



ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Fakulta elektrotechnická

Katedra elektroniky a informačních technologií

BAKALÁŘSKÁ PRÁCE

Ukázka kooperativního příjmu rádiových signálů pomocí sítě softwarově
definovaných přijímačů

Autor práce: Martin Tomášek

Vedoucí práce: Ing. Ivo Veřtát, Ph.D.

Plzeň 2022

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická
Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Martin TOMÁŠEK**
Osobní číslo: **E18B0036P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a telekomunikace**
Téma práce: **Ukázka kooperativního příjmu rádiových signálů pomocí sítě softwarově definovaných přijímačů**
Zadávající katedra: **Katedra elektroniky a informačních technologií**

Zásady pro vypracování

1. Popište základní princip kooperativního příjmu rádiových signálů.
2. Pomocí programu SDRSharp a sítě softwarově definovaných přijímačů SpyServer realizujte paralelně záznam přijatého signálu vhodného zdroje vysílání (APRS pakety, meteorologické sondy, radioamatérské satelity).
3. Záznamy zpracujte v Matlabu, vyhodnoťte je separátně i v kooperativním příjmu.
4. Zhodnoťte experiment z hlediska možnosti využití kooperativního příjmu pro zachytávání telemetrických signálů malých satelitů.

Rozsah bakalářské práce: **30 – 40**
Rozsah grafických prací: **dle doporučení vedoucího**
Forma zpracování bakalářské práce: **elektronická**

Seznam doporučené literatury:

1. J McGehee. Multi-site spatially diverse demodulation of HF propagated signals. Conference: Military Communications Conference, 2008.

Vedoucí bakalářské práce: **Ing. Ivo Veřtát, Ph.D.**
Katedra elektroniky a informačních technologií

Datum zadání bakalářské práce: **8. října 2021**
Termín odevzdání bakalářské práce: **26. května 2022**


Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Doc. Ing. Jiří Hammerbauer, Ph.D.
vedoucí katedry

V Plzni dne 8. října 2021

Abstrakt

Tato bakalářská práce se zabývá paralelním příjmem signálu z vhodně zvoleného zdroje vysílání a následným vyhodnocením získaných dat, jež jsou opatřena za použití softwarově definovaného rádia. Úvod obsahuje informace k datové komunikaci různých zdrojů signálu uvedených v zadání práce. Po zvolení radioamatérských satelitů jako vhodného zdroje příjmu jsou dále popsány datové rámce obsaženy ve vysílání satelitů Lucky-7 a VZLUSAT-2. Za účelem zachycení vysílání byla zprovozněna i pozemní stanice. Práce pokračuje detailním popisem skriptu pro vyhodnocení signálu vytvořeného v prostředí MATLAB. Z důvodu zachování splnitelnosti zadání nebyla vyhodnocována samotná komunikace satelitu s pozemní stanicí pro její komplexní datový tok a následně obtížnou manipulaci se signálem. Byly proto použity záznamy majáku satelitu, které obsahují podstatně méně dat. Konec práce zobrazuje výstup programu, který demonstruje několik základních metod pro vyhodnocení signálů při společném příjmu. V závěru jsou zhodnoceny výsledky práce a přínos metody kooperativního příjmu.

Klíčová slova

SDR, CubeSat, diverzitní příjem, měkké a tvrdé rozhodování, VZLUSAT-2, Lucky-7

Abstract

Tomášek, Martin. *Treatise on spatially diverse receivment of radio signals via network of software defined radios [Ukázka kooperativního příjmu rádiových signálů pomocí sítě softwarově definovaných přijímačů]*. Pilsen, 2022. Bachelor thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Electronics and Information Technologies. Supervisor: Ivo Veřtát

This bachelor thesis aims to demonstrate spatially diverse receivment of radio signal via network of software defined radios. Introduction contains information concerning all available sources of broadcast stated in work tasks. After the radioamateur sattelites being chosen as usable source of broadcast, details of data structures of VZLUSAT-2 and Lucky-7 are stated. The ground station was created to obtain these data. Thesis then describes signal evaluation script, that was created in MATLAB. Because the satellites' communication mode data structure is very complex and hard to manipulate with, we chose its beacon as a reference for evaluation script. End chapter then provides numerous plots of cooperative recieved signals that undergone data evaluation by basic methods. Conclusion chapter then discusses results of this work along with benefits of cooperative diversity.

Keywords

SDR, CubeSat, cooperative diversity, soft/hard desicion making, VZLUSAT-2, Lucky-7

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 26. května 2022

Martin Tomášek

.....

Podpis

Poděkování

Tímto bych rád poděkoval vedoucímu mé bakalářské práce Ing. Ivu Veřtátovi, Ph.D za cenné rady a metodické vedení práce.

Dále bych chtěl poděkovat komunitě, která se podílí na vývoji projektu SatNOGS, za výborný nástroj, jež významně napomohl k realizaci této práce.

Obsah

Seznam obrázků	viii
Seznam tabulek	ix
Seznam symbolů a zkratek	x
1 Úvod	1
2 Kooperativní příjem	2
2.1 Princip kooperativního příjmu	2
2.2 Softwarově definované rádio	3
2.2.1 Charakterizace SDR	3
2.2.2 Sféry uplatnění	3
2.2.3 Použité SDR	4
3 Popis možných zdrojů signálu	5
3.1 APRS pakety	5
3.2 Meteorologické satelity	6
3.2.1 Geostacionární orbit	6
3.2.2 Polární orbit	6
3.2.3 Příjem signálu	6
3.3 Radioamatérské satelity	7
4 Realizace kooperativního příjmu signálu	8
4.1 Příjem pomocí sítě SpyServer	8
4.2 Příjem pomocí webových rádií	9
4.3 Příjem pomocí sítě SatNOGS	9
4.3.1 Vytvoření pozemní stanice	10
4.3.2 Instalace a nastavení klienta	11
4.3.3 Výběr družice	14
5 Zpracování získaných dat	16
5.1 Datový zásobník (buffer)	16
5.2 Příprava signálu pro zpracování	18

5.2.1	Funkce finddelay	18
5.2.2	Funkce circshift	19
5.2.3	Vektor průběhu preamble	19
5.2.4	Časové zarovnání signálů	20
5.3	Demodulace signálu	21
5.3.1	Čtení datového zásobníku	21
5.3.2	Bitové vyhodnocení	22
5.3.3	Výstup demodulace	23
5.4	Dílčí metody vyhodnocení	23
5.4.1	Paralelní výpis	23
5.4.2	Kombinace bitů majoritou	24
5.4.3	Kombinace signálu před demodulací s vyhodnocením po demodulaci	25
6	Závěr	27
	Reference, použitá literatura	29
	Přílohy	32
A	Použité skripty, zdrojové kódy	32
A.1	posundemod.m	32
B	Spektrogramy zkušebních observací	35
C	Spektrogramy pro kooperativní příjem VZLUSAT-2	38

Seznam obrázků

2.1	Model prostorové diverzity kooperativního příjmu. Vytvořeno dle [1].	2
2.2	Blokové schéma RTL-SDR. Zkratky: LNA - Nízkošumový předzesilovač, VGA - Zesilovač s proměnným ziskem, LO - Lokální oscilátor, LPF - Dolnofrekvenční propust. Převzato z [2].	3
2.3	Různé produkty Airspy. Použité SDR, Airspy R2, je uprostřed. Převzato z [8].	4
3.1	Detail synchronizačního úseku vysílání APT. Převzato z [12].	7
3.2	Detaily komunikace satelitu Lucky-7. Převzato z [19].	7
4.1	Hlavní obrazovka aplikace SDRSharp s probíhajícím poslechem FM rádia .	8
4.2	Různé možnosti provedení pozemní stanice. Převzato z [14].	10
4.3	Detaily vlastní pozemní stanice ve webovém prostředí SatNOGS.	11
4.4	Hlavní nabídka SatNOGS klienta, který operuje na naší pozemní stanici. .	11
4.5	Výpis příkazu pro kontrolu funkčnosti SDR ovladače. Chyba na začátku výpisu pouze upozorňuje na duplikátní soubor, což nemá vliv na výslednou funkci.	13
4.6	Výpis je dle verifikace instalace ^[16] správný, stanice je tedy provozuschopná.	13
4.7	Vybrané observace z databáze SatNOGS pro satelit VZLUSAT-2.	15
5.1	Časově nezarovnané signály majáku satelitu VZLUSAT-2 získané metodou kooperativního příjmu. Je zřejmá přítomnost preambule následovaná daty.	18
5.2	Ideální průběh preambule zarovnaný s preambulí obsaženou v signálu. . . .	20
5.3	Časově zarovnané signály majáku VZLUSAT-2 získané pomocí kooperativního příjmu.	21
5.4	Paralelně vyhodnocené nahrávky majáku satelitu VZLUSAT-2.	24
5.5	Porovnání nahrávek komunikace majáku VZLUSAT-2 s vyhodnoceným signálem použitím metody kombinace bitů majoritou.	25
5.6	Porovnání nahrávek komunikace majáku VZLUSAT-2 s vyhodnoceným signálem použitím metody kombinace signálu před demodulací s vyhodnocením po demodulaci.	26

B.1	Spektrogram satelitu GRBAlpha zachycený pozemní stanicí 2405-PlzenFEL	
	Číslo observace: 4882705, převzato z [23]	36
B.2	Spektrogram satelitu Lucky-7 zachycený pozemní stanicí 2405-PlzenFEL	
	Číslo observace: 5042953, převzato z [24]	37
C.1	Spektrogram satelitu VZLUSAT-2 zachycený pozemní stanicí 1675-MAUSyagi	
	Číslo observace: 5477367, převzato z [25]	39
C.2	Spektrogram satelitu VZLUSAT-2 zachycený pozemní stanicí 2134-MAUSyagi	
	Číslo observace: 5477368, převzato z [26]	40
C.3	Spektrogram satelitu VZLUSAT-2 zachycený pozemní stanicí 2510-MAUSyagi	
	Číslo observace: 5477369, převzato z [27]	41

Seznam tabulek

3.1	Struktura APRS paketu dle standardu AX.25. Převzato s úpravami. ^[10]	5
4.1	Tabulka popisující formát datového rámce na jehož počátku je 16 bytová hlavička, tedy preamble. Převzato s úpravami. ^[19]	14

Seznam symbolů a zkratek

SDR	Software Defined Radio. Softwarově definované rádio.
SNR	Signal to noise ratio. Poměr signálu k šumu.
VHF	Very high frequencies. Pásmo velmi krátkých vln.
UHF	Ultra high frequencies. Pásmo ultra krátkých vln.
IQ	Inphase Quadrature Diagram. Konstelační diagram.
ESD	Electrostatic-sensitive device. Součástka citlivá na elektrostatický výboj.
SMA	SubMiniature version A. Označení konektoru.
Samp	Sample. Vzorek.
APRS	Automatic Packet Reporting System. Komunikační protokol pro přenášení informace o okamžitém stavu objektu.
APT	Automatic Picture Transmission. Komunikační protokol pro přenášení snímků pořízenými satelitem.
NOAA	National Oceanic and Atmospheric Administration. Národní úřad pro oceán a atmosféru.
LNA	Low-noise amplifier. Nízkošumový předzesilovač.
MIX	Mixer. Směčovač.
VGA	Voltage Gain Amplifier. Zesilovač s proměnným ziskem.
CRC	Cyclic redundancy check. Cyklický redundantní součet.

1

Úvod

V telekomunikační technice je kladen velký důraz na nízkou chybovost přenosu. Metody kooperativního příjmu jsou proto běžně k nalezení v nespočtu technických aplikací. Obecně lze kooperativní příjem definovat, jako spolupráci několika přijímačů při příjmu požadovaného signálu. Každá zachycená kopie toho signálu je při průchodu přenosovým prostředím deformována, tím pádem může obsahovat i množství chyb. Pomocí vyhodnocovacích metod jsou kooperativně přijaté signály kombinovány do výsledného datového toku, ve kterém je chybovost již značně redukována.

Ve snaze demonstrovat kooperativní příjem bylo využito moderní technologie softwarově definovaných rádií. Softwarově založené radiokomunikační zařízení nabízí řadu možností a rozšíření oproti čistě hardwarové konstrukci. Pro jeho rozmanitost použití je také prozkoumáno několik technik, které umožní získat více kopií signálu kooperativním příjmem pro následné zpracování a vyhodnocení. Na základě zvoleného signálu a jeho datové struktury bude vypracován program, který jej pomocí základních metod vyhodnotí a vykreslí jeho výslednou podobu.

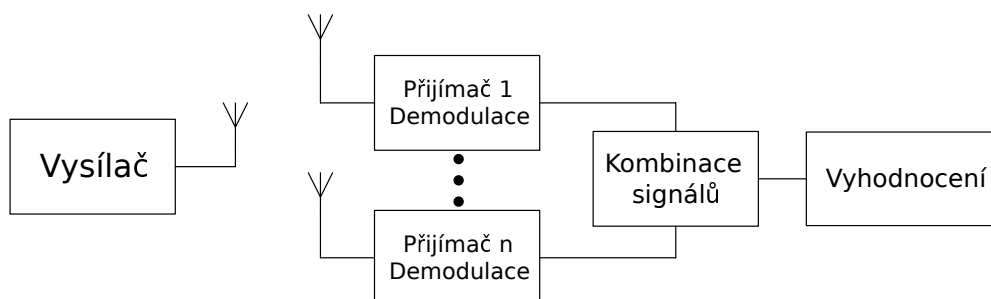
2

Kooperativní příjem

2.1 Princip kooperativního příjmu

Hlavní myšlenkou kooperativního příjmu je fakt, že kombinací několika, nezávisle zarušených, kopií přijatého signálu vzniká výsledný signál, jehož chybovost by měla být teoreticky menší, než chybovost obsažená v jeho individuálních replikách. Tato metoda příjmu má zajistit vyšší hodnoty SNR, tedy zvýšit kvalitu přijatého signálu a redukovat jeho chybovost.^[1] Obrázek 2.1 ukazuje princip kooperativního příjmu.

Kooperativní příjem lze dále dělit dle typu jeho diverzity. Prostorová (makro) diverzita například zapojuje do spolupráce několik antén které jsou od sebe fyzicky vzdáleny. U mikro diverzity jsou jednotlivé anténní prvky posazeny na stejnou nosnou konstrukci, každý z nich je potom natočen odlišným směrem. U obou metod jsou potom signály kombinovány a vyhodnoceny. Existuje několik dalších diverzních metod příjmu, některé lze i kombinovat.^[3] Tato práce bude využívat konkrétně diverzity prostorové.



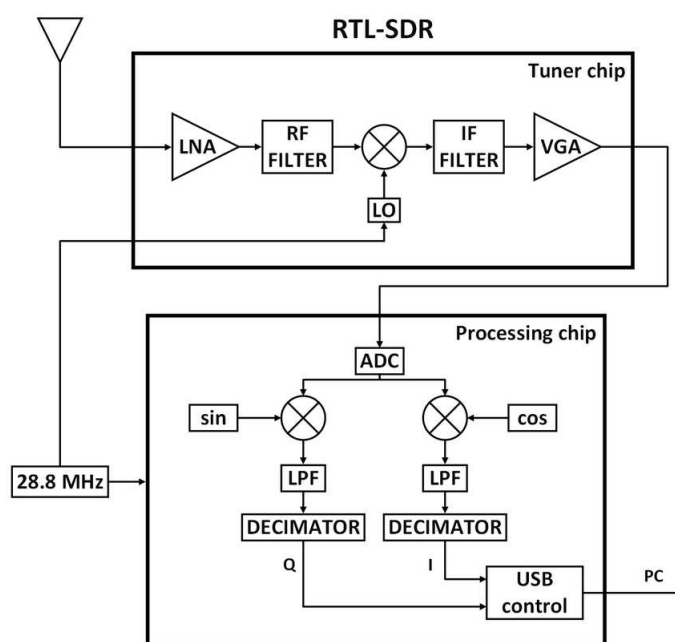
Obr. 2.1: Model prostorové diverzity kooperativního příjmu. Vytvořeno dle [1].

V minulosti byla realizace kooperativního příjmu obtížná. Mezi obstrukce patřila zejména omezená kapacita datových úložišť, málo precizní časování a synchronizace a také absence vysokorychlostních sítí. Pomocí dnešních technologií jsme však schopni tato omezení výrazně potlačit.^[1]

2.2 Softwarově definované rádio

2.2.1 Charakterizace SDR

Jedná se o moderní radiokomunikační technologii, kde zpracování signálu probíhá pomocí softwaru. Celá konstrukce radiového přijímače/vysílače se tak podstatně zjednodušuje. Pomocí softwarových změn dokážeme také měnit jeho parametry bez jakéhokoli fyzického zásahu do zapojení. Signál je zprvu veden do vstupních obvodů a posléze je vzorkován kvalitním A/D převodníkem, což zahrnuje hardwarovou část SDR. O jeho zpracovávání se dále stará již software.^[4] Obrázek 2.2 ukazuje strukturu konvenčního SDR, kde je zřejmý hardwarový tuner a také softwarová část, jež se stará o signálové zpracování.



Obr. 2.2: Blokové schéma RTL-SDR. Zkratky: **LNA** - Nízkošumový předzesilovač, **VGA** - zesilovač s proměnným ziskem, **LO** - Lokální oscilátor, **LPF** - Dolnofrekvenční propust. Převzato z [2].

Oproti klasickým rádiovým konstrukcím přináší SDR svou podstatou spoustu dalších možností, v budoucích aplikacích má tedy své místo jisté. V dnešní době je SDR záležitostí spotřebního zboží, i lepší přijímače jsou již cenově dostupné.^[5] Protože je softwarově definované rádio v hledáčku veřejnosti, začínají se zavádět patřičné regulace pro jeho provoz. Koncept SDR snížil míru potřebných znalostí pro základní provoz amatérského radiokomunikačního zařízení.

2.2.2 Sféry uplatnění

Využití softwarově definovaného rádia lze sledovat hned v několika oborech a disciplínách. Je hojně využíváno radioamatéry, nalezneme ho také ve vojenské technice či mobilních

komunikacích. Do všech zmíněných sfér je promítnuta hlavní výhoda SDR, tedy změna parametrů přijímače/vysílače pouze pomocí zásahu do softwaru. To samozřejmě vede k úspoře času i financí a lze tyto parametry měnit i dálkově.^[6]

2.2.3 Použité SDR

Pro realizaci kooperativního příjmu signálu bude použito SDR od firmy Airspy. Konkrétně se jedná o typ Airspy R2, jež dokáže operovat ve frekvenčním rozmezí VHF a UHF. Mezi vybrané technické specifikace patří:^[7]

- **Spojitý RX rozsah:** 24 - 1700 MHz
- **Výstup na IQ diagram:** 10 MSamp/s
- **Panoramatické prohlížení frekvenčního spektra:** šířka až 10 MHz
- **1 x Vysokofrekvenční vstup:** SMA konektor, ESD protekce až do 30kV
- **1 x Vysokofrekvenční výstup:** Průchozí smyčka (RF Loop), ESD protekce 15kV
- **2 x Vysokorychlostní ADC vstupy:** až 80 MSamp/s, ESD protekce 15 kV
- **Není potřeba dodatečná instalace ovladačů, vše je Plug-and-Play**



Obr. 2.3: Různé produkty Airspy. Použité SDR, Airspy R2, je uprostřed. Převzato z [8].

3

Popis možných zdrojů signálu

V této kapitole jsou shromážděny potřebné informace k realizaci kooperativního příjmu různých zdrojů vysílání, které jsou uvedeny v zadání bakalářské práce. Protože potřebujeme signály pro kooperativní vyhodnocení nejdříve časově zarovnat, nejpodstatnější částí vysílání bude opakující se sekvence bitů, která odděluje jednotlivé datové rámce, tzv. preamble. Pomocí této sekvence bude zaznamenaná komunikace časově zarovnána a vyhodnocena. Dle míry deformace preamble lze také předpovídat kvalitu následujících dat.

3.1 APRS pakety

APRS (Automatic Packet Reporting System) je komunikační protokol, pomocí kterého jsou přenášena aktuální informace z různých geografických zdrojů mezi účastníky v síti. Lze například přijímat data z meteostanice, určit polohu vybraných vysílačů, sledovat automobil či osobu vybavenou APRS vysílačem. Síť APRS umožňuje také posílat zprávy veřejnosti, například upozornění na blížící se nebezpečí a podobně.^[9]

Datové rámce APRS mají strukturu definovanou dle standardu AX.25. Každý rámec je oddělen sekvencí bitů 01111110 (0x7e). Tento byte je přítomen na počátku a konci každého rámce. V následující tabulce 3.1 je označen jako **Flag** byte.^[10] Pomocí této části rámce by bylo možné časově zarovnat kooperativně přijaté signály ze zdrojů APRS.

Flag	Source Address	..	Protocol ID	INFORMATION FIELD	FCS	Flag
1	7	..	1	1-256	2	1

Tab. 3.1: Struktura APRS paketu dle standardu AX.25. Převzato s úpravami. ^[10]

Mapu s objekty vysílající APRS lze najít například zde: <https://aprs-map.info/>

3.2 Meteorologické satelity

Meteorologické satelity mají za úkol nepřetržitě sbírat data o stavu atmosféry Země. Díky tomu lze přistupovat k informacím o počasí a jeho předpovědi. Mezi konkrétní jevy, které satelity pozorují, se řadí například teplota moře, pohyb ledovců a míra sněhové pokrývky. Satelity také dokáží detekovat vulkanické erupce a sledovat pohyb vyprodukovaného popelu. Existuje také mnoho pozemních senzorů, jejichž data jsou pomocí satelitu zpět předána centrální stanici na Zemi. Mezi ně patří pozemní meteorologické stanice, detektory zemětřesení, měrky vodních hladin nebo třeba data o přílivových vlnách.

V provozu jsou dva základní typy meteorologických satelitů. Ty se pohybují buď na geostacionární (~35,800 km nad Zemí) nebo polární oběžné dráze (~850 km nad Zemí).

3.2.1 Geostacionární orbit

Geostacionární satelity se pohybují spolu s otáčením Země, ze zemského povrchu se proto jeví jako stacionární. V reálném čase pořizují fotografie atmosféry, ze sekvence získaných obrázků lze sestavit záznam, kde je zřetelný pohyb mraků. Tím lze sledovat například bouře, fronty, hurikány či předpovídat jejich vývoj.

3.2.2 Polární orbit

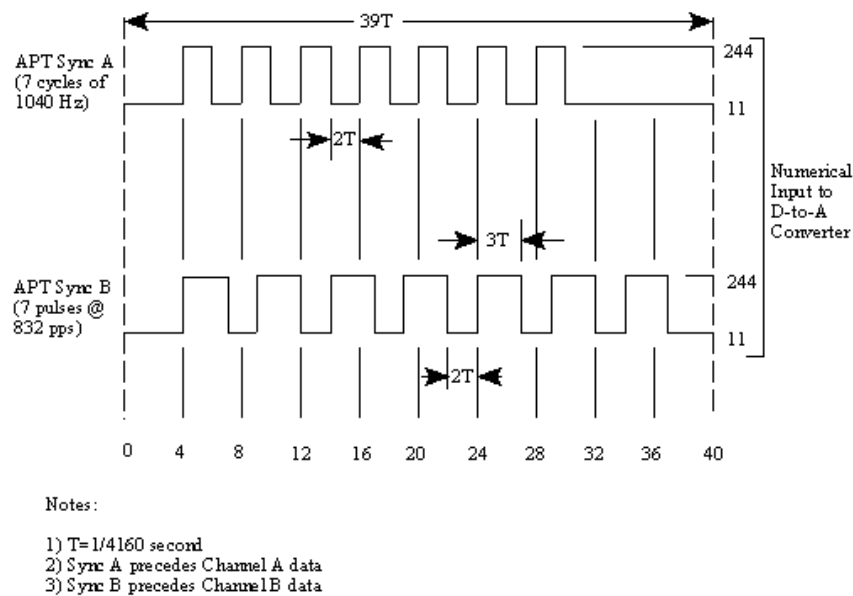
Satelity v polárním orbitu putují přibližně ve směru poledníků Země a mezi každým přeletem prochází oběma póly. Fotografie, které jsou těmito satelity pořízeny, lze poskládat do panoramatického pohledu a ten poté opět spojit s obrázky pořízenými při přeletu nad dalším poledníkem. Vzniká tak fotografie, která zobrazuje větší část povrchu. Jelikož satelity na polární oběžné dráze krouží v podstatně menší výšce nad Zemí, mohou sledovat bouře s větší blízkostí. Podávají tedy detailnější informace o bouřích, než satelity na geostacionární oběžné dráze.^[11]

3.2.3 Příjem signálu

Známým komunikačním módem, pomocí něhož posílají meteorologické satelity zaznamenané obrázky na pozemní stanice, je tzv. APT (Automatic Picture Transmission). Obrazová data odesílaná v tomto módu jsou analogově modulována. Dnes je tento způsob komunikace využíván hlavně meteorologickými satelity NOAA.

Vysílání je tvořeno synchronizačními daty, telemetrií a dvěma kanály, A a B, pro pořízené snímky. Kanál A je vyhrazen pro snímek ve viditelném spektru za dne, kanál B zase pro snímek infračerveného spektra za tmy. V datovém rámci se před oběma kanály nacházejí zmíněné synchronizační úseky.^[12] Ty by mohly být použity pro zarovnání

a vyhodnocení v případě použití tohoto zdroje signálu pro kooperativní příjem. Detail synchronizačního úseku je na obrázku 3.1.



Obr. 3.1: Detail synchronizačního úseku vysílání APT. Převzato z [12].

3.3 Radioamatérské satelity

Radioamatérské satelity, konkrétně malé družice, jsou v dnešní době, vzhledem ke své velikosti, funkčnosti a oproti klasickým satelitům výrazně redukováné pořizovací ceně, velmi populární technologie. Tento fakt naznačuje existenci velkého množství zdrojů signálu, které lze pro kooperativní příjem využít. Nejčastěji lze komunikaci s radioamatérskými satelity navázat ve vyhrazeném 70 cm UHF pásmu (440 MHz ham band). Díky rozmanitosti a počtu těchto zařízení není datový rámec nijak standardizovaný, je proto nutné si detaily o komunikaci satelitu nejdříve opatřit u konkrétních provozovatelů.

Callsign: OK0SAT (LUCKY7)
Frequency: 437.525 MHz
Output Power: 27 - 30 dBm (0.5 - 1 W)
Polarization: Linear (quasi-omni, V-dipole 120°)
Modulation: GFSK, BT = 0.5
Downlink Bitrate: 4800 bps
Beacon Period: 40 seconds
Sync Word: 0x2DD4
Scrambling Polynomial: PN9, Seed 0xFFFF
CRC: CRC-16 (IBM), $X^{16}+X^{15}+X^2+1$



Beacon Format:

Header (16 B)	Sync Word (2 B)	Payload (35 B)	CRC (2 B)
16x 10101010	0x2DD4	DATA	CRC-16 IBM ($X^{16}+X^{15}+X^2+1$)

Obr. 3.2: Detaily komunikace satelitu Lucky-7. Převzato z [19].

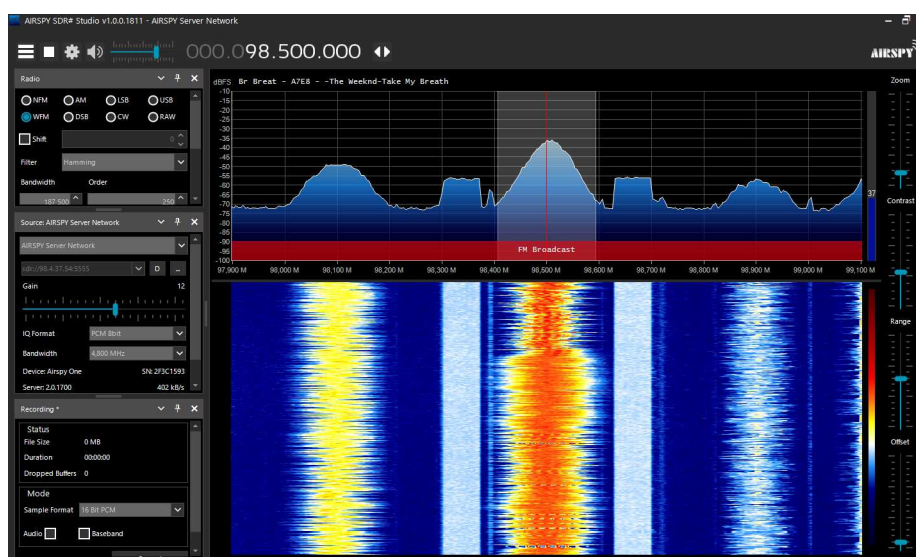
4

Realizace kooperativního příjmu signálu

V této kapitole jsou popsány provedené kroky pro realizaci kooperativního příjmu pomocí softwarových nástrojů. Jako vhodné zdroje vysílání byly zvoleny radioamatérské družice. K tomuto výběru přispěla hlavně jejich popularita a také fakt, že se na ZČU FEL nachází pozemní stanice pro komunikaci se satelitem VZLUSAT-2.

4.1 Příjem pomocí sítě SpyServer

Jedná se o celosvětovou síť SDR, ke které lze přistupovat pomocí programu SDRSharp a internetového připojení. Je možné se připojit na jakoukoli dostupnou stanici a v reálném čase sledovat frekvenční spektrum v rozsahu cílového SDR. Program také nabízí několik typů demodulace a okamžitý zvukový výstup. Lze tak například poslouchat rádio stanice téměř kdekoli po světě a zvuk nahrávat. Stejným způsobem lze zachytit i vysílání satelitů.



Obr. 4.1: Hlavní obrazovka aplikace SDRSharp s probíhajícím poslechem FM rádia

Užitečným doplňkem pro SDRSharp je také plugin, který umožňuje přímé propojení s programem Gpredict. V reálném čase tak SDRSharp dokáže automaticky ladit kmitočty, aby kompenzoval Dopplerův posun přijímaného signálu.

Pro kooperativní příjem je tato metoda však nepraktická. Bylo by nutné ručně operovat s několika instancemi programu SDRSharp najednou a také hlídat časy, kdy sledovaný satelit přeletí nad danou stanicí. Jako alternativa k síti SDR přijímačů SpyServer byla zvolena síť SatNOGS, kde tato omezení neexistují. Kooperativní příjem sítě SatNOGS je podrobně popsán v následující sekci. Od způsobu příjmu pomocí sítě SpyServer bylo upuštěno již v rané fázi práce, proto této sekci již nebude věnována další pozornost.

4.2 Příjem pomocí webových rádií

Na internetu je dostupná řada pozemních stanic, které mají svou vlastní webovou stránku. Je zde možné poslouchat různou vesmírnou komunikaci, pozorovat spektrogram¹ případně vše nahrát a uložit. Jedná se však o stejnou nepraktičnost z hlediska realizace kooperativního příjmu, stejně jako u metody popsané v předchozí sekci. Pro referenci jsou uvedeny internetové stránky, které tato separátní webová rádia shromažďují do seznamu a mapy:

- <https://www.receiverbook.de/>
- <http://websdr.org/>

4.3 Příjem pomocí sítě SatNOGS

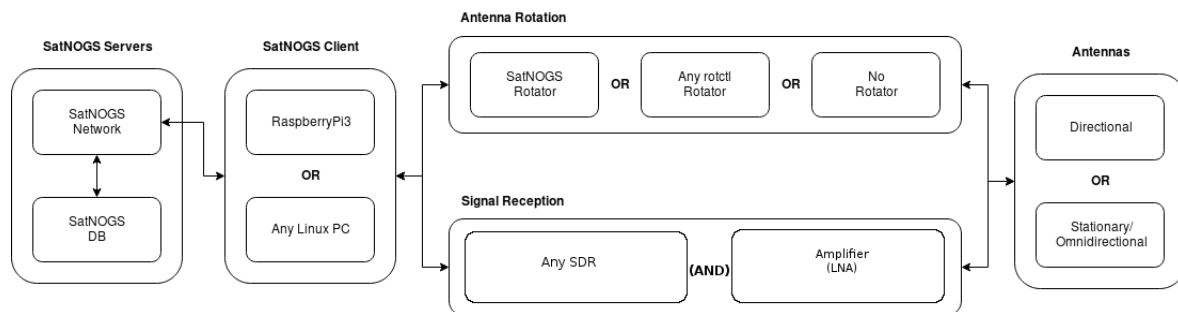
Hlavní myšlenkou projektu SatNOGS je vytvořit rozsáhlou databázi satelitů obsahující veškeré informace, mezi které patří například komunikační frekvence, použitá modulace, bitový tok, telemetrie satelitu a mnoho dalších. Nejdůležitější částí této databáze je ovšem samotné nahrávání komunikace satelitu, které je posléze uloženo a zpřístupněno ostatním uživatelům. Každý má možnost vybrat jakoukoli funkční pozemní stanicí a naplánovat observaci, přitom čas, kdy se sledovaná družice bude v dosahu příjmu, bude automaticky vypočítán. Data jsou, po přeletu satelitu a úspěšném zpracování pozemní stanicí, dostupná na webové stránce projektu. Jedná se tak o výborný nástroj nejen pro začínající radioamatéry, ale i znalé v oboru. Díky možnosti plánovat více observací satelitu paralelně na různých pozemních stanicích byla zvolena tato síť pro realizaci kooperativního příjmu.

Při prvotní snaze získat data kooperativním příjmem bylo nutné naplánovat observace na několik blízkých pozemních stanic, konkrétně pro satelity LUCKY-7 a VZLUSAT-2.

¹Spektrogram zobrazuje frekvenční spektrum signálu, které je proměnné v čase. V anglické literatuře je označován jako waterfall, tedy grafický průběh připomínající vodopád.

Podmínkou pro plánování vlastních observací, a tedy i užívání pozemních stanic ostatních, je aby uživatel do databáze také něčím přispěl. Možnost plánování observací konkrétního satelitu v síti přijímačů SatNOGS je tedy zpřístupněna až po založení své vlastní pozemní stanice, která musí být v provozu (tzn. režim online, nikoli testovací či offline). Tímto by bylo možné do kooperativního příjmu zapojit i vlastní stanici.

Nejčastěji volenou metodou při stavbě pozemní stanice je instalace klienta na minipočítač Raspberry Pi^[13], kam se ukládají data z observací, která lze následně číst přes webové rozhraní (z kterého lze observace i zakládat). Při tomto postupu lze minipočítač, příslušné SDR i pomocnou elektroniku umístit do jednotného pouzdra které je pak usazeno blízko antény. Alternativní cestou je instalace klienta na PC s operačním systémem na bázi Linuxu. Kvůli prostorové náročnosti na umístění antény a samotného klienta však nemusí být vždy možná. Proveditelné propojení antény s klientem a absence Raspberry Pi přispělo ke zvolení alternativní metody instalace klienta na PC.

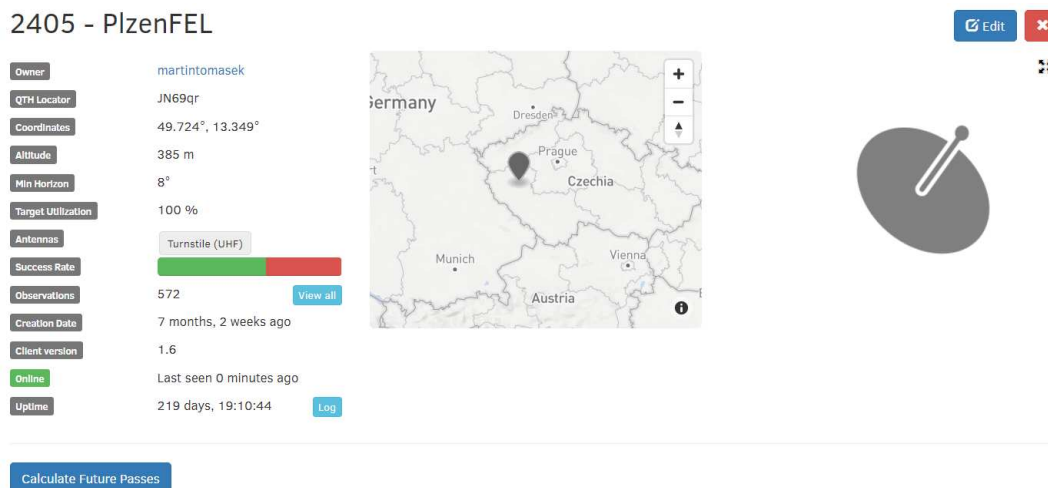


Obr. 4.2: Různé možnosti provedení pozemní stanice. Převzato z [14].

4.3.1 Vytvoření pozemní stanice

Prvním krokem je založení profilu a nové pozemní stanice na webu SatNOGS Network. Je nutné zadat hlavní parametry stanice, mezi které patří i geografická poloha, konstrukční typ antény, přítomnost rotátoru, atd. Námi založená pozemní stanice má tyto parametry:

- Název stanice: **PlzenFel**
- Číslo stanice v databázi SatNOGS: **2405**
- Typ antény: **stacionární, zkřížená dipólová anténa tzv. "turnstile"**
- Frekvenční rozsah: **430.000 MHz - 440.000 MHz, VKV (UHF)**
- Lokace: střecha Fakulty elektrotechnické ZČU
 - zeměpisná šířka: 49.724°
 - zeměpisná délka: 13.349°
 - nadmořská výška: 385 m



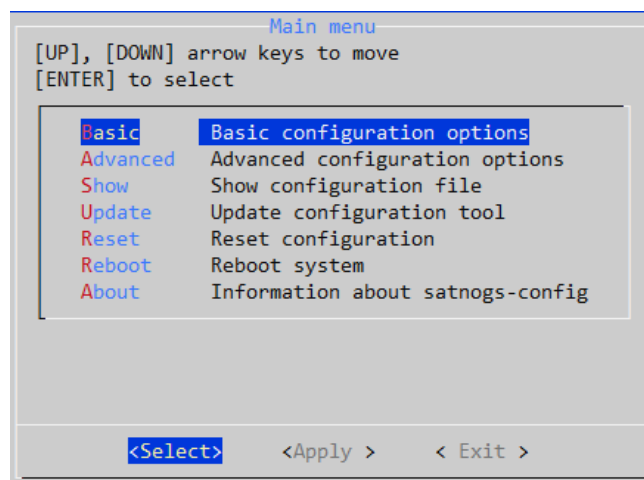
Obr. 4.3: Detaily vlastní pozemní stanice ve webovém prostředí SatNOGS.

4.3.2 Instalace a nastavení klienta

Stahování a instalace potřebných souborů pro instalaci klienta je poměrně metodický postup a nebude konkrétně popisován. V této subsekcí jsou hlavně shrnuty úkony pro úspěšné zprovoznění a ověření jeho funkce.

1. Pokyny k instalaci

Na webových stránkách projektu SatNOGS^[16] je popsán postup instalace klienta pro pozemní stanici a jeho následnou konfiguraci. Klienta lze instalovat pouze na PC s operačním systémem Debian (konkrétně Debian 10 Buster), což je volně dostupná distribuce Linuxu. Instalace klienta na ostatní linuxové distribuce končí vždy chybou. Na nejnovější verzi Debianu 11 Bullseye se instalace nezdařila taktéž. V popisu instalace lze také nalézt všechny potřebné balíky pro správný chod klienta, stejně tak jako odkazy na stažení a instalaci potřebných ovladačů pro používané SDR.



Obr. 4.4: Hlavní nabídka SatNOGS klienta, který operuje na naší pozemní stanici.

2. Konfigurace klienta

Po úspěšné instalaci je třeba zadat parametry stanice do samotného klienta, většina je totožná s parametry zadanými do webového rozhraní, není třeba je znovu dohledávat. Aby mohl klient komunikovat s webovým rozhráním je nutné nejdříve zadat tzv. **API klíč**, který je pro každého uživatele jedinečný. Lze ho získat na webovém profilu uživatele, který byl zakládán spolu s pozemní stanicí. Tento klíč je poté vložen do základní konfigurace **Main menu** → **Basic** → **SATNOGS_API_TOKEN** spolu s ostatními, již zmíněnými, údaji. Konfigurace pokračuje uvedením parametrů týkajících se použitého SDR. To je uvedeno na příslušné webové stránce SatNOGS Wiki^[17], kde je popsáno nastavení klienta pro různé typy podporovaných přijímačů. Následuje výčet zadaných parametrů SDR pro klienta SatNOGS a Airspy R2 SDR:

- **Main menu** → **Basic** → **SATNOGS_SOAPY_RX_SAMP_RATE**
 - vzorkovací kmitočet. Voleno $10e6$, tj. 1 MSamp/s.
- **Main menu** → **Basic** → **SATNOGS_SOAPY_RX_DEVICE**
 - volba ovladače pro SDR. Voleno *driver=airspy*.
- **Main menu** → **Advanced** → **Radio** → **SATNOGS_GAIN_MODE**
 - mód zisku, jsou k dispozici dvě možnosti: *Settings Field* a *Overall*
 - mód *Settings Field* umožňuje měnit parametry různých funkčních bloků SDR. Lze například ladit slučovač MIX, ziskově proměnný zesilovač VGA nebo zabudovaný nízkošumový předzesilovač LNA. Lze také specifikovat přítomnost externího LNA. Při volbě tohoto módu se konkrétní parametry zadávají do **SATNOGS_OTHER_SETTINGS**.
 - mód *Overall* automaticky rozdělí hodnotu zadanou do **SATNOGS_RF_GAIN** mezi funkční bloky SDR.
- **Main menu** → **Advanced** → **Radio** → **SATNOGS_OTHER_SETTINGS**
 - protože se mezi anténu a SDR nachází přídavný nízkošumový předzesilovač, byl zvolen mód zisku *Settings Field*. Doporučené nastavení při přítomnosti předzesilovače je $LNA=9$, $MIX=6$, $VGA=11$. Na základě kvality přijatého signálu lze tyto hodnoty později měnit a dosáhnout tak lepšího příjmu.

3. Ověření instalace

Ještě před samotným pozorováním satelitu pomocí právě zprovozněné pozemní stanice je nutno ověřit funkci a správné nastavení klienta. Mezi nejčastější zdroje chyb patří nefunkční ovladač daného SDR či chybná instalace klienta. Funkci již nainstalovaného ovladače lze ověřit pomocí příkazu (bez čárky)

```
SoapySDRUtil --probe="driver=airspy",
```

který je uveden na webové stránce^[17] pro instalaci ovladače na Airspy R2 SDR. Následující obrázek 4.5 ukazuje výpis příkazu klientem. Při správné funkci ovladače jsou ve výpisu správně vyčíslena rozmezí volitelných hodnot jako LNA Gain, MIX Gain, VGA Gain, měly by být také uvedeny ostatní parametry, jako frekvenční rozsah, či dostupné vzorkovací kmitočty. V opačném případě bude na uvedených pozicích nula.

```
#####
# Soapy SDR -- the SDR abstraction library #
#####

Probe device driver=airspy
[ERROR] SoapySDR::loadModule(/usr/local/lib/SoapySDR/modules0.7/libairspySupport.so)
duplicate entry for airspy (/usr/lib/x86_64-linux-gnu/SoapySDR/modules0.7/libairspySupport.so)

-----
-- Device identification
-----
driver=Airspy
hardware=Airspy
device_id=0
origin=https://github.com/pothosware/SoapyAirspy

-----
-- Peripheral summary
-----
Channels: 1 Rx, 0 Tx
Timestamps: NO
Other Settings:
  * Bias tee - Enable the 4.5V DC Bias tee to power SpyVerter / LNA / etc. via antenna connection.
    [key=bias_tee, default=false, type=bool]
  * Bit Pack - Enable packing 4 12-bit samples into 3 16-bit words for 25% less USB traffic.
    [key=bitpack, default=false, type=bool]

-----
-- RX Channel 0
-----
Full duplex: YES
Supports AGC: YES
Stream Formats: CS16, CF32
Native Format: CS16 [full-scale=32767]
Antennas: RX
Full gain range: [0, 45] dB
  LNA gain range: [0, 15] dB
  MIX gain range: [0, 15] dB
  VGA gain range: [0, 15] dB
Full freq range: [24, 1800] MHz
RF freq range: [24, 1800] MHz
Sample rates: 10, 2.5 Msps
```

Obr. 4.5: Výpis příkazu pro kontrolu funkčnosti SDR ovladače. Chyba na začátku výpisu pouze upozorňuje na duplikátní soubor, což nemá vliv na výslednou funkci.

Dále je nutné ověřit, zda je služba SatNOGS aktivní, vyloučíme tím tak chybnou instalaci klienta. V manuálu^[16] je uveden příkaz (bez čárky)

```
systemctl status satnogs-client,
```

který vypíše momentální status klienta, viz. obrázek 4.6.

```
ivo@satnogs:~$ systemctl status satnogs-client
● satnogs-client.service - SatNOGS client
   Loaded: loaded (/etc/systemd/system/satnogs-client.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-01-23 16:41:10 CET; 3 months 23 days ago
 Main PID: 470 (satnogs-client)
    Tasks: 22 (limit: 4915)
   Memory: 1.6G
    CGroup: /system.slice/satnogs-client.service
           └─470 /var/lib/satnogs/bin/python3 /var/lib/satnogs/bin/satnogs-client
```

Obr. 4.6: Výpis je dle verifikace instalace^[16] správný, stanice je tedy provozuschopná.

Pokud jsou předchozí kroky úspěšně dokončeny, stanice je nyní online a také propojena s webovým rozhraním SatNOGS. Na webových stránkách stanice^[15] nalezneme i její aktuální stav, a to **online**. Nyní lze plánovat observace přeletů satelitů na všech dostupných stanicích v síti SatNOGS. Společně s tím mohou ostatní uživatelé k tomuto účelu využít i naši, právě vytvořenou, stanici.

4. Zkušební pozorování

Spektrogramy testovacích observací lze nalézt v příloze B této práce. Pomocí čísla pozorování, které je zde uvedeno v popisku jednotlivých obrázků, lze další informace o observaci zpětně dohledat v databázi SatNOGS.

4.3.3 Výběr družice

Po úspěšném spuštění pozemní stanice je vybrána družice, jejíž vysílání bude použito jako zdroj signálu pro kooperativní příjem. Signály musí obsahovat tzv. preambuli, což je nejčastěji posloupnost jedniček a nul, která se nachází na počátku každého datového rámce. Právě pomocí preambule mohou být kooperativně přijaté signály zarovnány, demodulovány a vyhodnoceny. Samotná interpretace, tedy dekódování signálu, není předmětem této práce.

Prvním vytipovaným satelitem byla česká vědecká družice Lucky-7, jejíž signál byl mimo jiné zachycen i při zkušebním pozorování. Na webových stránkách družice^[18] je dostupný dokument s informacemi pro radioamatéry. Lze zde najít všechny základní údaje pro příjem vysílání, stejně tak jako formát datového rámce a detailní výpis dat, které obsahuje (tabulka 4.1). Po úspěšném zachycení komunikace družice naší pozemní stanicí

Hlavička [16 B]	Synchronizační slovo [2 B]	Užitečná data [35 B]	CRC [2 B]
16x10101010	0x2DD4	DATA	CRC-16 IBM

Tab. 4.1: Tabulka popisující formát datového rámce na jehož počátku je 16 bytová hlavička, tedy preambule. Převzato s úpravami.^[19]




(spektrogramy obdobné jako v příloze B) byl získaný signál nahrán do prostředí MATLAB, kde měla být zkoumána přítomnost preambule. Ani však po aplikaci autokorelační funkce² se nepodařilo najít opakující se úseky, dle kterých by bylo možno signál při kooperativním příjmu více přijímači zarovnat. Bylo zjištěno, že takto družice vysílá při výměně dat s pozemní stanicí, kdy dochází například ke stahování obrázků. To samozřejmě způsobuje, že je samotný signál poměrně obsáhlý a obtížně se zkoumá. Nabízí se tedy možnost pozorovat satelit v době, kdy je v periodických úsecích vysílán pouze maják, angl. beacon. Datový obsah signálu bude v tomto případě značně redukován.

Ve snaze zachytit maják družice Lucky-7 naší pozemní stanicí byly plánovány noční přelety satelitu. V této době by měl vysílat pouze maják i při přeletu nad ČR, kde má svou pozemní stanicí a normálně by probíhala výměna dat. Výsledkem několika plánovaných přeletů byl však vždy prázdný spektrogram.

²Jedná se o druh signálové korelace, kde je originální signál postupně posouván vůči své kopii na obě strany. Výsledkem je autokorelační funkce, tedy závislost míry korelace na relativním posuvu originálu a jeho kopie. Lze tak například určit opakující se úseky, které jsou v signálu obsaženy.

Po prvotním nezdaru se satelitem Lucky-7 byla pro nové pozorování vybrána další česká vědecká družice VZLUSAT-2, jejíž maják podobně obsahuje preambuli tvořenou alternující posloupností jedniček a nul.^[20]

Tentokrát však pro získání signálu nebyla použita naše pozemní stanice, ale databáze observací SatNOGS. Zde byly vybrány tři nahrávky majáku VZLUSAT-2, které byly zachyceny pozemními stanicemi s malou vzájemnou vzdáleností v desítkách metrů (obr. 4.7). Při kontrole v prostředí MATLAB byla ověřena přítomnost preambule i dostatečná kvalita signálu. Nahrávky budou tedy použity pro další vyhodnocení. Spektrogramy nahrávek se nacházejí v příloze C.

ID	Satellite	Frequency	Mode	Timeframe	Results	Observer	Station
5477367	VZLUSAT-2	437.325 MHz	MSK AX.100 Mode S 4800	2022-02-16 02:43:20 2022-02-16 02:48:59		MAUSyagi	1675 - MAUSyagi
5477368	VZLUSAT-2	437.325 MHz	MSK AX.100 Mode S 4800	2022-02-16 02:43:20 2022-02-16 02:48:59		MAUSyagi	2134 - MAUSyagi
5477369	VZLUSAT-2	437.325 MHz	MSK AX.100 Mode S 4800	2022-02-16 02:43:20 2022-02-16 02:48:59		MAUSyagi	2510 - MAUSyagi

Obr. 4.7: Vybrané observace z databáze SatNOGS pro satelit VZLUSAT-2.

5

Zpracování získaných dat

Tato kapitola je věnována popisu skriptu pro vyhodnocení získaných dat pomocí sítě SatNOGS, konkrétně pro maják satelitu VZLUSAT-2. Je členěna do několika sekcí, kde každá obsahuje popis stěžejních částí programu s příloženými úryvky kódu a komentářem. Celý skript lze nalézt v příloze A této práce, nachází se zde také každá proměnná zmíněná v následujících sekcích.

Každý získaný zvukový záznam obsahuje výstupní signál FM demodulátoru v požitém SDR. Úkolem programu je převést tento signál na bitový tok, který je následně pomocí různých přístupů vyhodnocen. Program pracuje v těchto hlavních krocích:

1. Uživatelský vstup, volba vyhodnocení
2. Načtení zvukových nahrávek získaných ze sítě SatNOGS
3. Generování ideálního průběhu preamble
4. Načtení pětisekundového časového úseku do datového zásobníku
5. Korelace ideálního průběhu preamble s preambulemi obsaženými v nahrávkách
6. Časové zarovnání nahrávek
7. Převod signálu do nespojitě digitální podoby
8. Výpis dle zvolené metody vyhodnocení

5.1 Datový zásobník (buffer)

Pro jednodušší a rychlejší manipulaci s daty jsou všechny nahrávky rozděleny do dílčích úseků. Po vyhodnocení celého úseku program načte do zásobníku další. Zásobník je tvořen maticí **martixBuffer**, která je postupně plněna vzorky. Sloupce zásobníku odpovídají počtu nahrávek získaných pomocí kooperativního příjmu, řádky pak počtu vzorků na jeden

dílčí úsek. Počet sloupců zásobníku program automaticky určí dle počtu prvků struktury, do které se nahrávky ukládají. Počet řádků je nutno dopočítat pomocí vzorkovací frekvence a zvolené délky úseku dle rovnice 5.1

$$sampleBuffer = \frac{timeBuffer}{sampleTime} \quad [-], \quad (5.1)$$

kde *sampleBuffer* - datový zásobník, *timeBuffer* - délka časového úseku [s] a *sampleTime* - časové trvání jednoho vzorku [s]. Funkce `audioread` čte nahrávky při vzorkovacím kmitočtu *sampleRate* o velikosti 48.000 vzorků/sek. Délka časového úseku v *timeBuffer* je 5 sekund. Datový zásobník bude tedy obsahovat 240.000 vzorků na jednu nahrávku.

Hodnotu délky časového úseku nelze libovolně volit, nemusí být totiž zaručeno, že při každé zadané hodnotě program skončí bez chyby. Při nalezení užitečných dat program nepřetržitě pokračuje ve čtení zásobníku až do konce délky datového rámce. Pokud je hodnota délky časového úseku zvolena špatně, může se stát, že se data nacházejí u konce zásobníku a program již čte mimo rozsah `matrixBuffer`. Délka *timeBuffer* byla volena záměrně, jelikož k tomuto chybnému stavu nedochází. Je také nutné dodat, že v každém časovém úseku musí být pouze jeden datový rámeček nahrávky, ostatní by byly ignorovány.

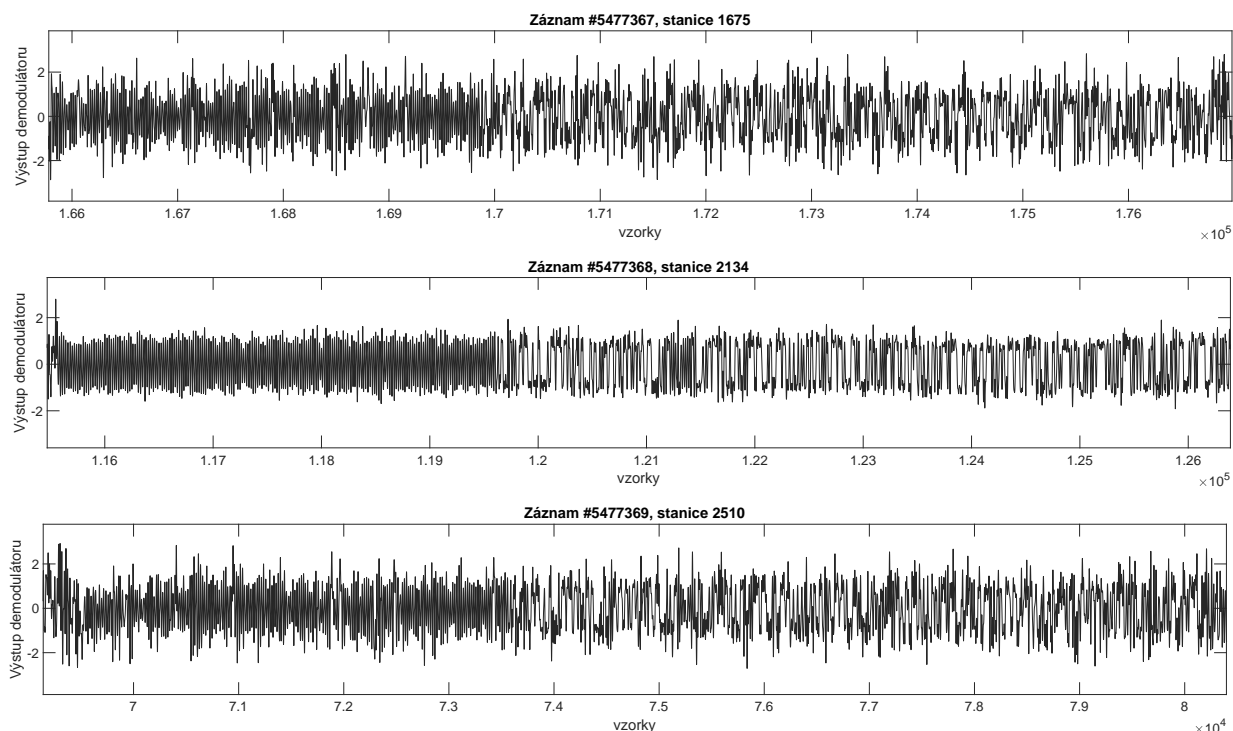
```
satnogs.obs1 = audioread('satnogs5477367.ogg');
satnogs.obs2 = audioread('satnogs5477368.ogg');
satnogs.obs3 = audioread('satnogs5477369.ogg');
(...)
for structElem = 1:numObservations
    obsElem(structElem) = numel(satnogs.(fieldNames{structElem}));
end
for k = 1 : floor(min(obsElem)/sampleBuffer)-1
    for row = 1:sampleBuffer
        for col = 1:numObservations
            matrixBuffer(row,col) = satnogs.(fieldNames{col})(row+sampleBuffer*k);
            (...)
        end
    end
end
```

Úryvek kódu popisuje funkci `buffer`¹. Získané nahrávky mohou být různě dlouhé, data se ukládají v časovém trvání nejkratšího záznamu.

První `for` cyklus zjišťuje počet vzorků v každé nahrávce, která je uložena do struktury na počátku skriptu. Zjistíme tak dobu trvání každé z nahrávek. Druhý `for` cyklus obstarává plnění matice `matrixBuffer` postupně po řádcích, konečný počet řádků bude odpovídat

¹Řádky obsahující znaky `'(..)'` označují vynechanou část kódu, která je pro vysvětlení momentální problematiky nepotřebná.

výsledku z rovnice 5.1. Proměnná k násobí ukazatel na strukturu záznamů. Po každém 5 sekundovém úseku, tedy po 240.000 vzorcích uložených do `matrixBuffer`, se k zvýší o



Obr. 5.1: Časově nezarovnané signály majáku satelitu VZLUSAT-2 získané metodou kooperativního příjmu. Je zřejmá přítomnost preamble následovaná daty.

5.2 Příprava signálu pro zpracování

V této sekci je popsána přípravná fáze zpracování signálu, která má za úkol upravit nahrávky pro konečný převod na bitový tok.

5.2.1 Funkce `finddelay`

Pro práci s přijatými signály a jejich následné vyhodnocení je potřeba tyto signály nejdříve časově zarovnat. Skript pro tento účel využívá funkci `finddelay`^[21], jejíž výstupní hodnota je počet vzorků, o který jsou dva vektory vzájemně posunuty. Jedná se v podstatě o rozšíření vzájemné korelační funkce, kde se proti referenčnímu vektoru posouvá vektor druhý. Výsledkem je počet vzorků o které jsou oba vektory vzájemně posunuty, tedy pouze nejvyšší hodnota křížové korelační funkce. Funkci lze demonstrovat na následujícím kódu:

```
vect1 = [1 2 3];
vect2 = [0 0 1 2 3];
```

```
vectOut = finddelay(vect1,vect2);
```

Výpis vektoru vectOut: 2

Vektor *vect2* je zde za referenčním vektorem *vect1* zpožděn. Funkce `finddelay` zpoždění vypočítá a výsledek je uložen do proměnné *vectOut*. Znaménko výsledku vyjadřuje, zda je druhý vektor za referenčním vektorem zpožděn, či ho předbíhá. V tomto případě má tedy proměnná *vectOut* znaménko kladné.

5.2.2 Funkce circshift

Při použití funkce `circshift`^[22] dochází k cirkulárnímu posuvu dat ve vektoru. Data lze posouvat na obě strany, dle znaménka v argumentu funkce. Pro časové zarovnání signálů je tato funkce použita v kombinaci s `finddelay`, jejíž výstup je právě argumentem pro `circshift`. Zde je jednoduchá ukázka funkce `circshift`:

```
vect1 = [1 2 3 4 5 6 7 8 9 10];
```

```
vectOut = circshift(vect1,3);
```

Výpis vektoru vectOut: [8 9 10 1 2 3 4 5 6 7]

```
vectOut = circshift(vect1,-3);
```

Výpis vektoru vectOut: [4 5 6 7 8 9 10 1 2 3]

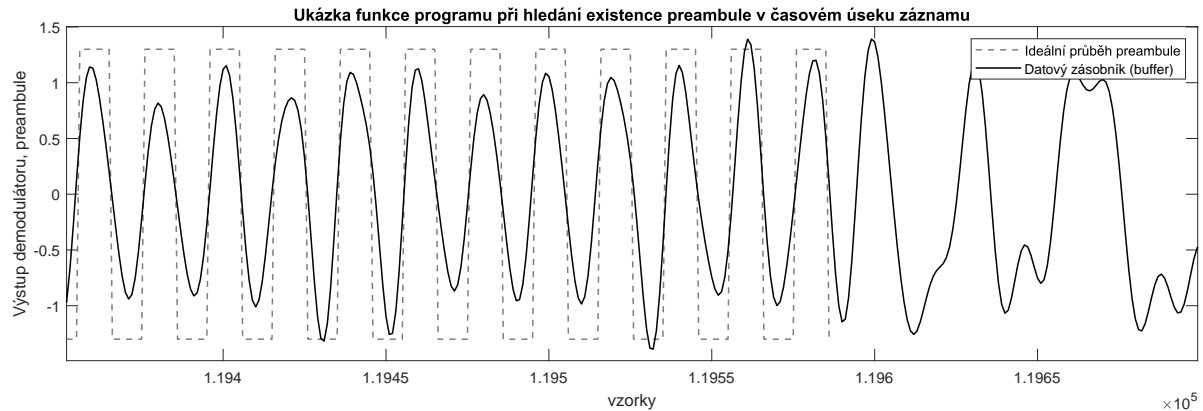
5.2.3 Vektor průběhu preambule

Každý časový rámeček v signálu získaného z VZLUSAT-2 obsahuje počáteční bitovou posloupnost jedniček a nul, která se nazývá preambule. Pro nás je tato část datového rámce velmi důležitá, protože právě díky preambuli lze přijaté signály časově zarovnat a také předpovídat kvalitu přenosu. Přítomnost preambule ve všech záznamech také zaručí, že má smysl signály časově zarovnat a dále porovnávat. Při chybějící preambuli (nebo její značné deformaci) v jedné z nahrávek by bylo časové zarovnání konkrétního signálu k ostatním značně nepřesné, vyhodnocený signál by již nemusel být relevantní.

```
for corr = 1:numObservations
    if(max(xcorr(matrixBuffer(:,corr),vectorPreamb))>corrSensitivity)
        corrTrue(corr,1) = 1;
        vectorPreambDelay = finddelay(vectorPreamb,matrixBuffer(:,1));
    else
        corrTrue(corr) = 0;
    end
end
```

Program tedy nejdříve potřebuje nalézt preambuli ve všech dostupných záznamech uložených v již zmíněné počáteční struktuře. K tomuto účelu je použita již popsaná funkce

`finddelay`. Všechny záznamy jsou v této funkci postupně porovnávány s uměle vytvořeným ideálním průběhem preamble. Pokud je hodnota korelace vysoká (konkrétně dáno proměnnou `corrSensitivity`), uloží se příznakový bit do `corrTrue` a stejným postupem je zkoumána i další nahrávka. Pokud je vektor `corrTrue` jedničkový znamená to, že byla preamble nalezena ve všech záznamech. Program v tomto případě posílá data dále k vyhodnocení.



Obr. 5.2: Ideální průběh preamble zarovnaný s preambulí obsaženou v signálu.

Do vektoru `vectorPreambDelay` se ukládá posuv mezi ideálním vektorem preamble a první nahrávkou (první sloupec `matrixBuffer`). Proměnná `vectorPreambDelay` obsahuje posun ideálního vektoru preamble vůči prvnímu záznamu. Užití `vectorPreambDelay` je popsáno v další subsekcí.

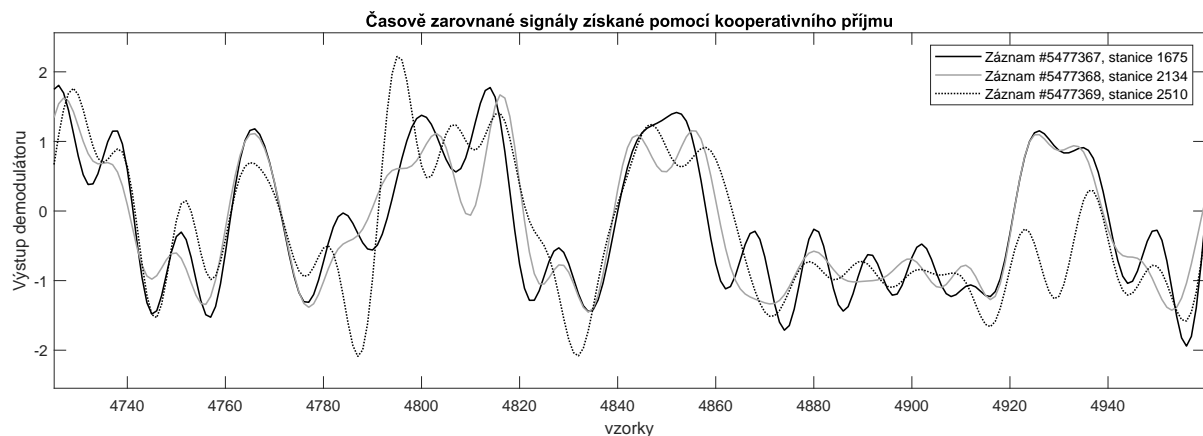
5.2.4 Časové zarovnání signálů

Jak už bylo v předchozích subsekcích nastíněno, k časovému zarovnání využívá samotný skript kombinaci funkcí `finddelay` a `circshift`. Časové zarovnání záznamů probíhá pouze za předpokladu, že je preamble nalezena ve všech nahrávkách.

```
for dataShift = 1 : numObservations
    if(dataShift == 1)
        matrixBuffer(:,1) = circshift(matrixBuffer(:,1),-vectorPreambDelay);
    else
        vectDemodDelay(dataShift)=finddelay(matrixBuffer(:,dataShift-1),
        matrixBuffer(:,dataShift));
        matrixBuffer(:,dataShift) = circshift(matrixBuffer(:,dataShift),
        -vectDemodDelay(dataShift));
        (...)
    end
end
```

Časové zarovnání signálů obstarává tato část kódu. Počet cyklů smyčky `for` je roven počtu zkoumaných záznamů. Při prvním cyklu kdy `dataShift = 1` dochází k posunu vektoru první nahrávky již načtené v datovém zásobníku o hodnotu `vectorPreambDelay`, která byla zmíněna v předchozí subsekcí. To zajistí posun prvního záznamu na začátek datového zásobníku. V dalších cyklech je nalezeno zpoždění ostatních nahrávek vůči prvnímu záznamu a je provedeno zarovnání pomocí funkce `circshift`.

Výhodou zarovnání dat na začátek zásobníku je jednodušší programové odkazování na buffer. Se čtením a následnou demodulací dat lze začít přímo od jeho počátku. Přípravná část pro zpracování signálu je tímto uzavřena.



Obr. 5.3: Časově zarovnané signály majáku VZLUSAT-2 získané pomocí kooperativního příjmu.

5.3 Demodulace signálu

V této sekci je popsán převod časově zarovnaného signálu získaného kooperativním příjmem na binární posloupnost. Konkrétně se jedná o dokončení demodulace, jelikož zvukové nahrávky získané ze SatNOGS jsou již výstupem FM demodulátoru.

5.3.1 Čtení datového zásobníku

Z datového zásobníku `matrixBuffer` se postupně načítají vzorky nahrávek, preambule je vynechána. Neobsahuje žádná užitečná data a slouží pouze k zarovnání signálu. Jelikož `matrixBuffer` obsahuje jen jeden datový rámec VZLUSAT-2, velká část jeho obsahu je náhodný šum. Je tedy zřízen ještě pomocný buffer `sampleBitBuffer`, kam je tento rámec postupně načítán po bitech. v datovém rámci by měl být přibližně stejný ve všech záznamech a je uložený v konstantě `dataBits`. Tato hodnota byla spočítána pomocí časového trvání datového rámce bez preambule `secondsData`, jež bylo předem změřeno v grafické

závislosti mimo program. Rovnice 5.2 konkretizuje výpočet:

$$dataBits = \frac{secondsData}{bitTime} \quad [-], \quad (5.2)$$

kde *secondsData* - časové trvání datového rámce VZLUSAT-2 bez preamble [s] a *bitTime* - časové trvání jednoho bitu [s].

```
for bitArray = 1:numObservations
    while(arrayCounterData < dataBits)
        while(~signChanged)
            sampleBitBuffer(bitBufferCounter,bitArray)=
                matrixBuffer(matrixCounter+samplePreamb,bitArray);
            (...)
        end
    end
end
```

Každá nahrávka se vyhodnocuje postupně. Při zapsání jednoho (či více) bitů do demodulační matice **bitDemod** se proměnná *arrayCounterData* patřičně zvýší. Program nepřetržitě pokračuje v demodulaci nahrávky až do splnutí hodnot *arrayCounterData* a *dataBits*. Je tak vyhodnocen datový rámec a pokračuje se datovým rámcem další nahrávky, hodnota *bitArray* je postupně zvyšována. Po zpracování rámců ve všech nahrávkách se do zásobníku **matrixBuffer** načte nový pětisekundový úsek.

5.3.2 Bitové vyhodnocení

Při bitovém vyhodnocení program nejdříve zkoumá, jak se mění polarita uložených vzorků v *sampleBitBuffer*. Je zřejmé, že pokud signál protne vodorovnou osu v čase dvakrát, skončila bitová posloupnost jedné polarity. Podle počtu vzorků, které momentálně obsahuje *sampleBitBuffer* mezi změnou polarity, se určuje výsledný počet bitů zapsaných do demodulační matice **bitDemod**. Přepočít vzorků na počet bitů je určen dle rovnice 5.3

$$bitLength = n \cdot \frac{bitrate}{sampleRate} \quad [bit], \quad (5.3)$$

kde *n* - počet vzorků *sampleBitBuffer* [vz], *bitrate* - přenosová rychlost [bit/s] a *sampleRate* - vzorkovací kmitočet [vz]. Následující úryvek kódu ukazuje výpočet pomocí prostředí Matlab:

```
bitLength = floor(numel(sampleBitBuffer(:,bitArray))*(bitRate/sampleRate));
```

výsledek *bitLength* je po výpočtu zaokrouhlen dolů funkcí **floor**. Například výsledek *bitLength = 0.9 bit* bude ignorován, vždy je zapotřebí celistvého bitu. Takovéto výsledky se nevyhodnocují, tudíž nejsou ani vypsány ve výstupní demodulační matici **bitDemod**.

5.3.3 Výstup demodulace

V této subsekcí je demonstrován výpis do výstupní matice **bitDemod**. Cyklus `for` se opakuje dle počtu bitů v *bitLength*, který byl určen počtem vzorků v *sampleBitBuffer*. Proměnná *bitLength* ale neobsahuje informaci o znaménku. To lze tedy zjistit pomocí funkce pro střední hodnotu `mean`, která je použita na momentální vzorky v *sampleBitBuffer*.

```
for bitWrite = 1:bitLength
    if(mean(sampleBitBuffer(:,bitArray)) > 0)
        bitDemodBuffer(arrayCounterData,bitArray) = 1;
    else
        bitDemodBuffer(arrayCounterData,bitArray) = 0;
    end
    arrayCounterData = arrayCounterData + 1;
    (...)
end
(...)
bitDemod = cat(1,bitDemodPrev,bitDemodBuffer);
```

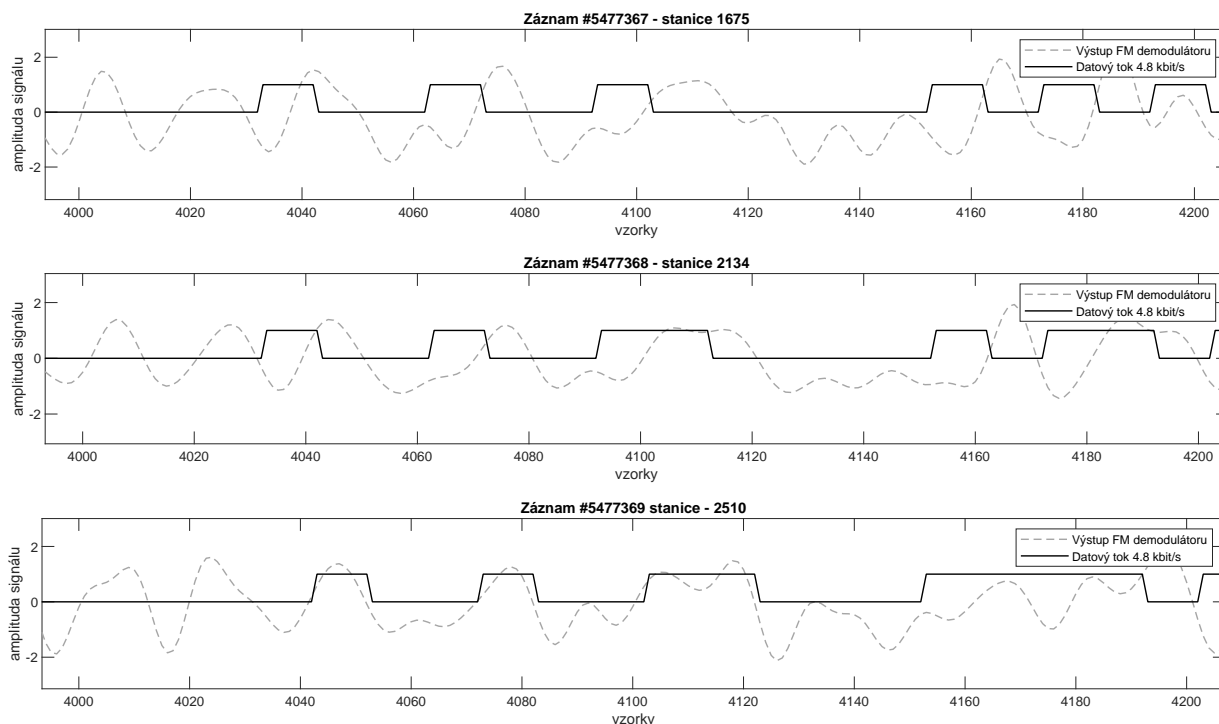
Pokud je tato střední hodnota větší než nula, do demodulační matice **bitDemod** je zapsán znak '1', v opačném případě znak '0'. Do matice **bitDemodBuffer** se ukládají demodulovaná data ze všech záznamů jako mezivýsledek z konkrétního pětisekundového úseku. Vyhodnocená data z dalších úseků se pojí za ta předchozí, výsledkem je tak demodulační matice **bitDemod**, která obsahuje vyhodnocená data ze všech pětisekundových úseků, tedy z celých záznamů. Spojování mezivýsledků v **bitDemodBuffer** obstarává funkce `cat`.

5.4 Dílčí metody vyhodnocení

V této sekci jsou detailněji popsány dílčí metody vyhodnocení, které program vykonává. Paralelní výpis je v podstatě obdoba tvrdé demodulace, kde je signál z každé nahrávky vyhodnocen pouze sledováním polaritu signálu. Zbývající dva způsoby vyhodnocení určují výsledný signál pomocí třech nezávislých získaných kopií. Lze je tedy řadit do demodulačních metod využívajících tzv. měkké rozhodování.

5.4.1 Paralelní výpis

Tímto výstupem je paralelní výpis demodulace všech přijatých signálů, které jsou uloženy ve výstupní matici **bitDemod**. Každý sloupec odpovídá jedné demodulované nahrávce. Tuto metodu vyhodnocení lze vyvolat zadáním znaku '0' při spuštění skriptu.



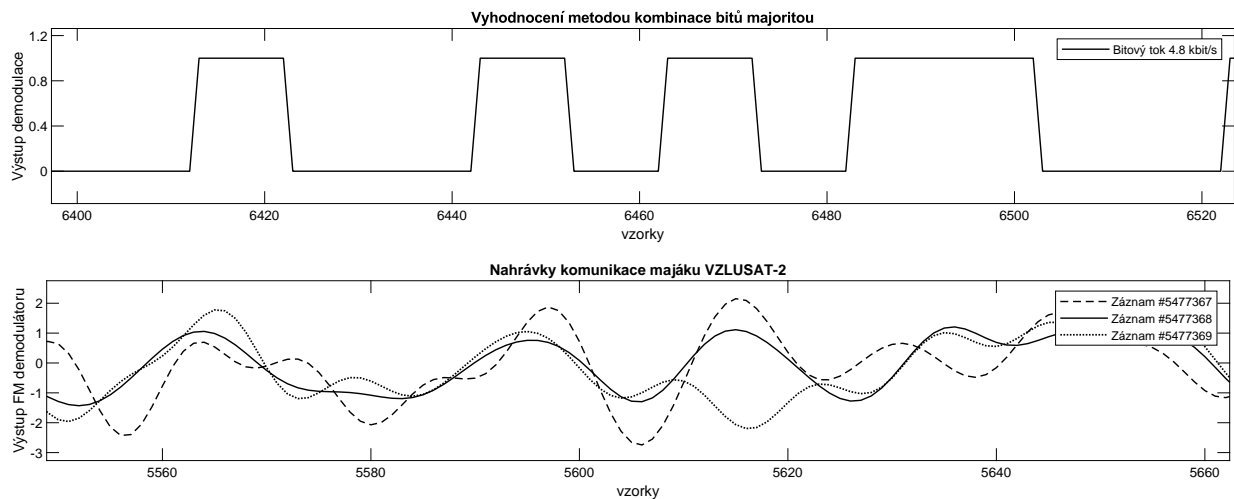
Obr. 5.4: Paralelně vyhodnocené nahrávky majáku satelitu VZLUSAT-2.

5.4.2 Kombinace bitů majoritou

Vyhodnocení přijatého signálu pomocí kombinace bitového toku majoritou spočívá v jednoduchém pravidle nejvyšší četnosti daného znaku. Pokud se v řádku demodulační matice **bitDemod** vyskytne vícekrát jednička než nula, je výsledný bitový tok v konkrétním časovém úseku interpretován znakem '1'. V opačném případě skript vypíše znak '0'. Řádky demodulační matice se tedy sjednotí a výsledný signál je zapsán do jejího prvního sloupce.

```
if(demodMajority)
    for bitMajorityDemod = 1 : numel(bitDemod(:,1))
        bitDemod(bitMajorityDemod,1)=mode(bitDemod(bitMajorityDemod,:));
    end
    bitDemod = bitDemod(:,1);
end
```

Vzniká tak sloupcový vektor, který obsahuje výsledný bitový tok. Tento bitový tok tak reprezentuje konečný přijatý signál, se kterým lze, dále dle vlastních potřeb, pracovat. Pro volbu vyhodnocení majoritou zadá uživatel při spuštění skriptu znak '1'.



Obr. 5.5: Porovnání nahrávek komunikace majáku VZLUSAT-2 s vyhodnoceným signálem použitím metody kombinace bitů majoritou.

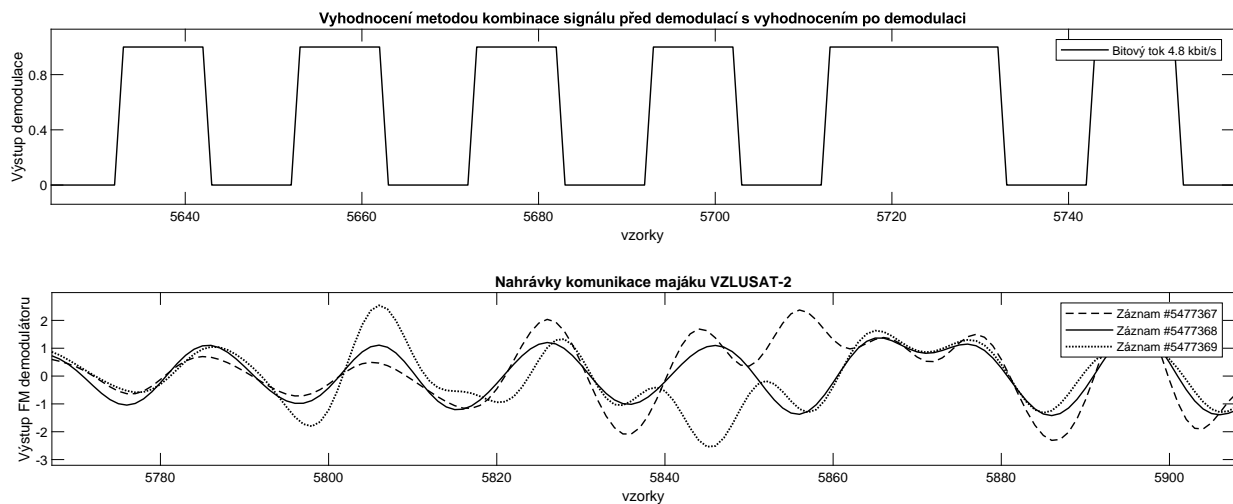
Při pohledu na obrázek 5.5 je nutné si uvědomit, že do demodulační matice **bitDemod** se zapisují bity, které prošly kritérii programu. To znamená, že pokud nebyl konkrétní časový úsek zaznamenán jako celistvý bit, demodulátor jej nebere jako užitečnou informaci a úsek je ignorován. Ve výsledku tedy signál získaný z majáku VZLUSAT-2 nebude vzorkově korespondovat s výstupem tvrdé demodulace. Účinek této metody vyhodnocení lze sledovat například mezi vzorky 5610 až 5640 na spodním subplotu.

5.4.3 Kombinace signálu před demodulací s vyhodnocením po demodulaci

V této metodě jsou dílčí průběhy signálů získané z jednotlivých přijímačů, ještě před demodulací, sečteny. To samozřejmě vyžaduje časové zarovnání všech přijatých signálů a posléze jejich kombinaci. Samotné zarovnání probíhá již automaticky po uložení hodnot časového úseku do zásobníku **matrixBuffer**, program už jen sečte hodnoty zarovnaných časových vzorků ze všech nahrávek a výsledky uloží do prvního sloupce zásobníku. Takto upravený zásobník již putuje k demodulaci, výsledek lze vidět v demodulační matici **bitDemod**.

```
if(demodPreCombine)
    for rowPre = 1:sampleBuffer
        matrixBuffer(rowPre,1) = sum(matrixBuffer(rowPre,:));
    end
end
```

Znovu tak vzniká sloupcový vektor obsahující výsledný datový tok. Pro volbu vyhodnocení kombinací před demodulací zadá uživatel při spuštění skriptu znak '2'.



Obr. 5.6: Porovnání nahrávek komunikace majáku VZLUSAT-2 s vyhodnoceným signálem použitím metody kombinace signálu před demodulací s vyhodnocením po demodulaci.

Při čtení obrázku 5.6 je nutné brát v úvahu stejné skutečnosti, které byly zmíněny v komentáři pod obrázkem 5.5.

6

Závěr

Cílem této bakalářské práce byla praktická realizace kooperativního příjmu pomocí technologie softwarově definovaného rádia a zpracování získaných dat za použití základních metod vyhodnocování které jsou prováděny zhotoveným programem v prostředí MATLAB.

Pro pořízení kooperativně přijatých dat bylo prozkoumáno hned několik možných metod. Z pohledu praktičnosti byla zvolena síť SatNOGS, díky které lze paralelně plánovat observace přeletu satelitů nad vybranými pozemními stanicemi. Získaná data jsou též po neomezenou dobu dostupná. Jako zdroj signálu byly zvoleny radioamatérské satelity. Za tímto účelem byla také zprovozněna vlastní pozemní stanice, která by společně s ostatními zachytila jednu z kopií signálu. Bylo zjištěno, že při komunikačním módu satelitu s pozemní stanicí je datový tok odesílaný z družice značně obsáhlý, tudíž je obtížné rozklíčovat jeho strukturu. Bylo proto zvoleno vysílání majáku družice, kde je datový tok značně redukován.

Za účelem využití databáze observací SatNOGS byl vybrán již kooperativně přijatý signál majáku satelitu VZLUSAT-2 přijatý ze tří, prostorově vzdálených pozemních stanic. Struktura nahrávek byla použita jako předloha při zhotovení vyhodnocovacího programu. Jelikož nemáme k dispozici původní signál vyslaný z VZLUSAT-2 nelze jej porovnat s výsledným, kooperativně přijatým a vyhodnoceným, signálem. Protože by měl kooperativní příjem již z principu sloužit ke kvalitnějšímu příjmu, lze se pouze domnívat, že se námi vyhodnocený signál přibližuje k jeho originální podobě a je v něm obsaženo méně chyb, než v jeho dílčích, kooperativně přijatých, kopiích.

V procesu zhotovení této práce bylo získáno množství informací o problematice softwarově definovaných rádií. Zkušenosti ze zprovoznění pozemní stanice mohou případně sloužit k zopakování procesu ve volném čase. I přestože se námi vytvořená stanice neúčastnila kooperativního příjmu, ze kterého byl vyhodnocen výsledný datový tok, přispěli jsme do komunity SatNOGS další stanicí, která bude rozšiřovat databázi o další observace.

V podkapitole 4.3.1 jsou dostupné další detaily stanice, k datu 25.05.2022 zde bylo realizováno již 580 pozorování.

Literatura

- [1] McGehee, Jared. *Multi-site spatially diverse demodulation of HF propagated signals*. IEEE Military Communications Conference, 2008.
- [2] BniLam, Noori & Joosens, Dennis & Steckel, Jan & Weyn, Maarten. *Low Cost AoA Unit for IoT Applications*. 13th European Conference on Antennas and Propagation, 2019.
- [3] Linnartz, Jean Paul. *Methods for obtaining multiple replicas* [online]. JPL's Wireless Communication Reference Website ©1993-1995 [cit. 22.05.2022]. Dostupné z: <http://www.wirelesscommunication.nl/reference/chaptr05/diversit/divers.htm>
- [4] Matony - OK2IP. *Co je vlastně SDR?* [online]. SDR, Software Defined Radio ©2008-2022 [cit. 22.05.2022]. Dostupné z: <http://sdr.ipip.cz/>
- [5] Matony - OK2IP. *Proč se o SDR zajímat?* [online]. SDR, Software Defined Radio ©2008-2022 [cit. 22.05.2022]. Dostupné z: <http://sdr.ipip.cz/>
- [6] Princy, A. J. *Significant Applications of Software Defined Radios* [online]. All You Need to Know About Software Defined Radio ©2022 [cit. 22.05.2022]. Dostupné z: <https://www.researchdive.com/blog/all-you-need-to-know-about-software-defined-radio>
- [7] AIRSPY. *Technical specifications* [online]. Airspy Products ©2022 [cit. 22.05.2022]. Dostupné z: <https://airspy.com/airspy-r2/>
- [8] shoc Inc. *Gallery* [online]. Airspy HF+/-R2/Mini HF/VHF/UHF RX ©2022 [cit. 22.05.2022]. Dostupné z: <https://www.shoc.ch/airspy.html>
- [9] The APRS Working Group. *INTRODUCTION TO APRS* [online]. APRS PROTOCOL REFERENCE, ©2000 [cit. 19.05.2022]. Dostupné z: <http://http://www.aprs.org/doc/APRS101.PDF>
- [10] The APRS Working Group. *APRS AND AX.25* [online]. APRS PROTOCOL REFERENCE ©2000 [cit. 19.05.2022]. Dostupné z: <http://http://www.aprs.org/doc/APRS101.PDF>

- [11] Galactics. *Weather Satellites* [online]. Satellites Spacesim. Poslední změna 08.08.1997 [cit. 22.05.2022]. Dostupné z: <http://satellites.spacesim.org/english/function/weather/index.html>
- [12] NOAA. *Section 4.2 APT System* [online]. NOAA KLM User's Guide. ©2000-2013 [cit. 23.05.2022]. Dostupné z: <https://web.archive.org/web/20070316190349/http://www2.ncdc.noaa.gov/docs/klm/html/c4/sec4-2.htm>
- [13] Libre Space Foundation. *Raspberry Pi* [online]. SatNOGS Wiki, CC BY-SA 3.0. Poslední změna 06.04.2022 [cit. 14.05.2022]. Dostupné z: https://wiki.satnogs.org/Raspberry_Pi
- [14] Libre Space Foundation. *[Station] Build a Ground Station* [online]. SatNOGS Wiki, CC BY-SA 3.0. Poslední změna 08.02.2019 [cit. 14.05.2022]. Dostupné z: https://wiki.satnogs.org/File:Satnogs_imagemap.png
- [15] Libre Space Foundation. *Ground Stations* [online]. SatNOGS Network, ©2014-2022 [cit. 14.05.2022]. Dostupné z: <https://network.satnogs.org/stations/2405/>
- [16] Libre Space Foundation. *SatNOGS Client Setup* [online]. SatNOGS Wiki, CC BY-SA 3.0. Poslední změna 15.04.2022 [cit. 13.05.2022]. Dostupné z: https://wiki.satnogs.org/SatNOGS_Client_Setup
- [17] Libre Space Foundation. *Software Defined Radio - Airspy Support* [online]. SatNOGS Wiki, CC BY-SA 3.0. Poslední změna 06.02.2022 [cit. 13.05.2022]. Dostupné z: https://wiki.satnogs.org/Software_Defined_Radio#Airspy_support
- [18] SkyFox Labs s.r.o. *Amateur Radio Details* [online]. Lucky 7 Satellite, ©2022 [cit. 15.05.2022]. Dostupné z: <https://www.lucky7satellite.org/radioamateurs>
- [19] SkyFox Labs s.r.o. *Amateur Radio Information* [online]. Lucky 7 Satellite, ©2022 [cit. 15.05.2022]. Dostupné z: https://www.lucky7satellite.org/download/Lucky-7-_Amateur_Radio_Information.pdf
- [20] Výzkumný a zkušební letecký ústav, a. s. *Amateur Radio Information* [online]. Pilsen Ground Station, [cit. 15.05.2022]. Dostupné z: https://www.pilsencube.zcu.cz/vzlusat2/vzlusat2_ham.pdf
- [21] The MathWorks, Inc. *finddelay - Estimate delay(s) between signals* [online]. Matlab Documentation, ©1994-2022 [cit. 14.05.2022]. Dostupné z: <https://www.mathworks.com/help/signal/ref/finddelay.html>
- [22] The MathWorks, Inc. *circshift - Shift array circularly* [online]. Matlab Documentation, ©1994-2022 [cit. 14.05.2022]. Dostupné z: https://www.mathworks.com/help/matlab/ref/circshift.html?s_tid=doc_ta

- [23] Libre Space Foundation. *Observations* [online]. SatNOGS Network, ©2014-2022 [cit. 14.05.2022]. Dostupné z: <https://network.satnogs.org/observations/4882705/>
- [24] Libre Space Foundation. *Observations* [online]. SatNOGS Network, ©2014-2022 [cit. 14.05.2022]. Dostupné z: <https://network.satnogs.org/observations/4882705/>
- [25] Libre Space Foundation. *Observations* [online]. SatNOGS Network, ©2014-2022 [cit. 14.05.2022]. Dostupné z: <https://network.satnogs.org/observations/5477367/>
- [26] Libre Space Foundation. *Observations* [online]. SatNOGS Network, ©2014-2022 [cit. 14.05.2022]. Dostupné z: <https://network.satnogs.org/observations/5477368/>
- [27] Libre Space Foundation. *Observations* [online]. SatNOGS Network, ©2014-2022 [cit. 14.05.2022]. Dostupné z: <https://network.satnogs.org/observations/5477369/>

Příloha A

Použité skripty, zdrojové kódy

A.1 posundemod.m

```
1 %Demodulátor nahrávek SatNOGS
2 %Martin Tomášek
3 sampleRate = 48000;
4 %Struktura pro nahrávky
5 satnogs.obs1 = audioread('satnogs5477367.ogg');
6 satnogs.obs2 = audioread('satnogs5477368.ogg');
7 satnogs.obs3 = audioread('satnogs5477369.ogg');
8 %Definice proměnných
9 fieldNames = fieldnames(satnogs);
10 corrSensitivity = 2800;
11 %bitTreshold = 0.5;
12 numObservations = numel(fieldNames);
13 obsElem = zeros(numObservations,1);
14 timeBuffer = 5;
15 scaleAmpl = 1.3; %Skalár amplitudy preambule
16 bitRate=4800;
17 arrayFill = 1;
18 arrayCounterPreamb = 0;
19 arrayCounterData = 1;
20 signChangeCounter = 0;
21 bitBufferCounter = 1;
22 matrixCounter = 1;
23 secondsPreamb=0.0014*60; %Trvání preambule, změřeno
24 secondsData=0.00241*60; %Trvání dat (po preambuli do náhodného šumu), změřeno
25 sampleTime = 1 / sampleRate;
26 bitTime = 1 / bitRate;
27 samplePreamb = round(secondsPreamb/sampleTime);
28 vectorPreamb = zeros(samplePreamb,1);
29 sampleBuffer = round(timeBuffer/sampleTime);
30 matrixBuffer = zeros(sampleBuffer,numObservations);
31 corrTrue = zeros(numObservations,1);
32 vectorPreambDelay = 0;
33 sampleBitBuffer = zeros(1,numObservations);
34 bitDemodBuffer = zeros(1,numObservations);
35 bitDemodPrev = zeros(1,numObservations);
36 bitDemod = zeros(1,numObservations);
37 dataBits = round(secondsData/bitTime);
38 signChanged = false;
39 demodMajority = false;
40 demodPreCombine = false;
41 signalSign = 0;
42 signalSignPrev = 0;
43 vectDemodDelay = zeros(numObservations,1);
44 bitLenght = 0;
45 test = 0;
46 %Uživatelský vstup
47 demodChosen = input('Zvolte metodu vyhodnocení demodulace (0 - Paralelní výpis, 1 - Kombinace majoritou,
48 2 - Kombinace před demodulací: ', "s");
49 %Přiřazení log. hodnot
```

```

50 switch(demodChosen)
51     case '0'
52     case '1'
53         demodMajority = true;
54     case '2'
55         demodPreCombine = true;
56     otherwise
57         error('Zadaná hodnota je chybná.');
```

%Chybný uživatelský vstup

```

58 end
59 %Generování ideálního průběhu preambule
60 for s= 1:samplePreamb
61     switch(arrayFill)
62         case 1
63             vectorPreamb(s) = arrayFill*scaleAmpl;
64             arrayCounterPreamb = arrayCounterPreamb +1;
65             if(arrayCounterPreamb == round(bitTime/sampleTime))
66                 arrayFill = -1;
67                 arrayCounterPreamb = 0;
68             end
69         case -1
70             vectorPreamb(s) = arrayFill*scaleAmpl;
71             arrayCounterPreamb = arrayCounterPreamb +1;
72             if(arrayCounterPreamb == round(bitTime/sampleTime))
73                 arrayFill = 1;
74                 arrayCounterPreamb = 0;
75             end
76         end
77 end
78 %Najde nejkratší nahrávku, od tohoto se odvozuje počet 5s úseků
79 for structElem = 1:numObservations
80     obsElem(structElem) = numel(satnogs.(fieldNames{structElem}));
81 end
82 %Cyklování mezi 5s úseky, jejich načtení a zpracování, buffer
83 for k= 1 : floor(min(obsElem)/sampleBuffer)-1
84     for row = 1:sampleBuffer
85         for col = 1:numObservations
86             matrixBuffer(row,col) = satnogs.(fieldNames{col})(row+sampleBuffer*k);
87         end
88     end
89
90     for corr = 1:numObservations%Zde se porovná, zda vektor ideálního průběhu preambule
91         %koreluje s daným úsekem
92         if(max(xcorr(matrixBuffer(:,corr),vectorPreamb))>corrSensitivity)%Lze nastavit "práh" korelace,
93             %při shodě preambulí se hodnota "corrSensitivity" pohybuje kolem 2800-3000,
94             %lze donastavit na počátku skriptu
95                 corrTrue(corr,1) = 1;
96                 vectorPreambDelay = finddelay(vectorPreamb,matrixBuffer(:,1)); %Sem se ukládají dílčí
97                 %posuvy nahrávek ku preambuli
98             else
99                 corrTrue(corr) = 0;
100             end
101         end
102         if(corrTrue == true(numObservations,1))%Ideální vektor preambule musí souhlasit s preambulemi
103             %ve všech nahrávkách, jestli-že tomu tak není, 5s úsek nebude zpracován a přejde se na další
104
105             %Zarovnání 5s úseku všech nahrávek (zpoždění je počítáno vzhledem k první nahrávce)
106             for dataShift = 1 : numObservations
107                 if(dataShift == 1)
108                     matrixBuffer(:,1) = circshift(matrixBuffer(:,1),-vectorPreambDelay);
109                 else
110                     vectDemodDelay(dataShift)=finddelay(matrixBuffer(:,dataShift-1),matrixBuffer(:,dataShift));
111                     matrixBuffer(:,dataShift) = circshift(matrixBuffer(:,dataShift),-vectDemodDelay(dataShift));
112                     if -vectDemodDelay(dataShift)>0
113                         matrixBuffer(1:-vectDemodDelay(dataShift),dataShift) = 0;
114                     else
115                         matrixBuffer(end+(-vectDemodDelay(dataShift))+1:end,dataShift) = 0;
116                     end
117                 end
118             end
119             if(demodPreCombine)
120                 for rowPre = 1:sampleBuffer
121                     matrixBuffer(rowPre,1) = sum(matrixBuffer(rowPre,:));
122                 end

```

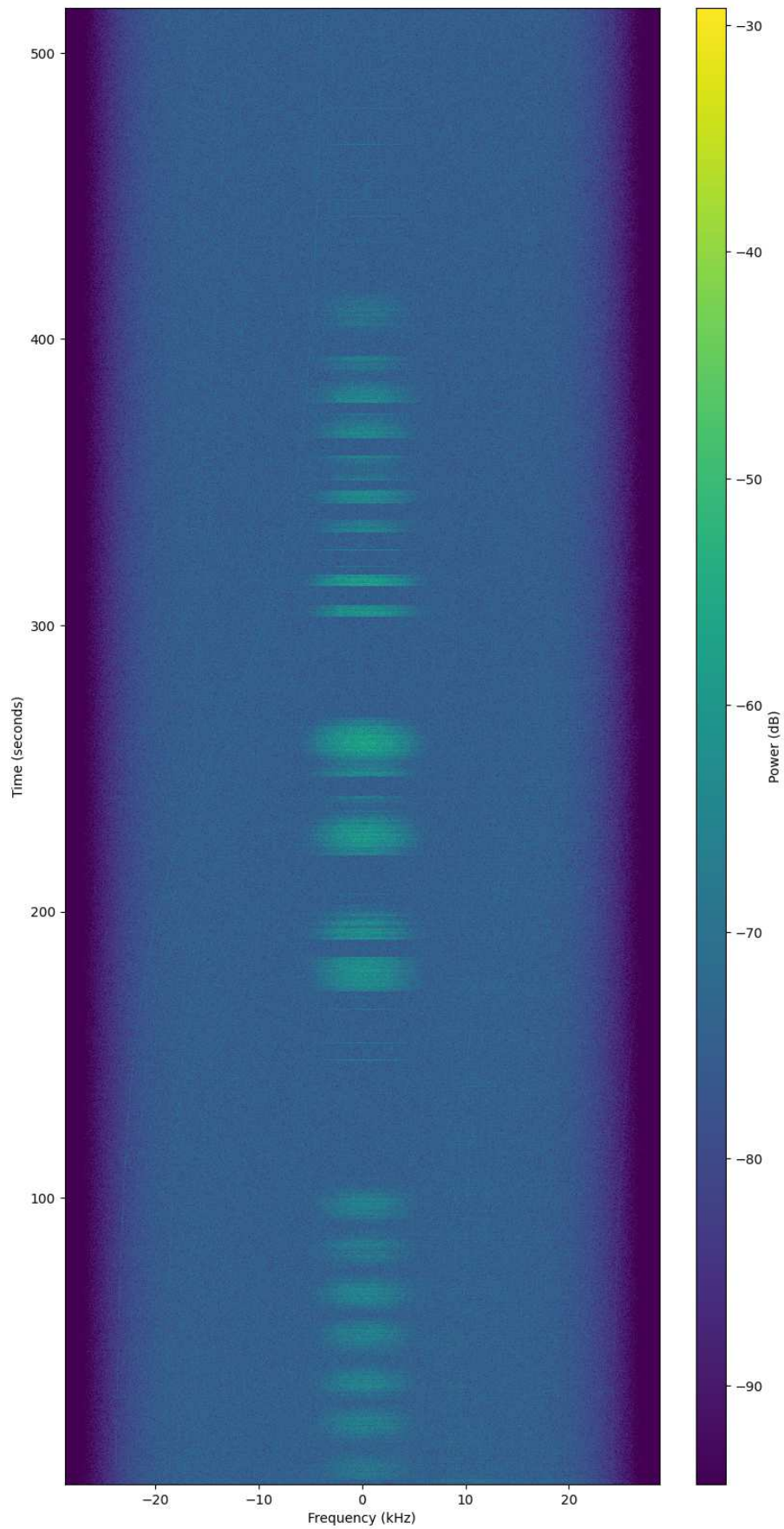
```

123     end
124     for bitArray = 1:numObservations%Postupné načítání hodnot z bufferu pro jednotlivé nahrávky
125         while(arrayCounterData < dataBits)%Cykl trvá, dokud se nenačte
126             %celý datový rámec (kromě preamble)
127             while(~signChanged)%Po změně znaménka dojde k bitovému vyhodnocení,
128                 %nejdříve je plně zpracována jedna nahrávka, poté se přejde na další
129                 sampleBitBuffer(bitBufferCounter,bitArray)=matrixBuffer(matrixCounter+samplePreamb,bitArray);
130                 %5s úsek vychází na 240000 vzorků, program počítá s tím, že buffer tuto hodnotu nepřesáhne
131                 %(všechna data by se tedy měla vejít do dílčích 5s úseků), zatím není nijak ošetřeno
132                 signalSign = sign(sampleBitBuffer(bitBufferCounter,bitArray));
133                 matrixCounter = matrixCounter + 1;
134                 bitBufferCounter = bitBufferCounter+1;
135                 if(~(signalSign == signalSignPrev))
136                     signChangeCounter = signChangeCounter+1; %Indikuje změnu znaménka
137                 end
138                 signalSignPrev=signalSign;
139                 if(signChangeCounter == 2)%Znaménko (polarita) se musí dvakrát změnit
140                     signChanged = true;
141                     bitBufferCounter = 1;
142                     bitLenght = floor(numel(sampleBitBuffer(:,bitArray))*(bitRate/sampleRate));%Zde se spočítá,
143                     %kolik bitů se zapíše za sebou
144                     matrixCounter = matrixCounter - 1;
145                     signChangeCounter = 0;
146                 end
147             end
148             for bitWrite = 1:bitLenght%Cykl, který vyhodnocuje hodnoty mezi změnou polarit
149                 if(mean(sampleBitBuffer(:,bitArray)) > 0)
150                     bitDemodBuffer(arrayCounterData,bitArray) = 1;%1
151                 else
152                     bitDemodBuffer(arrayCounterData,bitArray) = 0;%0
153                 end
154                 arrayCounterData = arrayCounterData + 1;
155             end
156             sampleBitBuffer = zeros(1,numObservations);
157             signChanged = false;
158             signalSign = 0;
159             signalSignPrev = 0;
160         end
161         arrayCounterData = 1;
162         matrixCounter = 1;
163     end
164     bitDemod = cat(1,bitDemodPrev,bitDemodBuffer); %Výsledný bitový tok ze všech nahrávek,
165     %které prošly korelací s preambulí, připojené postupně za sebe
166     bitDemodPrev = bitDemod;
167     bitDemodBuffer = zeros(1,numObservations);
168     if(demodMajority) %Upravení demodulační matice výsledných bitových toků pro majoritní vyhodnocení
169         for bitMajorityDemod = 1 : numel(bitDemod(:,1))
170             bitDemod(bitMajorityDemod,1)=mode(bitDemod(bitMajorityDemod,:));
171         end
172     end
173     if(demodPreCombine || demodMajority) %Upravení matice výsledného bitového toku pro kombinací
174     %před demodulací nebo majoritou
175         bitDemod = bitDemod(:,1);
176     end
177 end
178 end

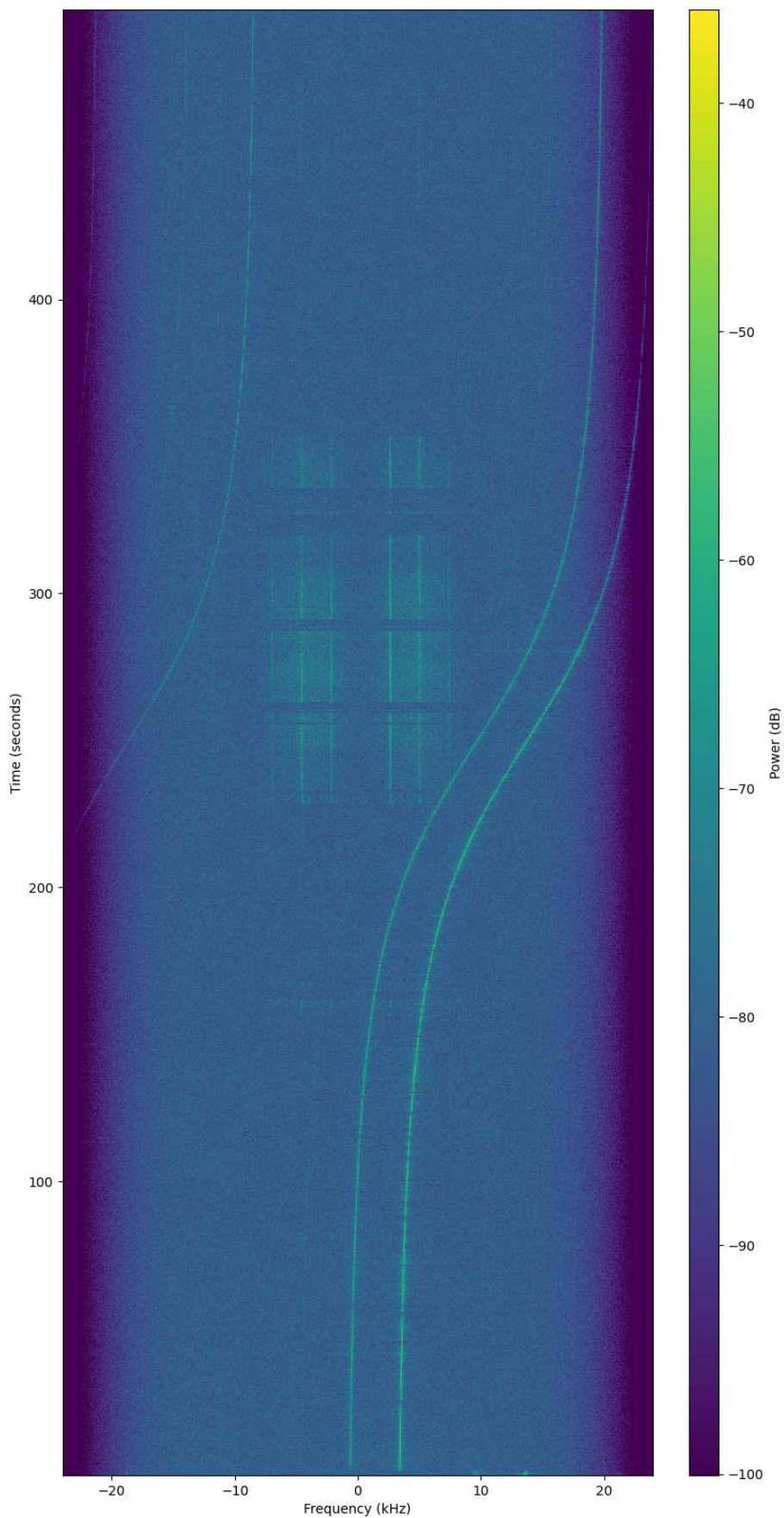
```

Příloha B

Spektrogramy zkušebních observací



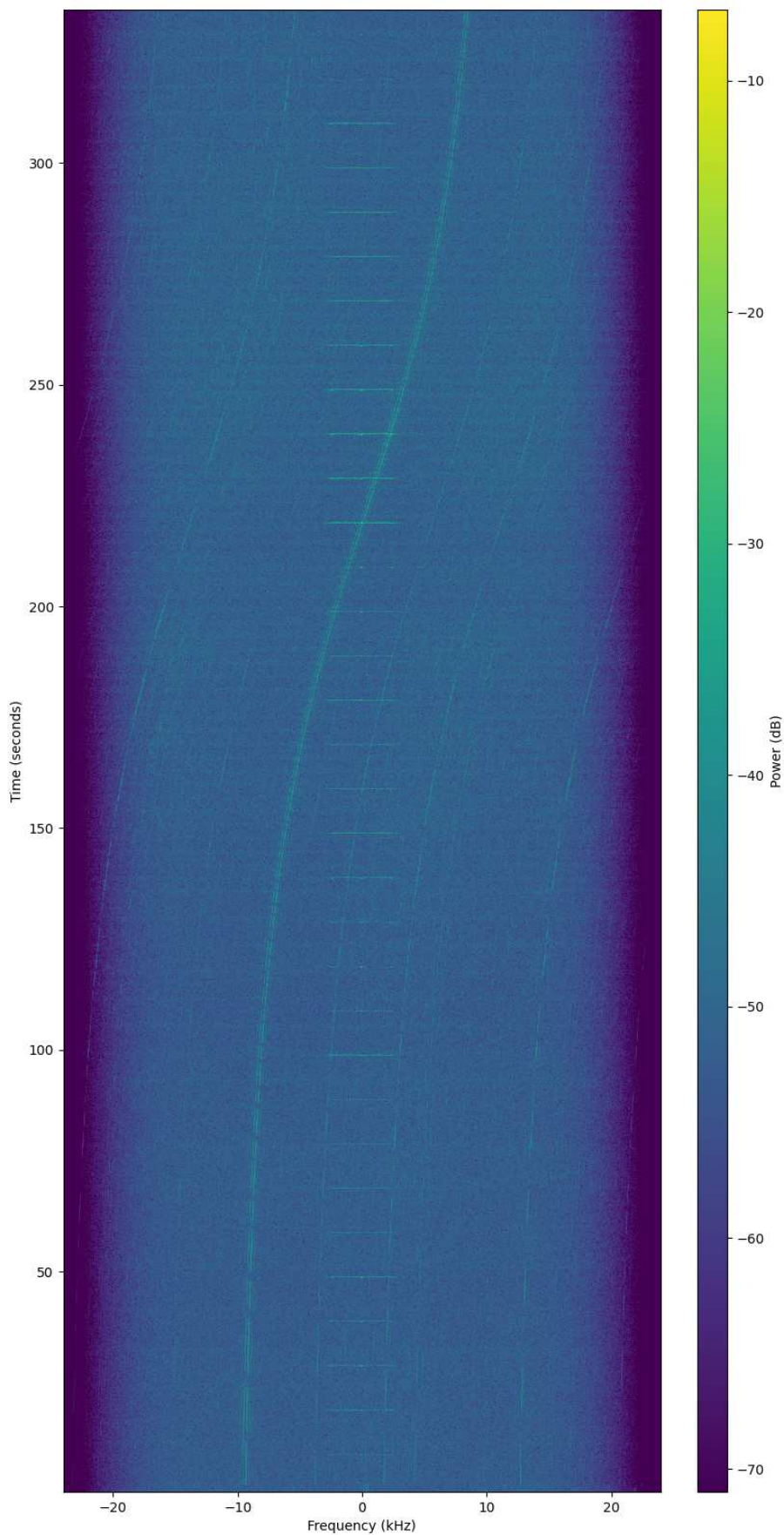
Obr. B.1: Spektrogram satelitu GRBAlpha zachycený pozemní stanicí 2405-PlzenFEL
Číslo observace: 4882705, převzato z [23]



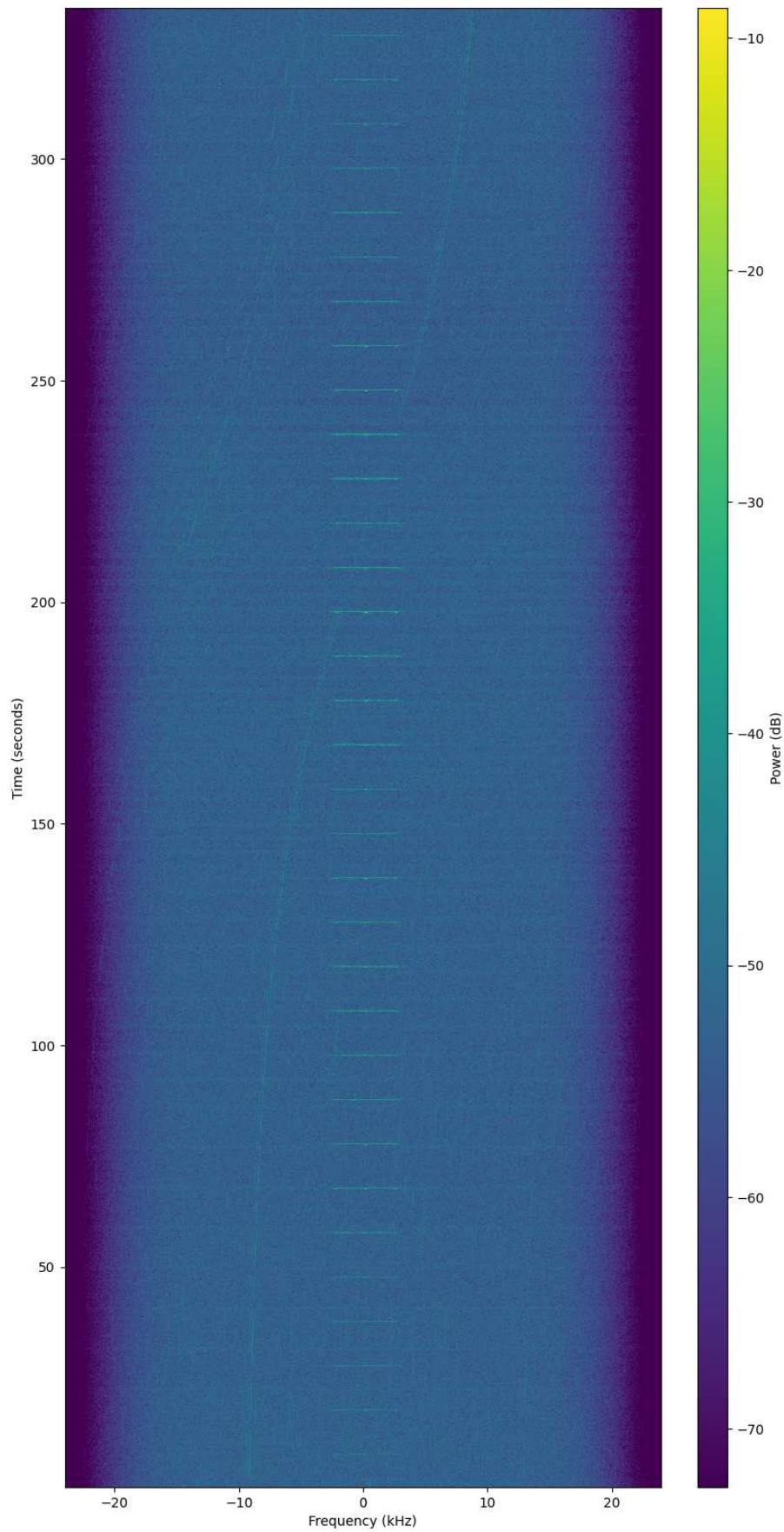
Obr. B.2: Spektrogram satelitu Lucky-7 zachycený pozemní stanicí 2405-PlzenFEL
Číslo observace: 5042953, převzato z [24]

Příloha C

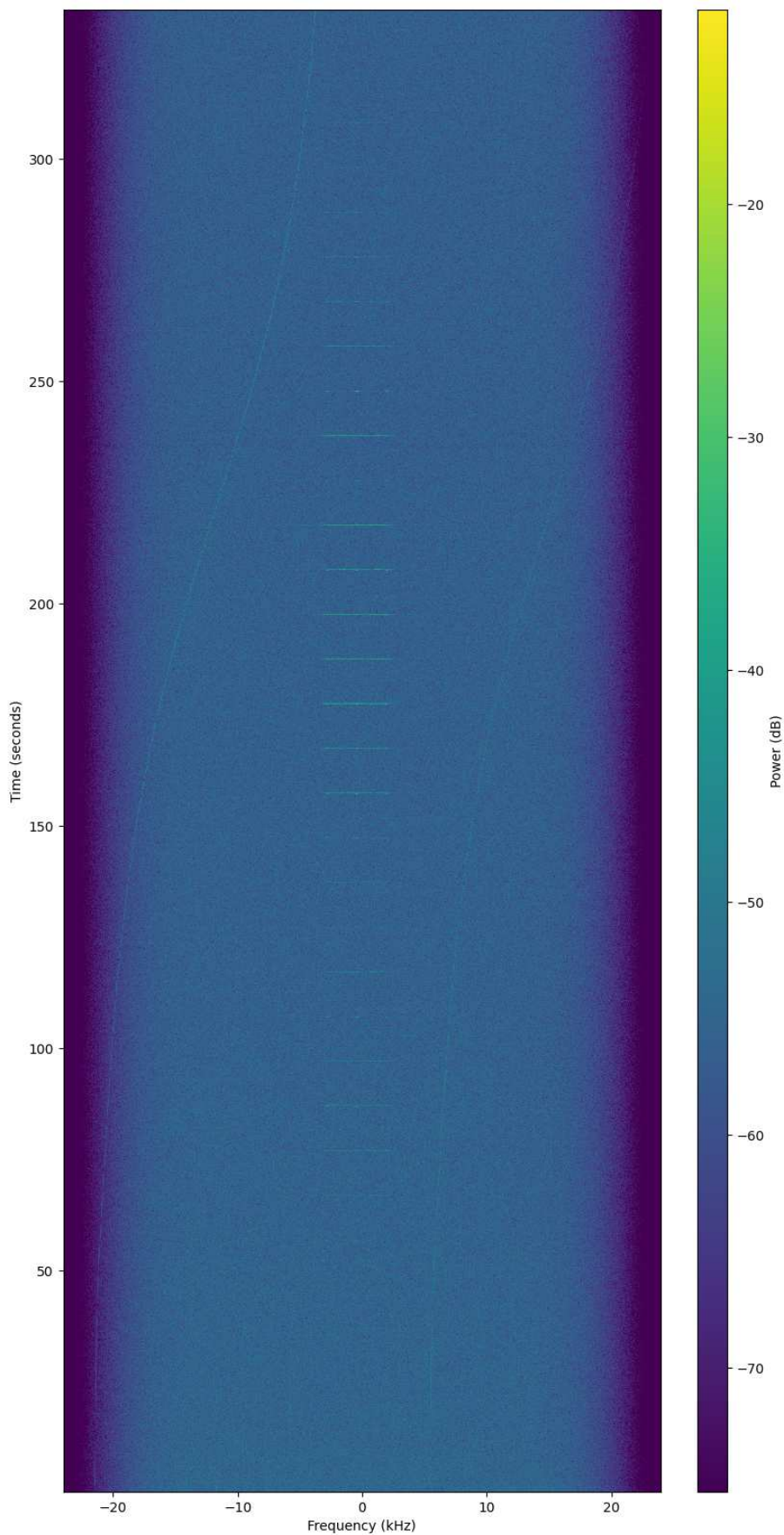
Spektrogramy pro kooperativní příjem VZLUSAT-2



Obr. C.1: Spektrogram satelitu VZLUSAT-2 zachycený pozemní stanicí 1675-MAUSyagi
Číslo observace: 5477367, převzato z [25]



Obr. C.2: Spektrogram satelitu VZLUSAT-2 zachycený pozemní stanicí 2134-MAUSyagi
Číslo observace: 5477368, převzato z [26]



Obr. C.3: Spektrogram satelitu VZLUSAT-2 zachycený pozemní stanicí 2510-MAUSyagi
Číslo observace: 5477369, převzato z [27]