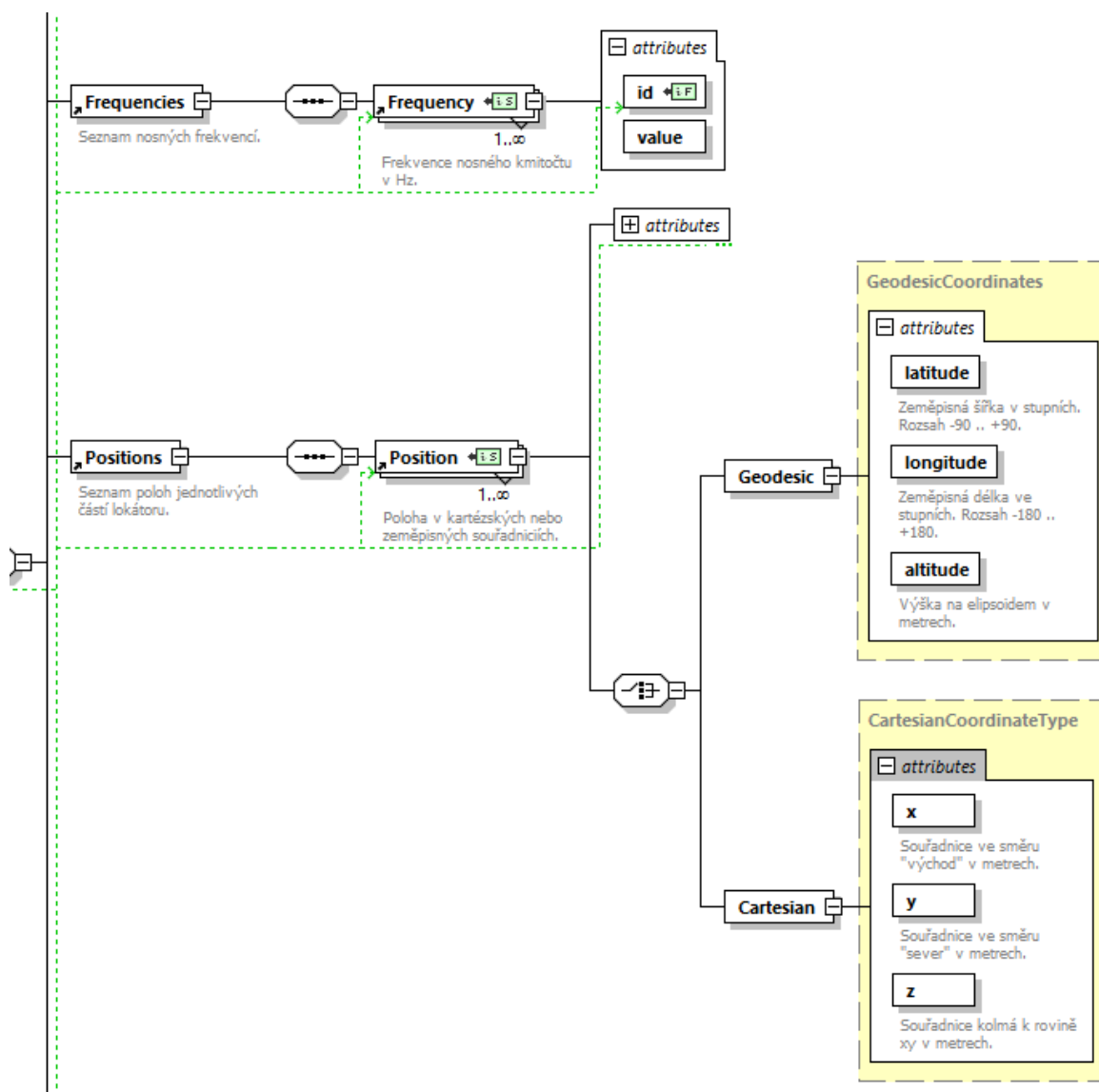
Následuje **files**, ve kterém se nastavuje zdroj ze souboru na disku. Je nutné nastavit *id* a cestu k souboru, ze kterého budeme čerpat příchozí zprávy. Zdrojový soubor by měl obsahovat zprávy ve formátu XML, jinak program nebude fungovat.

Jestliže nemáme k dispozici žádný soubor, můžeme zprávy přijímat ze serveru. Toto nastavení využijeme, jestliže chceme emulovat činnost v reálném čase. V elementu tedy definujeme opět unikátní *id*, *IP adresu* protokolu TCP/IP a *port*, na kterém budeme poslouchat. Server by měl opět vysílat zprávy ve formátu XML, který definuje XSD.

Jako poslední element je hodnota na nastavení rychlosti světla. Tento element zatím nemá využití a slouží pro potřebné rozšíření aplikace do budoucna.

7.3.2 Pozice

Další důležitou součástí jsou frekvence a pozice, které řeší následující značky.



Obrázek 26: Frekvence a pozice

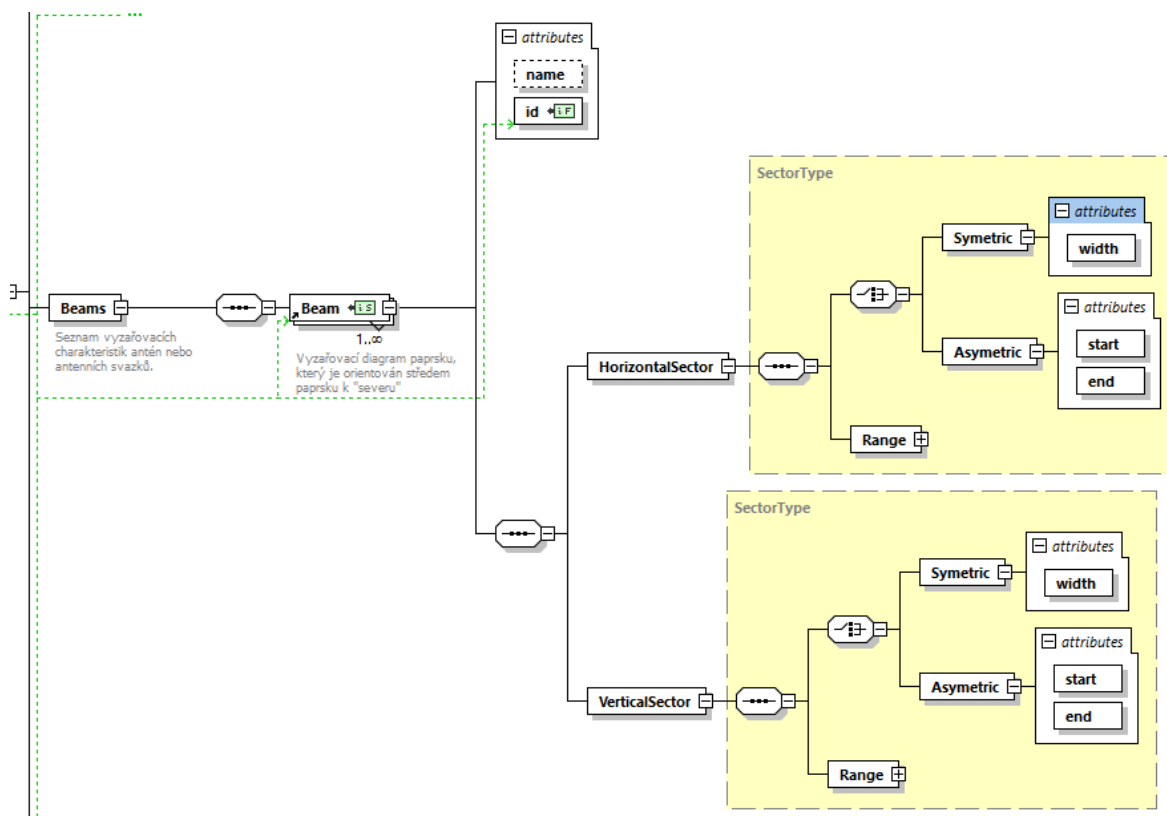
Frekvence jsou definovány unikátním *id* a *hodnotou*, která udává kmitočet v Hz. V jakých jednotkách je specifikováno přímo v dokumentu v popise, tudíž je hned jasné, jaké hodnoty se mají zadat.

Dále následuje seznam pozic, které jsou definovány v zeměpisných souřadnicích nebo kartézských systémech. Zeměpisné údaje vyjadřuje zeměpisná šířka, zeměpisná výška a nadmořská výška. Tento systém je často využíván například pro GPS souřadnice. Druhým způsobem je možnost určit polohu pomocí souřadnic X, Y, Z. Tyto souřadnice jsou vhodnější například pro vykreslení na mapě. Jelikož při použití zeměpisných souřadnic, by v důsledku zakřivení

Země, bylo grafické zobrazení dvou bodů různé na rovníku a blízko Severního nebo Jižního pólu Země.

7.3.3 Paprsky

Seznam vyzařovacích charakteristik antén nebo anténních svazků, vyjadřuje diagram paprsku.

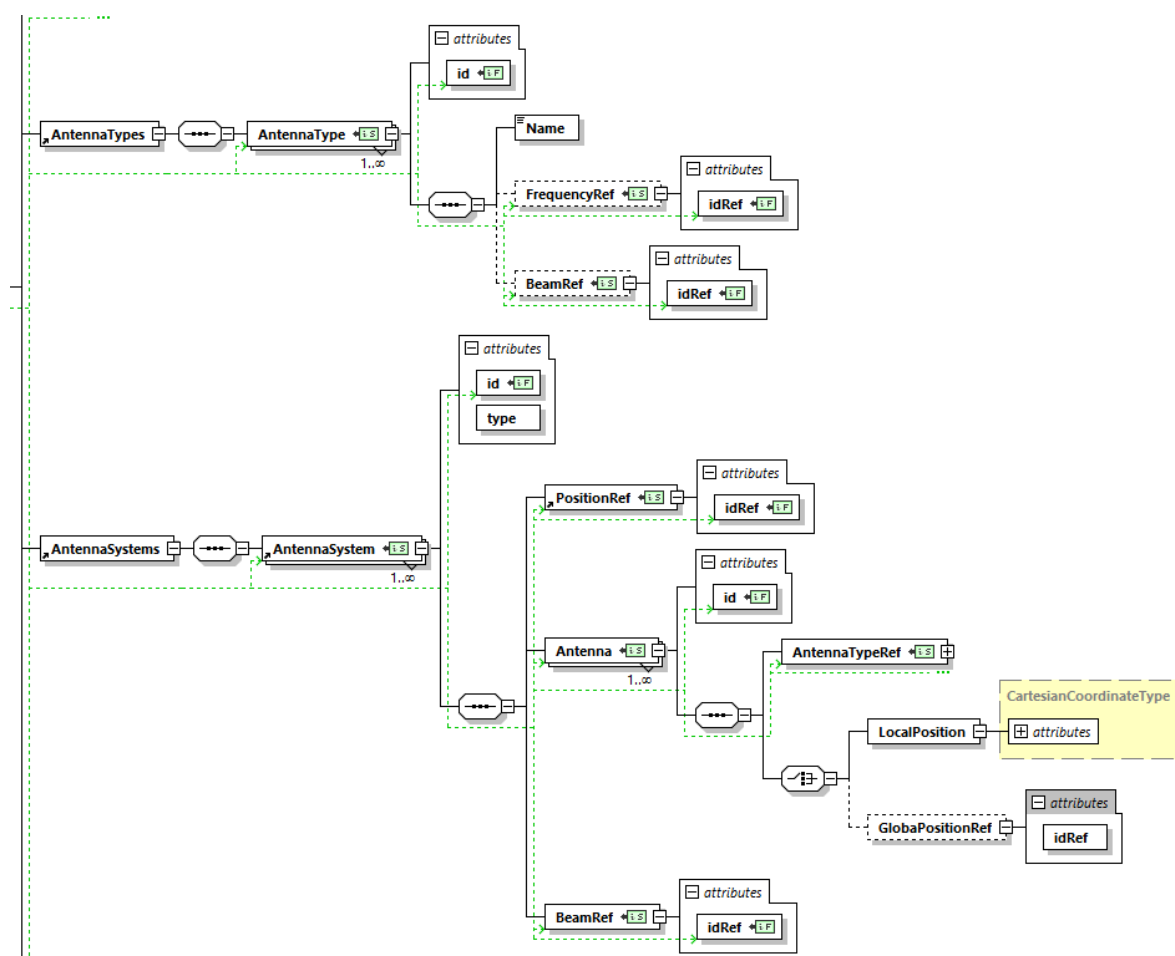


Obrázek 27: Beams

Každý paprsek definuje buď horizontální nebo vertikální sektor. Dále se dělí na symetrický nebo asymetrický. Symetrický znamená, že má rozsah například od -10° do 10° . Asymetrický potom, že má rozsah například od -2° do 18° .

7.3.4 Antény

Dále je potřeba nastavit jednotlivé antény, které stanice používá. Jednotlivé frekvence a paprsky se nastavují přes reference, stejně jako pozice.

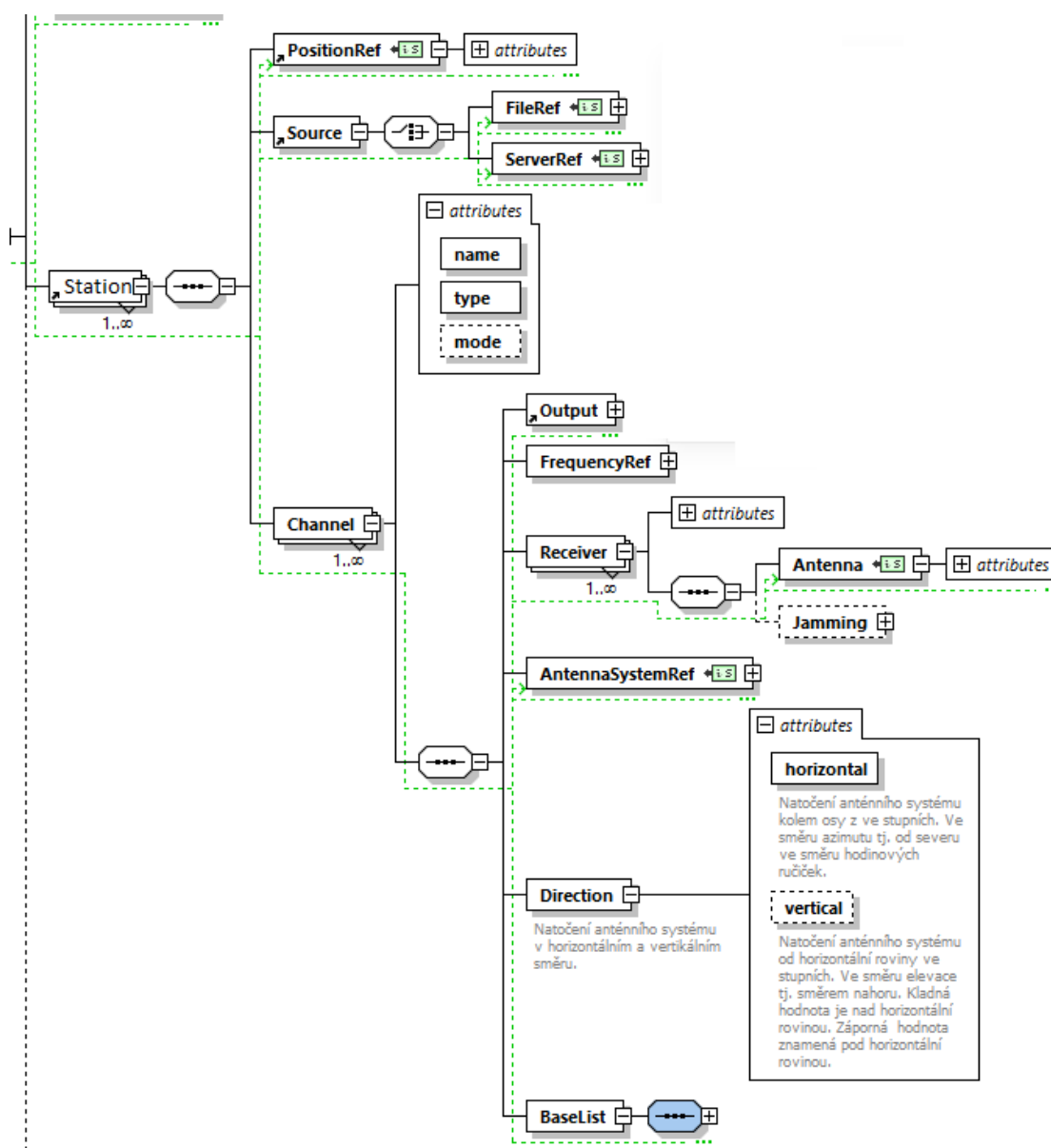


Obrázek 28: Antény

Toto nastavení pomůže určit, zda je signál vůbec schopna stanice zachytit. Opět pouze definujeme, stejně jako doposud, možné konfigurace.

7.3.5 Stanice

Jako poslední je finální nastavení stanice. Zde definujeme, jaké nastavení bude program používat. Jednotlivé nastavení probíhá čistě přes reference na již vytvořené objekty.



Obrázek 29: Lokátor

V souboru můžeme mít předpřipraveno několik možných nastavení a měnit jej pouze podle referenčních unikátních *id*.

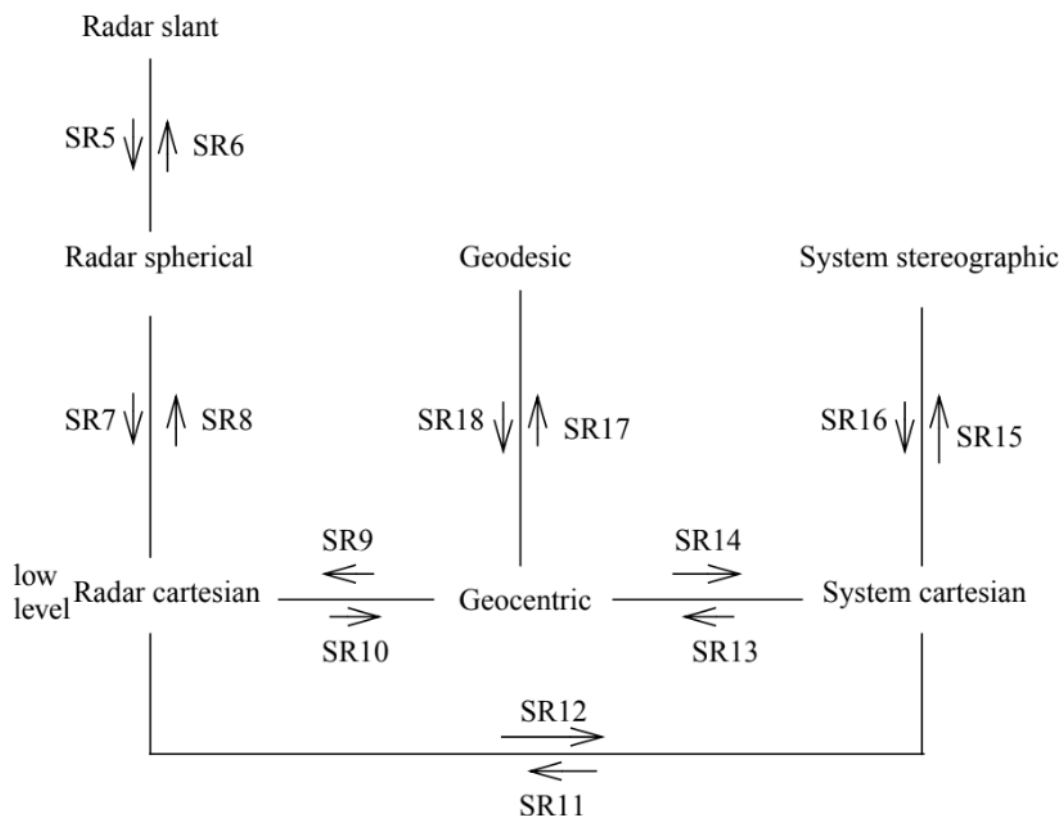
7.4 Aplikace

Jak již bylo zmíněno, aplikace je implementovaná v Javě za použití technologie Spring framework. Logika aplikace je rozdělená do několika Java balíčků.

7.4.1 Balíček coordinates

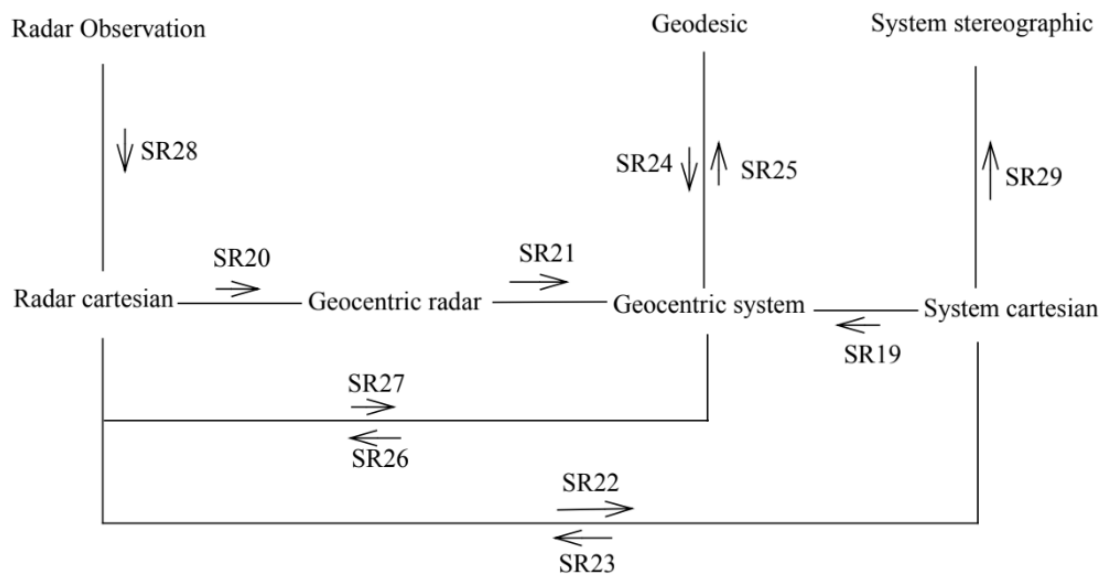
Tento balíček je rozdělený na dva menší balíčky. Jeden s názvem *basic* a druhý *auxiliary*. Balíček *basic* obsahuje třídy, které reprezentují jednotlivé typy souřadnic. Najdeme zde také pomocné třídy na 3D vektor a elipsoid. Nejdůležitější třída je třída *Transformation*, která slouží k převodu mezi jednotlivými typy souřadnic.

Na následujících obrázcích najdeme různé možnosti, jak souřadnice převádět.



Obrázek 30: ARTAS transformace [6]

Zde vidíme možné převody souřadnic pro ARTAS systém. Na dalším obrázku najdeme transformace NLR.



Obrázek 31: NLR transformace [6]

Jednotlivé kódy **SR_x** symbolizují převodní metodu, která není součástí této práce a můžeme ji vynechat. Tato práce pouze využívá již implementované převodní metody.

Druhý balíček, *auxiliary*, poskytuje doplňující třídy. Najdeme zde třídy symbolizující úhly, vzdálenosti a jednotky.

7.4.2 Balíček jaxb

Tento balíček obsahuje třídu *SimpleSerializer*, která má za úkol načíst konfigurační soubor XML. Tento soubor se validuje oproti námi vytvořenému XSD.

7.4.3 Balíček utils

Slouží pro uchování pomocných tříd. Třída *CircularFifoQueue* slouží pro implementaci fronty. Fronta má fixní nastavení velikosti, nahradí tedy nejstarší prvek, pokud je fronta plná. Pořadí iterace a odstraňování položek je založeno na pořadí vkládání. Prvky jsou odstraněny v pořadí, ve kterém byly vkládány.

Druhá třída, která stojí za zmínku, je *AutocorrelatedGaussianGenerator*. Tato třída slouží pro vygenerování korelované Gaussovy křivky, pro zatížení dat chybou.

7.4.4 Balíček locator

Jedná se o stěžejní balíček, slouží pro konfiguraci celé aplikace. Najdeme zde třídy, které slouží pro načtení XML souboru do logických celků a tříd pro snadnou manipulaci s daty dále.

Po načtení dat do tříd se naplní a vytvoří komplexnější třídy, které už obsahují i logiku, a ne pouze poskytování dat.

7.4.5 Balíček jama

Pro potřeby aplikace byl taky využit balíček třetí strany. Konkrétně se jedná o balíček JAMA: A Java Matrix Package.

Jedná se o základní balíček lineární algebry pro Javu. Poskytuje nám uživatelské úrovně pro manipulaci a konstrukci s reálnými hustými maticemi. Cílem je tedy poskytnout potřebnou funkčnost pro rutinní problémy i neprofesionálům v dané problematice. Jednoduchá implementace referencí pro veřejnou doménu byla vyvinuta MathWorks a NIST.

Skládá se ze šesti tříd: *Matrix*, *CholeskyDecomposition*, *LUDecomposition*, *QRDecomposition*, *SingularValueDecomposition* a *EigenvalueDecomposition*. Třída *Matrix* poskytuje základní operace algebry. *Cholesky* poskytuje rozložení symetrických, pozitivně definovaných matic. *LUDecomposition* rozklad pravoúhlých matic. Do budoucna se počítá s rozšířením o komplexní matice.

Oznámení o autorských právech: „Software je kooperativní produkt společnosti The MathWorks a Národního institutu pro standardy a technologie (NIST), který byl zveřejněn. MathWorks ani NIST nepřebírají žádnou odpovědnost za své užívání jinými stranami a neposkytují žádné záruky, vyjádřené ani předpokládané, o jeho kvalitě, spolehlivosti nebo jiných vlastnostech.“.

7.4.6 Zkreslení chybou

Výstupní data je v některých případech nutné zkreslit chybou. K tomu slouží servisní třída s názvem *DistortionService*, která to má za úkol.

Třída nabízí zkreslení úhlových i zeměpisných souřadnic. V případě zkreslení úhlových souřadnic vezme z konfiguračního souboru možný rozsah antény a z nastavení chybovosti vezme rozsah chyby a náhodnou hodnotou z tohoto rozsahu zkreslí úhel. Zkreslí jak azimut, tak elevaci.

Ve druhém případě využívá principu pevné chyby. Tedy spočítá si vzdálenost stanice od polohy odpovídače. Vypočítanou vzdálenost ještě podle daných pravidel zkreslí a použije k vypočítání chyby. V konfiguračním souboru je nastavená maximální vzdálenost a maximální chyba této vzdálenosti. Chyba se vypočítá poměrem k maximu. Následně se tato chyba aplikuje na dané souřadnice a pošle na výstup.

Ve třetím případě žádnou chybu nevypočítává. Pouze opět z konfigurace načte pravděpodobnost příjmu a na základě této pravděpodobnosti zprávu zahodí. Než dojde k pravděpodobnosti, ověří se, že zpráva přichází ze směru antény, jinak se zahodí. Na výstupu se pak objeví prázdná zpráva, aby bylo možné ověřit fungování.

7.4.7 Instalace aplikace

Pro instalaci či build aplikace, je využíván nástroj **Apache Maven**. Ačkoliv je možné využít tento nástroj pro různé projekty psané v různých programovacích jazycích, podporován je především jazyk Java, což přináší velké výhody.

Název *maven* znamená *znalec*. Maven byl vytvořen pro zjednodušení buildů pro projekt Jakarta Turbine. Hlavní důvod pro vznik byla snaha o standardizaci a znovu použitelnosti buildovacích skriptů.

U každého projektu lze nadefinovat závislosti na externích knihovnách a nastavení aplikace. U Mavenu se toto děje v souboru pom.xml.

Celou aplikaci tedy můžeme zbuildovat jednoduchým příkazem. Ten vytvoří nový balíček s aplikací, původní (pokud existuje) přejmenuje na zálohu. Zkompilované třídy a výsledný balíček potom najdeme ve složce **target**. Příkaz, který celý proces spustí:

```
mvn clean package
```

Tento příkaz přeloží aplikaci a při použití Javy, se podle nastavení vytvoří soubor **war** nebo **jar**, což je balíček, který obsahuje celou aplikaci. Tento příkaz je dostupný při použití Mavenu, nejčastěji na Linuxu. V případě použití na Windows se používá *Wrapper* a příkaz je pak následujícího tvaru:

```
mvnw clean package
```

7.4.8 Spuštění aplikace

Spuštění aplikace probíhá ve třídě *LocatorApplication*. Nejprve se načte odpovídající soubor XML a poté se spustí samostatné vlákno aplikace. V případě chyby se chyba vypíše a aplikace se ukončí.

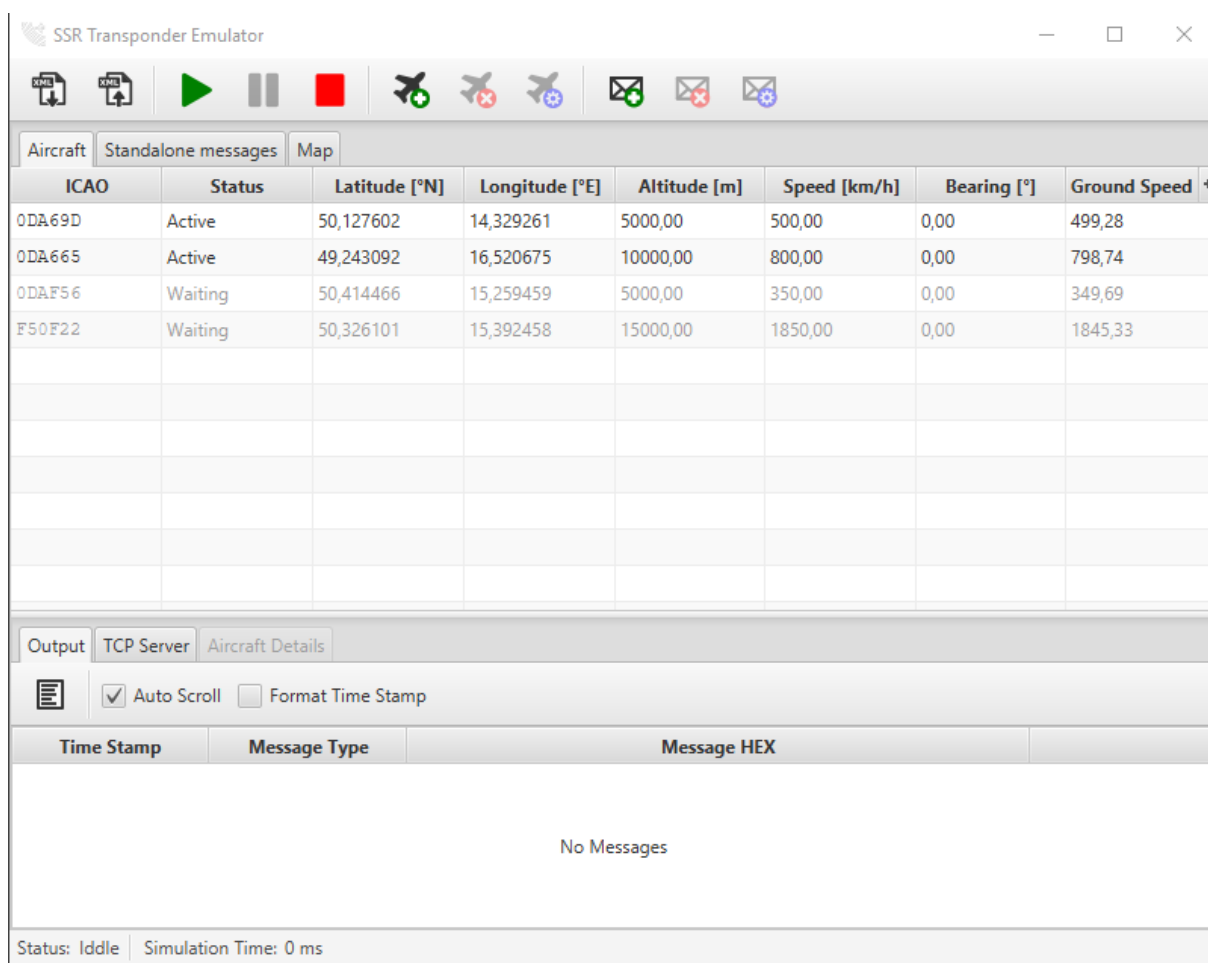
Aplikace se spouští pomocí příkazového řádku. Pokud se při spuštění nezadá žádný parametr, dojde k vybrání konfiguračního souboru uloženého ve složce **resources** s názvem **config.xml**. Zde se také nachází základní soubor.

Druhá možnost je zadat do parametru cestu k jinému souboru a následně bude tento soubor vybrán jako konfigurační.

7.5 Vstupní data

Aplikace využívá zdrojových dat, které zpracovává, z aplikace od Ing. Moravčíka. Výstup programu z jeho diplomové práce použijeme jako vstup pro naši aplikaci.

V jeho aplikaci SSR Transponder Emulator si spustíme vytvořený scénář a zprávy ve formátu XML budeme zachytávat a dále zpracovávat.



Obrázek 32: Aplikace SSR Transponder Emulator [12]

Výstup aplikace je buď textový. Může taky sloužit jako TCP server a odesílá zprávy všem přihlášeným aplikacím na daném portu. Ukázka výstupu:

Time Stamp	Message Type	Message HEX
35	Airborne Position	8D0DA69D5957616B21066E26915F
46	Airborne Velocity	8D0DA6659914C330600481AB249E
282	Airborne Velocity	8D0DA69D99110E83C034030B0157
394	Airborne Position	8D0DA66558A990D44B94544154B6
481	Airborne Velocity	8D0DA6659914C33060048254CC8C
482	Airborne Position	8D0DA69D595764DC8AF218727A56
731	Airborne Velocity	8D0DA69D99110E83C034030B0157
919	Airborne Position	8D0DA66558A994485D7CCB3E93E6
1035	Airborne Position	8D0DA69D5957616B1F068706329D
1046	Airborne Velocity	8D0DA6659914C330600483AB3885
1282	Airborne Velocity	8D0DA69D99110D83C034030E1E05

Obrázek 33: Výstup zdrojových dat [12]

Tento výstup zasílá zprávu zakódovanou v HEX. Jestliže přepneme na XML verzi, dostaneme pro uživatele více srozumitelnou zprávu.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Message
  ICAO="0DA69D"
  messageType="AIRBORNE_POSITION"
  sourceAltitude="5009.433603550393"
  sourceLatitude="50.127292028752976"
  sourceLongitude="14.333754309185224"
  timestamp="2334000000"
>
  8D0DA69D5957A16B0F06AB62DFEE
</Message>
```

Vidíme, že o původní HEX zprávu jsme nepřišli, ale již ji nemusíme dekodovat a jednotlivé atributy jsou rovnou k dispozici. Nejsou tedy k dispozici úplně všechny, ale jen ty, co potřebujeme.

Pokud nakonfigurujeme naši aplikaci, aby přijímala zprávy z TCP serveru a nastavíme spojení na tuto aplikaci a obě spustíme, bude emulátor generovat výstupní zprávy.

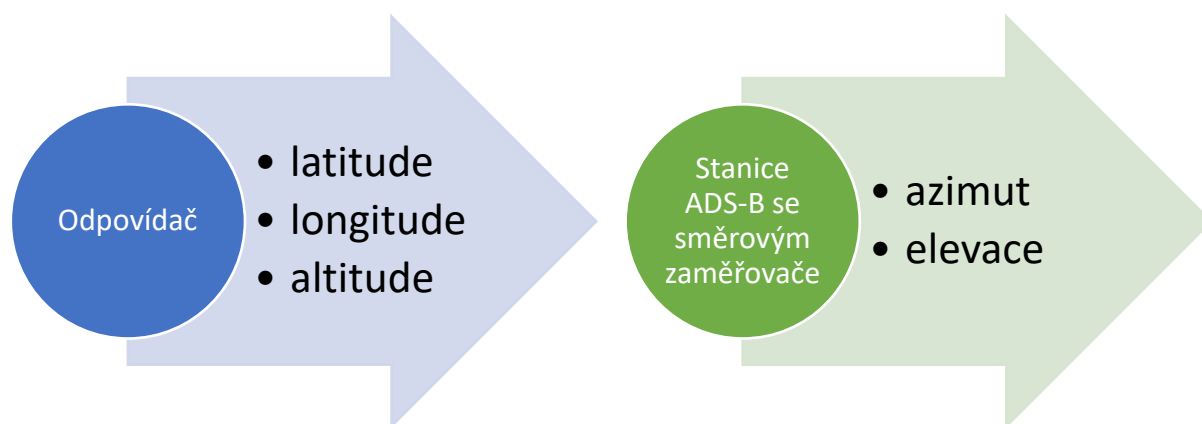
7.6 Výstupní zprávy

Jakou stanici má aplikace emulovat se nastaví v konfiguračním souboru. Element *Station* nabízí atribut *type*, do kterého lze zadat následující typ:

- MTL – pro stanici s pasivním multilateračním systémem,
- ADSB – pro stanici se směrovými anténami,
- Locator – pro stanici se směrovým zaměřovačem.

7.6.1 ADS-B se směrovým zaměřovačem

Tato stanice vydává zprávu, která obsahuje azimut a elevaci. Úhly jsou zatíženy chybou. Rozsah a rozložení chyb odpovídá reálným hodnotám reálných zaměřovačů. Tato stanice nezjistí přesnou polohu cíle, jen směr.



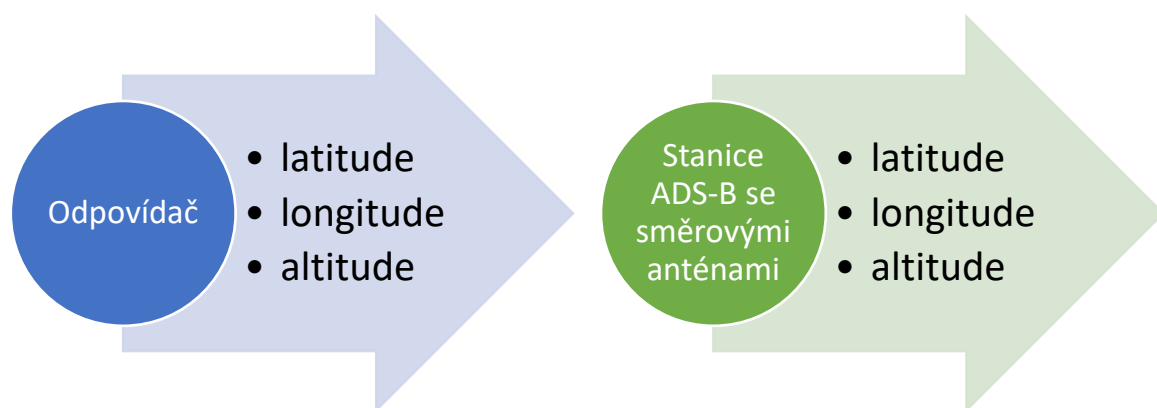
Obrázek 34: Stanice ADS-B se směrovým zaměřovačem

Výstupní XML zpráva má tedy následující tvar. Azimut je úhel, který ve vodorovné rovině svírá směr se Severním nebo Jižním pólem. Slouží tedy k určení polohy pozorovaného objektu. Elevace potom vyjadřuje úhel ve významu zdvihání či vyzdvihování.

```
<Message  
  azimuth="2.6003506952393485"  
  elevation="0.08405494288348858"  
>
```

7.6.2 ADS-B se směrovými anténami

Stanice vezme zprávu od odpovídače a vystaví ji na výstup. Nepochází zde k žádnému dalšímu zatížení chybou. Stanice pouze vyfiltruje některé zprávy na základě pravděpodobnosti příjmu a filtru na směr antény.



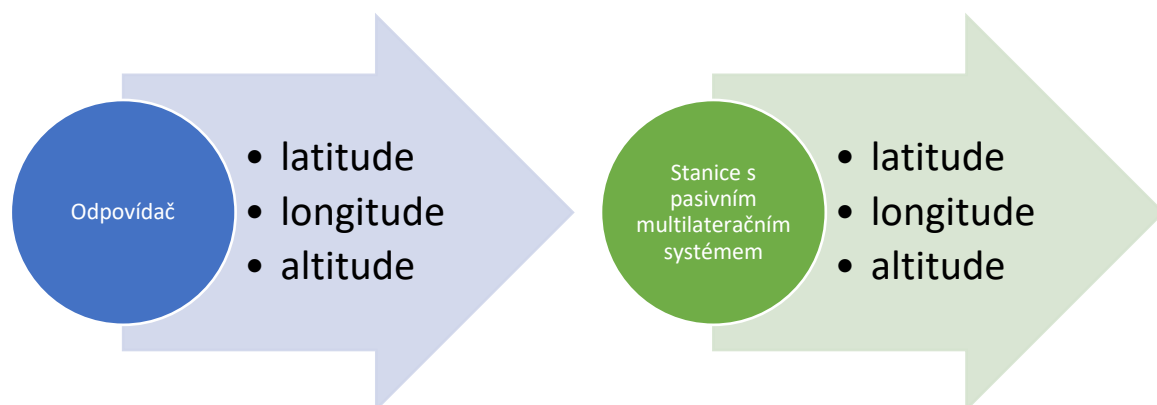
Obrázek 35: Stanice ADS-B se směrovými anténami

Pravděpodobnost příjmu představuje výpadky signálů. Jedná se o zadanou hodnotu, se kterou zpráva projde stanicí a nebude zahozena. Tuto hodnotu lze nastavit v konfiguračním souboru. Filtrování na směr potom určí, z nastavení antény, zda se daná poloha nachází v rozsahu antény. Pokud ne, je zpráva automaticky zahozena a nejde na výstup. Pokud splňuje rozsah antény, je stanicí vystavena dál.

```
<Message
  longitude="14.335297062899912"
  latitude="50.12718556075936"
  altitude="287204.9845800409"
/>
```

7.6.3 Pasivní multilaterační systém

Výstup stanice je opět zatížen chybou. Jelikož známe přesnou polohu cíle, můžeme zde uplatnit princip pevné chyby. Chybu získáme z matice chyb po určení vzdálenosti mezi objektem a stanicí.



Obrázek 36: Stanice s pasivním multilateračním systémem

Výstupní zpráva tedy opět obsahuje zeměpisné souřadnice, avšak navíc zatížené další chybou. Nedochází zde k žádné filtraci na směr.

```
<Message  
  longitude="14.334178567139107"  
  latitude="50.127262752098034"  
  altitude="287070.4381044866"  
>
```

ZÁVĚR

V úvodní části byly představeny základy radiolokace se základními principy fungování radiolokace. Tato práce je zaměřená na aplikaci emulátoru pozemních stanic. Protože autor této práce studoval informační technologie, bylo nutné dostudovat tyto informace a základní principy shromáždit pro lepší pochopení. To přineslo veliký přínos v rozšíření do jiné oblasti vědomostí a pomohlo to k návrhu a realizaci dané aplikace.

Byly představeny primární a sekundární radary s odkazem na základní principy v první části. Jejich rozdíly a vzájemné doplňování a problematika.

V další části je vysvětlena problematika chyb a měření. Je zde přínos pochopení modelování chyb a jejich umělé vytváření. Postupy, jak lze chybou zatížit data, aby co nejvíce odpovídali realitě a nebyly moc umělé. Celkové pochopení přesnosti a správnosti dat. Důraz na bezpečnostní rizika.

V praktické části byl navržen emulátor pozemních stanic na platformě Java za využití frameworku Spring. Při vývoji byly dodržovány principy objektově orientovaného programování s důrazem na snadnou rozšiřitelnost a udržovanost.

Výstupem jsou jednotlivé zprávy ve formátu XML. Konverze do formátu Asterix není součástí této práce. Zpráva XML obsahuje jednotlivé informace o poloze nebo úhlech. Je možné zprávy vypisovat či jej poskytovat dále TCP/IP protokolem. Aplikaci lze tedy nakonfigurovat zároveň jako TCP klienta i jako server.

Tato aplikace společně s dalšími bude tvořit systém pro validaci ADS-B zpráv. Umožní jednodušší vývoj algoritmů na to zaměřených. Usnadní daný vývoj a plně nahradí hardware, a tudíž sníží náklady.

Možným cílem pro další vývoj je přidělení grafického rozhraní, které by mohlo spravovat více instancí emulátoru s různými nebo stejnými konfiguracemi. Sledování jejich činnosti a vytížení.

POUŽITÁ LITERATURA

- [1] WOLF, Christian. *Radars Accuracy. Radar basics* [online]. [cit. 2019-02-03]. Dostupné z: <http://www.radartutorial.eu/01.basics/Radars%20Accuracy.en.html>
- [2] SOLAŘ, Erik. *Systém multilaterace*. VŠB-TU Ostrava, 2018. Seminární práce. VŠB-TU
- [3] ZÁZVORKA, Jan. *Přesnost určení polohy v multilateračních systémech*. Praha, 2016. Bakalářská práce. ČVUT.
- [4] ŽIHLA, Zdeněk. *Technologie a řízení letecké dopravy*. Pardubice: Univerzita Pardubice, 2000. ISBN 80-7194-291-X.
- [5] BELLMAN, Richard. *Dynamic programming*. Dover ed. Mineola, N.Y.: Dover Publications, 2003. ISBN 0-486-42809-5.
- [6] MAGNI, L. *Software Requirements Document TransLib*. 4. 1999.
- [7] BOTHA, Roelf Cornelis. *Averaging, Errors and Uncertainty* [online]. 2016 [cit. 2019-06-29]. Dostupné z: http://avntraining.hartrao.ac.za/images/Error_Analysis.pdf
- [8] SATISH C., Satish C. a Wang GANGHUI. *Modeling ADS-B Position and Velocity Errors for Airborne Merging and Spacing in Interval Management Application*.
- [9] PŘIBYL, David. *Radary, současně používané systémy*. 2005. Semestrální práce. ČVUT.
- [10] BEZOUŠEK, Pavel a Pavel ŠEDIVÝ. *Radarová technika*. 2004.
- [11] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [12] MORAVČÍK, David. *Emulátor letadlového odpovídače SSR*. Pardubice, 2018. Diplomová práce. Univerzita Pardubice.

PŘÍLOHY

Příloha A – Konfigurační soubor	72
---------------------------------------	----

PŘÍLOHA A – KONFIGURAČNÍ SOUBOR

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LocatorConfiguration.xsd">
  <Verbose value="008e"/>
  <Files>
    <File id="SourceFile01" name="xml/test01.xml"/>
    <File id="OutputFile01" name="Output01.xml"/>
  </Files>
  <Servers>
    <Server id="inputTcp" ip="127.0.0.1" port="36016"/>
    <Server id="outputTcp" ip="127.0.0.1" port="36026"/>
  </Servers>
  <SpeedLight value="299792458.0"/>
  <Frequencies>
    <Frequency id="F1090MHz" value="1090e6"/>
    <Frequency id="F1060MHz" value="1060e6"/>
  </Frequencies>
  <Positions>
    <Position id="PositionLocator">
      <Geodesic latitude="50.065781" longitude="15.768515" alti-
tude="222.0"/>
    </Position>
    <Position id="PositionHorizontalAntennaSystem">
      <Geodesic latitude="50.065781" longitude="15.768515" alti-
tude="222.0"/>
    </Position>
    <Position id="PositionVerticalAntennaSystem">
      <Geodesic latitude="50.065781" longitude="15.768515" alti-
tude="222.0"/>
    </Position>
  </Positions>
  <Beams>
    <Beam id="BeamID1">
      <HorizontalSector>
        <Symetric width="30"/>
        <Range min="10" max="300000"/>
      </HorizontalSector>
      <VerticalSector>
        <Asymetric start="-1" end="40"/>
        <Range min="10" max="300000"/>
      </VerticalSector>
    </Beam>
    <Beam id="BeamID2">
      <HorizontalSector>
        <Symetric width="45"/>
        <Range min="10" max="300000"/>
      </HorizontalSector>
      <VerticalSector>
        <Asymetric start="-1" end="40"/>
        <Range min="10" max="300000"/>
      </VerticalSector>
    </Beam>
  </Beams>
  <AntennaTypes>
    <AntennaType id="Type01">
      <Name>AntennaType 1</Name>
      <FrequencyRef idRef="F1090MHz"/>
      <BeamRef idRef="BeamID1"/>
    </AntennaType>
  </AntennaTypes>
</Configuration>
```



```

<AntennaType id="Type02">
  <Name>AntennaType 2</Name>
  <BeamRef idRef="BeamID1"/>
</AntennaType>
<AntennaType id="Type03">
  <Name>AntennaType 2</Name>
  <BeamRef idRef="BeamID2"/>
</AntennaType>
</AntennaTypes>
<AntennaSystems>
  <AntennaSystem id="AS01" type="horizontal">
    <PositionRef idRef="PositionLocator"/>
    <Antenna id="AH0">
      <AntennaTypeRef idRef="Type01"/>
      <LocalPosition x="0.0" y="0.0" z="0.0"/>
    </Antenna>
    <Antenna id="AH1">
      <AntennaTypeRef idRef="Type01"/>
      <LocalPosition x="0.550077904587156" y="0.0" z="0.0"/>
    </Antenna>
    <Antenna id="AH2">
      <AntennaTypeRef idRef="Type01"/>
      <LocalPosition x="1.32018697100917" y="0.0" z="0.0"/>
    </Antenna>
    <Antenna id="AH3">
      <AntennaTypeRef idRef="Type01"/>
      <LocalPosition x="8.25116856880734" y="0.0" z="0.0"/>
    </Antenna>
    <BeamRef idRef="BeamID1"/>
  </AntennaSystem>
  <AntennaSystem id="AS02" type="vertical">
    <PositionRef idRef="PositionLocator"/>
    <Antenna id="AV0">
      <AntennaTypeRef idRef="Type02"/>
      <LocalPosition x="0.0" y="0.0" z="0.0"/>
    </Antenna>
    <Antenna id="AV1">
      <AntennaTypeRef idRef="Type02"/>
      <LocalPosition x="0.0" y="0.0" z="0.550077904587156"/>
    </Antenna>
    <Antenna id="AV2">
      <AntennaTypeRef idRef="Type02"/>
      <LocalPosition x="0.0" y="0.0" z="1.32018697100917"/>
    </Antenna>
    <Antenna id="AV3">
      <AntennaTypeRef idRef="Type02"/>
      <LocalPosition x="0.0" y="0.0" z="8.25116856880734"/>
    </Antenna>
    <BeamRef idRef="BeamID2"/>
  </AntennaSystem>
</AntennaSystems>
<Station type="MTL">
  <PositionRef idRef="PositionLocator"/>
  <Source>
    <ServerRef idRef="inputTcp" />
  </Source>
  <Channel name="Horizontal channel" type="horizontal">
    <Output>
      <FileRef idRef="OutputFile01"/>
    </Output>
  </Channel>
</Station>

```

```

<FrequencyRef idRef="F1090MHz"/>
<Receiver name="R1">
  <Antenna idRef="AH0"/>
  <Jamming>
    <Noise min="-1" max="1" skew="2" bias="3"/>
    <Outage value="0.90"/>
    <Jump max="0.05" frequency="1090.0"/>
  </Jamming>
</Receiver>
<Receiver name="R2">
  <Antenna idRef="AH1"/>
  <Jamming>
  </Jamming>
</Receiver>
<Receiver name="R3">
  <Antenna idRef="AH2"/>
  <Jamming>
  </Jamming>
</Receiver>
<AntennaSystemRef idRef="AS01"/>
<Direction horizontal="-85" vertical="0"/>
<BaseList>
  <Base name="A0A1" antenna0="AH0" antennaN="AH1"/>
  <Base name="A0A2" antenna0="AH0" antennaN="AH2"/>
  <Base name="A0A3" antenna0="AH0" antennaN="AH3"/>
</BaseList>
</Channel>
<Channel name="Vertical Channel" type="vertical">
  <Output>
    <FileRef idRef="OutputFile01"/>
  </Output>
  <FrequencyRef idRef="F1090MHz"/>
  <Receiver name="R1">
    <Antenna idRef="AV0"/>
  </Receiver>
  <Receiver name="R2">
    <Antenna idRef="AV1"/>
  </Receiver>
  <Receiver name="R3">
    <Antenna idRef="AV2"/>
  </Receiver>
  <Receiver name="R4">
    <Antenna idRef="AV3"/>
  </Receiver>
  <AntennaSystemRef idRef="AS02"/>
  <Direction horizontal="0.0" vertical="0.0"/>
  <BaseList>
    <Base name="A0A1" antenna0="AV0" antennaN="AV1"/>
    <Base name="A0A2" antenna0="AV0" antennaN="AV2"/>
    <Base name="A0A3" antenna0="AV0" antennaN="AV3"/>
  </BaseList>
</Channel>
</Station>
<RadarMap range="500000" tickUnit="3.14159265358979" minWidth="1.0"
  minHeight="1.0"/>
</Configuration>

```