



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIOENGINEERING

SOFTWAREVÝ PŘIJÍMAČ GNSS

SOFTWARE GNSS RECEIVER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Jedlička

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Filip Záplata, Ph.D.

BRNO 2020

Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Bc. Petr Jedlička

ID: 173667

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Softwarový přijímač GNSS

POKYNY PRO VYPRACOVÁNÍ:

Důkladně prostudujte volně dostupné signály GNSS. Zaměřte se především na signály systémů GPS a Galileo. Prostudujte koncepty přijímačů satelitní navigace a vytvořte podrobný koncept softwarové verze takového přijímače. Koncept rozpracujte v Matlabu a vytvořte zde simulaci přijímače. Hlavní pozornost věnujte druhé fázi příjmu, tj. sledování (tracking) již získaných signálů z družic.

Přijímač v Matlabu převedte s využitím specializovaných algoritmů a operací s pevnou desetinnou čárkou do formy vhodné pro implementaci na FPGA. Postupně funkci přijímače ověřte na dostupné vývojové desce s vybraným FPGA. Testy provádějte v součinnosti s PC. Proveďte testy a vyhodnocení práce s reálnými navzorkovanými signály.

DOPORUČENÁ LITERATURA:

[1] NURMI, Jari, Elena Simona LOHAN, Stephan SAND a Heikki HURSKAINEN, ed. GALILEO Positioning Technology [online]. Dordrecht: Springer Netherlands, 2015 [cit. 2018-05-11]. Signals and Communication Technology. ISBN 978-94-007-1829-6.

[2] Understanding GPS: principles and applications. 2nd ed. Editor Elliott D. KAPLAN, editor C. HEGARTY. Boston: Artech House, c2006. ISBN 978-1-58053-894-7.

Termín zadání: 3.2.2020

Termín odevzdání: 28.5.2020

Vedoucí práce: Ing. Filip Záplata, Ph.D.

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Diplomová práce se zabývá analýzou a příjmem volně dostupných signálů navigačních družic v pásmech L1 a E1 pro systémy GPS a Galileo. Mezi etapy příjmu signálu, o kterých tato práce pojednává, patří proces nalezení fáze rozprostíracího kódu a přibližného frekvenčního posuvu nosné frekvence, synchronizace a sledování frekvence a fáze nosného kmitočtu, sledování fáze rozprostíracího kódu, demodulace signálu a kanálové dekódování získaných symbolů. Simulace celého přijímače je provedena v programu MATLAB. Největší pozornost je věnována frekvenční a fázové synchronizaci a sledování nosné frekvence, kdy je většinou uvedeno i více způsobů řešení daných problémů a jejich následné porovnání. Část, která se zabývá synchronizací a následným sledováním frekvence a fáze nosné frekvence a sledováním rozprostíracího kódu, je implementovaná i na FPGA.

Klíčová slova

GNSS, frekvenční synchronizace nosné, fázová synchronizace nosné, GPS, Galileo, FPGA, VHDL.

Abstract

The thesis deals with the analysis and the reception of the freely available signals of the navigation satellites in the L1 and E1 bands of the GPS and Galileo systems. The described signal reception sections include the process of the acquisition, the carrier frequency and phase synchronization and tracking, the spreading code phase tracking, the signal demodulation and the channel decoding. The simulation of the entire receiver is performed in MATLAB. The deeply analyzed signal reception component is the one responsible for the carrier phase and frequency synchronization and tracking. In that case, more methods and their comparison are usually listed. The signal reception component, which is responsible for the carrier phase and frequency tracking and the spreading code phase tracking, is also implemented in FPGA.

Keywords

GNSS, carrier frequency synchronization, carrier phase synchronization, GPS, Galileo, FPGA, VHDL.

JEDLIČKA, Petr. Softwarový přijímač GNSS. Brno, 2020, 83 stran. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/126117>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce Filip Záplata.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Softwarový přijímač GNSS jsem vypracoval samostatně pod vedením vedoucí/ho diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....
podpis autora

Děkuji vedoucímu diplomové práce Ing. Filipu Záplatovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....

podpis autora

Obsah

Seznam obrázků	VIII
Seznam tabulek	XI
Úvod	1
1 Družicové navigační systémy	2
1.1 Systém GPS	2
1.2 Systém Galileo	6
1.3 Ostatní navigační satelitní systémy	9
2 Vývoj softwarového přijímače	10
2.1 Nalezení navigačního signálu	10
2.1.1 Akvizice s postupným hledáním	10
2.1.2 Akvizice s paralelním hledáním fáze kódu	11
2.1.2.1 Akvizice signálu GPS	12
2.1.2.2 Akvizice signálu Galileo	13
2.2 Sledování rozprostíracího kódu	14
2.3 Sledování nosného kmitočtu	18
2.3.1 Fázová synchronizace	18
2.3.1.1 Costasova smyčka – popis	18
2.3.1.2 Costasova smyčka – návrh	20
2.3.1.3 Costasova smyčka - simulace	22
2.3.1.4 Smyčka fázového závěsu	24
2.3.2 Frekvenční synchronizace	26
2.3.2.1 Synchronizace pomocí FFT	26
2.3.2.2 Synchronizace pomocí derivace fáze	27
2.3.2.3 Frekvenční estimátor ML	30
2.3.2.4 Kvadrikorelátor	33
2.3.3 Volba metody	35
2.4 Detekce synchronizace	35
2.4.1 Detekce frekvenční synchronizace	36

2.4.2	Detekce fázové synchronizace	37
2.5	Bitová synchronizace	39
2.6	Kanálové dekodování	39
2.6.1	Signál L1 C/A	39
2.6.2	Signál E1-OS	41
3	Implementace softwarového přijímače na FPGA	44
3.1	Struktura softwarového přijímače	44
3.1.1	MAX2771	46
3.1.2	Data FIFO	47
3.1.3	Fázový akumulátor	48
3.1.4	DDS	53
3.1.5	IQ & DDS synchronizátor	54
3.1.6	Komplexní násobička	54
3.1.7	EPL korelátor	55
3.1.8	Změna hodinové domény	57
3.1.9	Diskriminátory a smyčkový filtr	57
3.2	Maximální počet kanálů	61
3.3	Využití hardwarových prostředků	64
4	Verifikace hardwarové implementace	66
4.1	Signál L1 C/A	66
4.1.1	Frekvenční sledování	67
4.1.2	Fázové sledování	70
4.2	Signál E1-OS	73
4.2.1	Frekvenční sledování	73
4.2.2	Fázové sledování	75
	Závěr	78
	Literatura	80
	Seznam symbolů, veličin a zkratk	82

Seznam obrázků

Obr. 1.1 Blokové schéma generátoru navigačního C/A signálu.	4
Obr. 1.2 Blokové schéma generátoru C/A kódu.	5
Obr. 1.3 Korelační vlastnosti C/A kódů.	6
Obr. 1.4 Generování rozprostíracího kódu pro pilotní signál.	7
Obr. 1.5 Časové průběhy CBOC subnosných - a) datový signál, b) pilotní signál.	9
Obr. 2.1 Akvizice s paralelním hledáním fáze rozprostírací sekvence.	12
Obr. 2.2 Akvizice L1 C/A signálu.	13
Obr. 2.3 Blokové schéma DLL smyčky.	14
Obr. 2.4 Grafy korelací rozprostíracích sekvencí.	15
Obr. 2.5 Závislost chybového signálu na rozdílu fáze přijímané a lokální sekvence pro signál L1 C/A.	16
Obr. 2.6 Blokové schéma Costasovy smyčky pro komplexní BPSK signál.	18
Obr. 2.7 Chybový signál fázového diskriminátoru.	19
Obr. 2.8 Smyčkový filtr.	20
Obr. 2.9 Frekvenční posuv sledovaný Costasovou smyčkou, L1 C/A.	22
Obr. 2.10 Frekvenční posuv sledovaný Costasovou smyčkou, L1 C/A, detail.	23
Obr. 2.11 Frekvenční posuv sledovaný Costasovou smyčkou, E1-OS.	24
Obr. 2.12 Chybový signál fázového diskriminátoru PLL.	25
Obr. 2.13 Frekvenční posuv sledovaný PLL smyčkou, E1-OS.	25
Obr. 2.14 Porovnání Costasovy smyčky a PLL smyčky.	26
Obr. 2.15 Frekvenční spektrum L1 C/A signálu s frekvenčním posuvem.	27
Obr. 2.16 Frekvenční synchronizace s derivací fáze, L1 C/A.	29
Obr. 2.17 Frekvenční synchronizace s derivací fáze, E1-OS.	29
Obr. 2.18 Fázový diskriminátor frekvenčního estimátoru ML.	30
Obr. 2.19 S - křivka pro délku impulsní charakteristiky 20, L1 C/A.	31
Obr. 2.20 S - křivka pro délku impulsní charakteristiky 7, L1 C/A.	31
Obr. 2.21 Sledování frekvence s ML estimátorem, L1 C/A.	32
Obr. 2.22 Sledování frekvence s ML estimátorem, E1-OS.	33
Obr. 2.23 Sledování frekvence kvadrikorelátorem, E1-OS.	34
Obr. 2.24 Porovnání Kvadrikorelátoru a ML estimátoru, E1-OS.	35

Obr. 2.25 Detekce frekvenční synchronizace, E1-OS.....	36
Obr. 2.26 Detekce frekvenční synchronizace, L1 C/A.....	37
Obr. 2.27 Detekce fázové synchronizace, E1-OS.....	38
Obr. 2.28 Blokové schéma konvolučního kodéru, E1-OS.	41
Obr. 3.1 Blokové schéma přijímacího modulu	45
Obr. 3.2 Zjednodušené blokové schéma fázového akumulátoru	48
Obr. 3.3 Zjednodušené blokové schéma EPL korelátoru	55
Obr. 3.4 Zjednodušené blokové schéma bloku diskriminátorů a smyčkového filtru.....	57
Obr. 3.5 Závislost minimální hodnoty clk H na hodnotě clk L u signálu L1 C/A.....	63
Obr. 3.6 Závislost minimální hodnoty clk H na hodnotě clk L u signálu E1-OS.....	63
Obr. 4.1 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz	67
Obr. 4.2 Družice číslo 7, vlevo MATLAB, vpravo FPGA, +90 Hz	67
Obr. 4.3 Družice číslo 10, vlevo MATLAB, vpravo FPGA, +90 Hz	67
Obr. 4.4 Družice číslo 15, vlevo MATLAB, vpravo FPGA, +90 Hz	68
Obr. 4.5 Družice číslo 16, vlevo MATLAB, vpravo FPGA, -90 Hz	68
Obr. 4.6 Družice číslo 20, vlevo MATLAB, vpravo FPGA, -90Hz	68
Obr. 4.7 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz	69
Obr. 4.8 Družice číslo 26, vlevo MATLAB, vpravo FPGA, -90 Hz	69
Obr. 4.9 Družice číslo 29, vlevo MATLAB, vpravo FPGA, +90 Hz	69
Obr. 4.10 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz	70
Obr. 4.11 Družice číslo 7, vlevo MATLAB, vpravo FPGA, +90 Hz	70
Obr. 4.12 Družice číslo 10, vlevo MATLAB, vpravo FPGA, +90 Hz	70
Obr. 4.13 Družice číslo 15, vlevo MATLAB, vpravo FPGA, +90 Hz	71
Obr. 4.14 Družice číslo 16, vlevo MATLAB, vpravo FPGA, -90 Hz	71
Obr. 4.15 Družice číslo 20, vlevo MATLAB, vpravo FPGA, -90 Hz	71
Obr. 4.16 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz	72
Obr. 4.17 Družice číslo 26, vlevo MATLAB, vpravo FPGA, -90 Hz	72
Obr. 4.18 Družice číslo 29, vlevo MATLAB, vpravo FPGA, +90 Hz	72
Obr. 4.19 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz	73
Obr. 4.20 Družice číslo 5, vlevo MATLAB, vpravo FPGA, +90 Hz	73
Obr. 4.21 Družice číslo 9, vlevo MATLAB, vpravo FPGA, -90 Hz	74
Obr. 4.22 Družice číslo 11, vlevo MATLAB, vpravo FPGA, +90 Hz	74

Obr. 4.23 Družice číslo 14, vlevo MATLAB, vpravo FPGA, -90 Hz	74
Obr. 4.24 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz	75
Obr. 4.25 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz	75
Obr. 4.26 Družice číslo 5, vlevo MATLAB, vpravo FPGA, +90 Hz	75
Obr. 4.27 Družice číslo 9, vlevo MATLAB, vpravo FPGA, -90 Hz	76
Obr. 4.28 Družice číslo 11, vlevo MATLAB, vpravo FPGA, +90 Hz	76
Obr. 4.29 Družice číslo 14, vlevo MATLAB, vpravo FPGA, -90 Hz	76
Obr. 4.30 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz	77

Seznam tabulek

Tabulka 1.1 Navigační signály GPS (počty družic aktuální pro listopad 2019).....	3
Tabulka 1.2 Frekvenční pásma systému Galileo.	6
Tabulka 2.1 Kontrola parity, L1 C/A.	40
Tabulka 2.2 Nalezené družice, L1 C/A.	41
Tabulka 2.3 Nalezené družice, E1-OS.	43
Tabulka 3.1 Položky M_AXIS PHASE TUSER sběrnice fázového akumulátoru	49
Tabulka 3.2 Kanálové registry fázového akumulátoru	51
Tabulka 3.3 Položky sběrnice M_AXIS CTRL TDATA	60
Tabulka 3.4 Využití hardwarových prostředků, L1 C/A	64
Tabulka 3.5 Využití hardwarových prostředků, E1-OS	65

Úvod

Globální navigační satelitní systémy (GNSS) nacházejí své uplatnění v mnoha odvětvích lidské činnosti, jako jsou například doprava, zemědělství nebo geodézie, přičemž přesnost těchto systémů se, v závislosti na aplikaci, pohybuje od jednotek metrů až po jednotky centimetrů. Za účelem lepší přesnosti a spolehlivosti jsou GNSS systémy stále předmětem vývoje. Zavedené systémy jako GPS (*Global Positioning System*) uvádí do provozu nové modernizované navigační signály. Zároveň probíhá i vývoj zcela nových GNSS systémů, jako je například systém Galileo. Aby ovšem bylo možné využívat výhod moderních systémů, je nutný vývoj i moderních GNSS přijímačů.

Tato práce se zabývá vývojem, simulací a hardwarovou implementací algoritmů funkčních bloků, které jsou nedílnou součástí softwarově definovaných přijímačů volně dostupných signálů GPS a Galileo v pásmu L1 (1575,42 MHz). Veškeré simulace jsou provedeny v prostředí MATLAB s ohledem na následnou implementaci do systému pracujícím v reálném čase. Hardwarová implementace je provedena s využitím vývojové desky osazené obvodem FPGA.

První kapitola představuje teoretický úvod do družicových navigačních systémů. Jedná se především o základní popis těchto systémů a způsob generování navigačních signálů.

Ve druhé kapitole jsou v rámci jednotlivých podkapitol rozpracovány funkční bloky softwarově definovaného přijímače. Je popsán proces hledání přibližné hodnoty frekvenčního posuvu nosné frekvence a hodnoty fáze rozprostíracího kódu, dále je popsáno několik způsobů synchronizace a sledování frekvence a fáze nosné frekvence, sledování fáze rozprostíracího kódu a kanálové dekódování se získáním datových bitů, přičemž největší pozornost je věnována synchronizaci a sledování.

Třetí kapitola pojednává o hardwarové implementaci vybraných funkčních bloků na vývojovou desku s FPGA. Jsou implementovány funkční bloky přijímače, které mají na starosti synchronizaci a sledování nosného kmitočtu a sledování fáze rozprostírací sekvence, což jsou také části přijímacího řetězce, které mají největší výpočetní náročnost.

Ověření správné implementace přijímače na FPGA, včetně porovnání výsledků dosažených při simulacích v MATLABu a výsledků hardwarové implementace, je rozebráno ve čtvrté kapitole.

1 Družicové navigační systémy

Obecně je struktura každého družicového navigačního systému tvořena třemi částmi [1]:

- pozemní segment
- kosmický segment
- uživatelský segment.

Pozemní segment se skládá z řídicích a sledovacích stanic rozmístěných na Zemi a má za úkol dohlížet na chod celého systému. Sledovací stanice pozemního segmentu monitorují, mimo jiné, pohyb po oběžné dráze, polohu a funkční stav jednotlivých družic systému. Na základě získaných informací jsou prostřednictvím řídicích stanic aktualizovány odpovídající údaje na jednotlivých družicích a případně korigován jejich pohyb tak, aby byly zachovány jejich požadované oběžné dráhy. Jednou z dalších důležitých funkcí pozemního segmentu je udržovat celý systém časově synchronní. [1]

Kosmický segment je tvořen konstelací družic obíhajících kolem Země po definovaných oběžných drahách. Jednotlivé družice vysílají v pravidelných intervalech informace týkající se především jejich pohybu, polohy a času, přičemž každá z družic vysílá průběžně tyto informace i o ostatních družicích daného systému. [1]

Uživatelským segmentem jsou míněny uživatelské přijímače navigačních signálů družic. Výpočet polohy těchto přijímačů většinou vychází z přijatých informací o aktuálních polohách družic a z přijímačem změřených časových rozdílů okamžiků přijetí časových značek jednotlivých družic. Pro správný výpočet polohy je nutné, aby byly všechny družice systému časově synchronní, a aby všechny družice vyslaly časovou značku ve stejný okamžik. V případě, že by byl i přijímač časově synchronní s daným navigačním systémem, k výpočtu polohy by stačil signál od tří družic, avšak takové situace není prakticky možné dosáhnout, a proto jsou k výpočtu polohy přijímače potřeba minimálně čtyři družice daného systému. Časovou značkou se rozumí například počátek vyslání konkrétního datového paketu. [1][2]

V případě, že je vysílač vzhledem k přijímači v pohybu, dochází k tzv. Dopplerovému frekvenčnímu posuvu, který se projevuje příjmem frekvence, která se liší od skutečně vysílané frekvence. Velikost frekvenčního posuvu je přímo úměrná vysílané frekvenci a zároveň roste s rostoucí velikostí rychlosti pohybu. Dochází-li ke vzájemnému přibližování obou objektů, frekvenční posuv má kladnou hodnotu a naopak. [3] Vzhledem k tomu, že i navigační družice mají při svém obíhání kolem Země nenulovou rychlost ve vztahu k přijímačům na Zemi, dochází k tomuto jevu i u družicové navigace a je úkolem uživatelského segmentu se s touto skutečností vypořádat.

1.1 Systém GPS

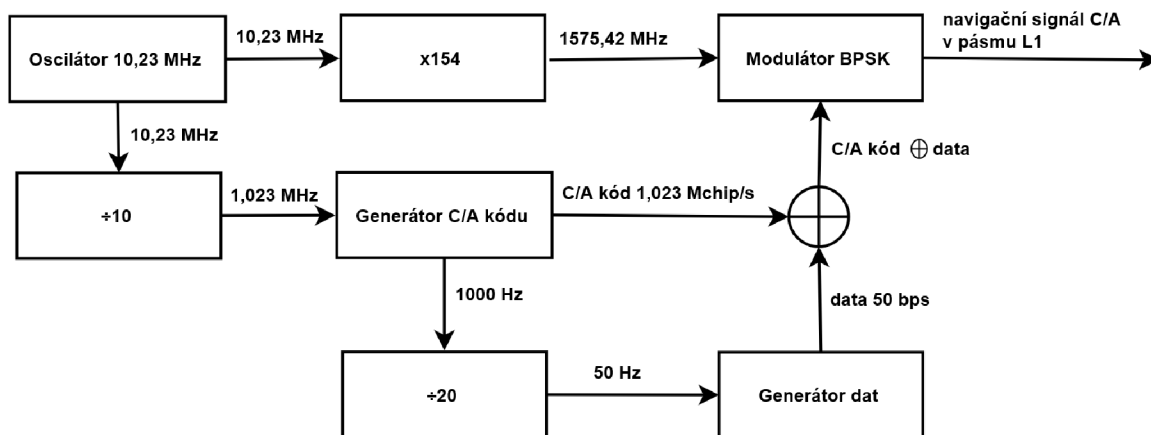
Systém GPS je globální navigační satelitní systém vlastněn a spravován Spojenými státy americkými. Družice se pohybují na šesti kruhových oběžných drahách s inklinací

55° ve výšce přibližně 20200 kilometrů nad Zemí [4]. Ve smyslu délky vzestupného uzlu se sousední oběžné dráhy liší o 60° [4]. Plně operační systém je tvořen minimálně 24 družicemi rovnoměrně rozmístěnými na výše uvedených oběžných drahách [4]. V době vzniku této práce (listopad 2019) je konstelace tvořena 31 různě starými družicemi s různým počtem podporovaných navigačních signálů [5]. Systém se skládá z celkem 6 navigačních signálů ve 3 frekvenčních pásmech. Tyto signály jsou uvedeny v tabulce 1.1 [5][22]. Pro odlišení navigačních signálů mezi jednotlivými družicemi i v rámci jedné družice jsou použity rozprostírací kódy, metodou vícenásobného přístupu je tedy CDMA (*Code Division Multiple Access*) [6][8].

Tabulka 1.1 Navigační signály GPS (počty družic aktuální pro listopad 2019).

Označení signálu	Pásmo [MHz], (označení pásma)	Poznámka
C/A	1575,42 (L1)	původní volně dostupný signál, k dispozici na všech družicích
L1C	1575,42 (L1)	modernizovaný volně dostupný signál, zatím není k dispozici
L2C	1227,60 (L2)	modernizovaný volně dostupný signál, zatím k dispozici pouze na 19 družicích
L5	1176,45 (L5)	modernizovaný volně dostupný signál, zatím k dispozici pouze na 12 družicích
P(Y)	1575,42 (L1) 1227,60 (L2)	původní vojenský signál s autorizovaným přístupem, k dispozici na všech družicích
M	1575,45 (L1) 1227,60 (L2)	modernizovaný vojenský signál s autorizovaným přístupem, zatím k dispozici pouze na 19 družicích

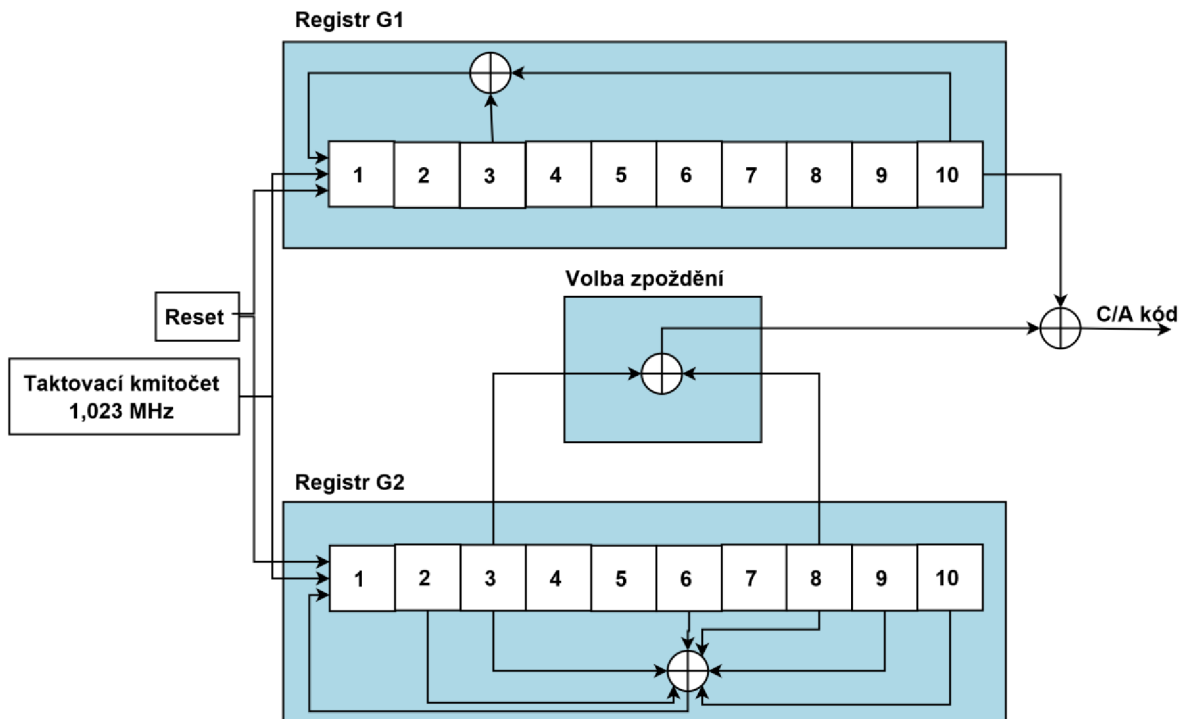
Jak plyne z tabulky 1.1, v době vzniku této práce se pro pásmo L1 nachází v plné operační dostupnosti pouze jeden volně dostupný signál, rozprostíraný kódem C/A (*Coarse Acquisition*) na frekvenci L1 (L1 C/A). Tímto navigačním signálem se zabývá i tato práce, a proto je rozebrán podrobněji. Zjednodušené blokové schéma generování tohoto signálu je na obrázku 1.1 [2].



Obr. 1.1 Blokové schéma generátoru navigačního C/A signálu.

Základní frekvencí, ze které jsou odvozeny nosné kmitočty i hodinové signály generátorů rozprostíracích kódů, je 10,23 MHz. Nosná frekvence L1 se nachází na výstupu násobičky základní frekvence s koeficientem 154. Generátor rozprostíracího C/A kódu je taktován hodinovým signálem, jehož frekvence odpovídá desetina frekvence základní, C/A kód má tedy rychlost 1023000 chipů za sekundu, přičemž jedna perioda C/A kódu má 1023 chipů, a tím pádem trvá 1 milisekundu. Bitová rychlost navigačních dat je 50 bitů za sekundu, z čehož plyne, že jeden datový bit je rozprostřen 20 periodami C/A kódu. Rozprostřena data jsou před vysláním z družice modulována pomocí BPSK (*Binary Phase Shift Keying*) modulace na nosnou frekvenci L1. [2]

Rozprostírací kódy jsou pseudonáhodné sekvence, které jsou též označovány jako Goldovy sekvence. Generátor rozprostíracích kódů se skládá ze dvou posuvných registrů typu LFSR (Linear Feedback Shift Register), označovaných jako G1 a G2. Samotné rozprostírací kódy představují exkluzivní logický součet (operace XOR) výstupů těchto dvou registrů. Generování různých kódů je dosaženo volbou různých zpoždění výstupu registru G2. Omezeného počtu zpoždění lze dosáhnout vyjádřením výstupu registru G2 exkluzivním logickým součtem dvou definovaných pozic tohoto registru. [2][6] V době vzniku této práce jsou pro L1 C/A signály použity sekvence, které je možné generovat tímto způsobem [7]. Počáteční hodnotou obou registrů jsou logické jedničky na všech pozicích. Systémem GPS jsou rovněž definovány rozprostírací kódy, kterých je možné dosáhnout pouze nastavením hodnoty, odpovídající požadovanému zpoždění, do registru G2 [6]. V době vzniku této práce je však využívána pouze první skupina kódů [7]. Blokové schéma generátoru C/A kódů je na obrázku 1.2 [2].



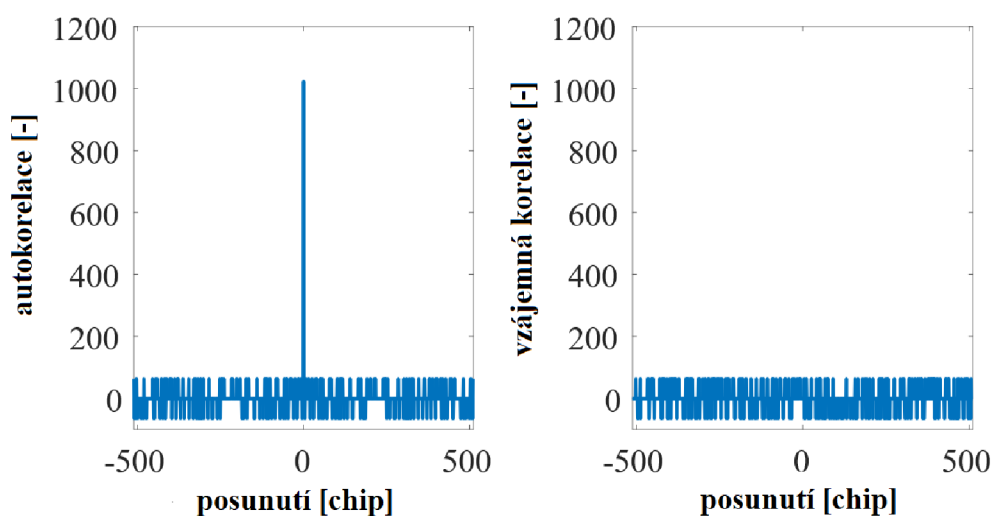
Obr. 1.2 Blokové schéma generátoru C/A kódu.

Zpětné vazby registrů G1 a G2 jsou definovány polynomy vyjádřenými rovnicemi 1.1 a 1.2 [2].

$$f_1(x) = 1 + x^3 + x^{10} \quad (1.1)$$

$$f_2(x) = 1 + x^2 + x^3 + x^6 + x^8 + x^9 + x^{10} \quad (1.2)$$

Goldovy kódy byly pro oddělení jednotlivých družic a navigačních signálů použity kvůli jejich korelačním vlastnostem. Tyto kódy mají mezi sebou velmi nízkou korelaci pro jakékoliv vzájemné posunutí. Autokorelační funkce těchto kódů je rovněž velmi nízká pro všechna nenulová vzájemná posunutí. [2] Příklad cyklické autokorelace a cyklické vzájemné korelace C/A kódů je zobrazen na obrázku 1.3.



Obr. 1.3 Korelační vlastnosti C/A kódů.

1.2 Systém Galileo

Systém Galileo je globální navigační systém vlastněn a vyvíjen Evropskou unií. Družice se pohybují na 3 kruhových oběžných drahách s inklinací 56° ve výšce přibližně 23200 kilometrů nad Zemí. Ve smyslu délky vzestupného uzlu se sousední oběžné dráhy liší o 120° . [11] Systém je plánován na 24 operačních a 6 záložních družic [10]. V době vzniku této práce je dosažení tohoto stavu plánováno na rok 2020 [9]. Přístupovou metodou je, stejně jakou GPS, CDMA [12]. Systém používá 3 frekvenční pásma označována jako E1, E5 a E6, přičemž E5 se dále dělí na E5a a E5b. Frekvence pásem jsou uvedeny v tabulce 1.2 [12]. V pásmech E1 a E5 jsou vysílány volně dostupné navigační signály, též označované jako OS (*Open Service*), a v pásmu E6 se nachází navigační signál s autorizovaným přístupem pro komerční využití, též označován jako CS (*Commercial Service*) [12]. Další služby zahrnuté v systému Galileo jsou PRS (*Public Regulated Service*) a SAR (*Search and Rescue Service*) [11].

Tabulka 1.2 Frekvenční pásma systému Galileo.

Označení pásma	Nosná frekvence [MHz]
E1	1575,420
E6	1278,750
E5	1191,795
E5a	1176,450
E5b	1207,140

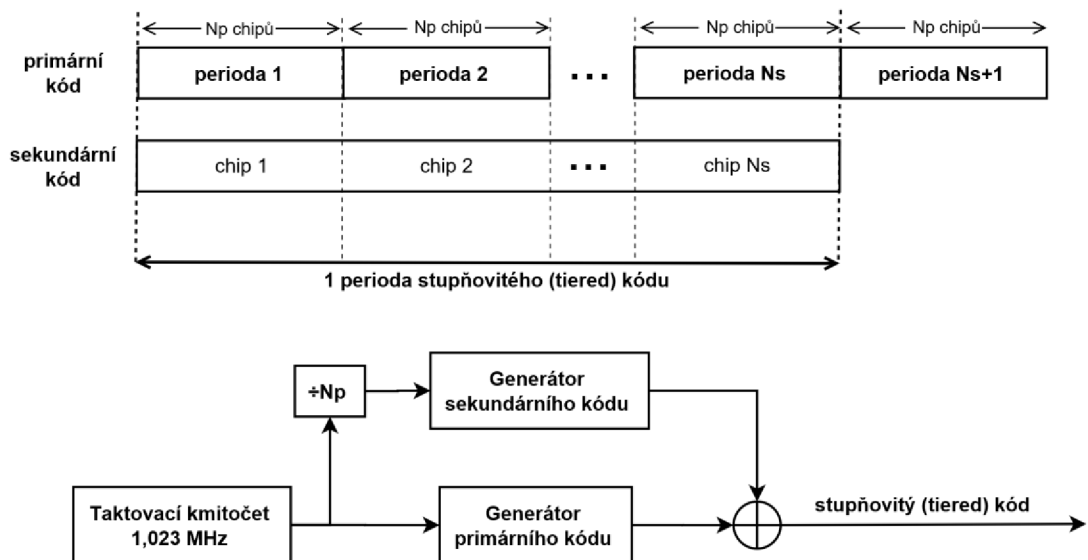
Tato práce se zabývá volně dostupným signálem v pásmu E1 (E1-OS), a proto je v následujících odstavcích rozebrán podrobněji.

Navigační signál E1-OS se skládá ze dvou složek – signálu pilotního (E1-C) a signálu datového (E1-B), přičemž každý z nich představuje 50% výkonu výsledného navigačního signálu na výstupu družice. Oba signály jsou vysílány na stejné frekvenci a jsou vzájemně rozlišeny rozprostíracím kódem. Pilotní signál nese žádnou informaci a může sloužit pouze pro snadnější synchronizaci přijímače. Datový signál s sebou nese i navigační data, synchronizace přijímače je možná i s tímto signálem. [12]

Pro rozprostírání signálů jsou použity tak zvané paměťové kódy, což znamená, že nejsou generovány pomocí LFSR registrů, ale jsou uloženy v paměti, ze které jsou vyčítány. Rozprostírací kódy pro E1-OS jsou tvořeny primárními kódy a jedním sekundárním kódem. Primární kódy jsou generovány rychlostí 1023000 chipů za sekundu, mají délku 4092 chipů a doba jedné periody činí 4 milisekundy. Sekundární kód má délku 25 chipů a dobu trvání jednoho čipu odpovídající době jedné periody primárního kódu. Generování sekundárního kódu je synchronní se začátkem periody primárního kódu. [12] Korelační vlastnosti kódů jsou podobné jako u C/A kódů v GPS.

U datového signálu jsou navigační data rozprostírána pouze primárním kódem. Symbolová perioda odpovídá periodě primárního kódu. Generování symbolů je synchronní s generováním rozprostíracího kódu. [12]

Pilotní signál je tvořen kódem, který je v anglicky psané literatuře označován jako *tiered code* (do češtiny možné přeložit jako stupňovitý kód). Tento rozprostírací kód vznikne rozprostřením sekundárního kódu kódem primárním, generování je znázorněno na obrázku 1.4 [12]. Primární kódy pilotního a datového signálu jsou synchronní.



N_p ... počet chipů 1 periody primárního kódu

N_s ... počet chipů 1 periody sekundárního kódu

Obr. 1.4 Generování rozprostíracího kódu pro pilotní signál.

Pilotní i datový signál jsou každý zvlášť modulovány pomocí subnosné typu CBOC (*Composite Binary Offset Carrier*) tvořené dvěma dílčími subnosnými typu BOC(1, 1) a BOC(6, 1) (*Binary Offset Carrier*), přičemž převážná část výkonu je soustředěna do BOC(1, 1). Výsledný navigační signál E1-OS připravený na modulaci na nosnou frekvenci E1, lze vyjádřit rovnicí 1.3. [12]

$$s_{E1-OS}(t) = \sqrt{\frac{1}{2}}(e_{E1-B}(t) \cdot s_{CBOC,E1-B}(t) - e_{E1-C}(t) \cdot s_{CBOC,E1-C}(t)), \quad (1.3)$$

kde význam proměnných je následující:

- t je čas,
- $e_{E1-B}(t)$ a $e_{E1-C}(t)$ jsou časové průběhy rozprostřeného datového a pilotního signálu,
- $s_{CBOC,E1-B}(t)$ a $s_{CBOC,E1-C}(t)$ jsou časové průběhy subnosných typu CBOC pro datový a pilotní signál, tyto subnosné jsou vyjádřeny rovnicemi 1.4 a 1.5 [12],
- $s_{E1-OS}(t)$ je časový průběh výsledného navigačního signálu připraveného na modulaci na nosnou frekvenci E1.

Koeficient $\sqrt{\frac{1}{2}}$ v rovnici 1.3 má význam výkonového normování na hodnotu 1, protože se ve vztahu nachází suma dvou signálů, z nichž každý má normovaný výkon o hodnotě 1. Koeficient má odmocninu, protože vztah vyjadřuje amplitudu signálu, jejíž druhá mocnina je přímo úměrná výkonu.

$$s_{CBOC,E1-B}(t) = \alpha \cdot s_{BOC(1,1)}(t) + \beta \cdot s_{BOC(6,1)}(t), \quad (1.4)$$

$$s_{CBOC,E1-C}(t) = \alpha \cdot s_{BOC(1,1)}(t) - \beta \cdot s_{BOC(6,1)}(t), \quad (1.5)$$

kde význam proměnných je následující:

- t je čas,
- $s_{CBOC,E1-B}(t)$ a $s_{CBOC,E1-C}(t)$ jsou časové průběhy subnosných typu CBOC pro datový a pilotní signál, průběhy jsou zobrazeny na obrázku 1.5 [12],
- $s_{BOC(1,1)}(t)$ a $s_{BOC(6,1)}(t)$ jsou časové průběhy subnosných typu BOC(1, 1) a BOC(6, 1), subnosné jsou vyjádřeny rovnicemi 1.8 a 1.9 [12],
- koeficienty α a β mají význam, jednak výkonového normování součtu BOC subnosných na hodnotu 1, a jednak určují rozložení výkonu mezi tyto subnosné. Hodnoty těchto koeficientů jsou dány rovnicemi 1.6 a 1.7 [12].

$$\alpha = \sqrt{\frac{10}{11}} \quad (1.6)$$

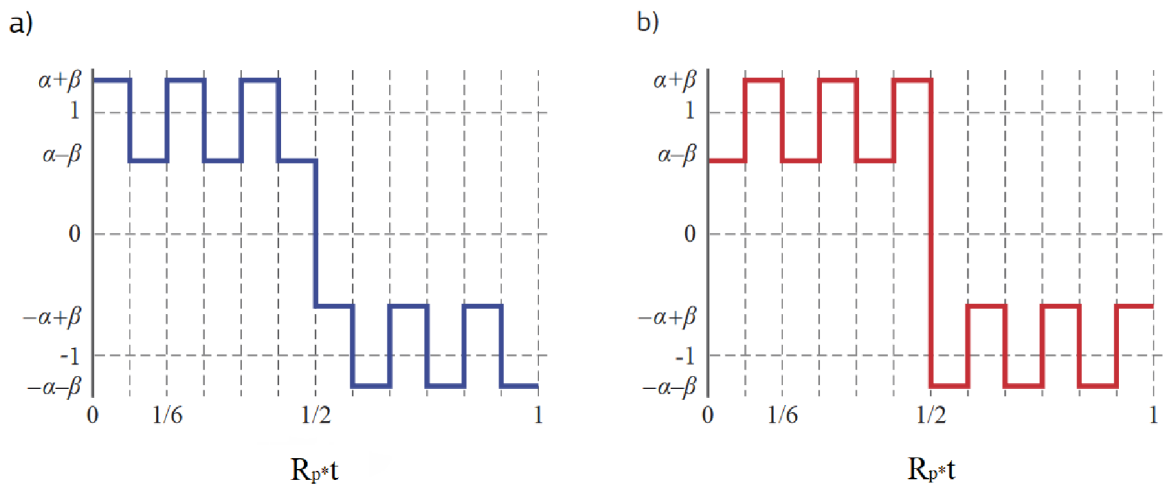
$$\beta = \sqrt{\frac{1}{11}} \quad (1.7)$$

$$s_{C_{BOC(1,1)}}(t) = \text{sgn}[\sin(2\pi R_p t)], \quad (1.8)$$

$$s_{C_{BOC(6,1)}}(t) = \text{sgn}[\sin(6 \cdot 2\pi R_p t)], \quad (1.9)$$

kde význam proměnných je následující:

- t je čas,
- $s_{C_{BOC(1,1)}}(t)$ a $s_{C_{BOC(6,1)}}(t)$ jsou časové průběhy subnosných typu BOC(1, 1) a BOC(6, 1),
- R_p je čipová rychlost primárních kódů, $R_p = 1023000 \text{ s}^{-1}$.



Obr. 1.5 Časové průběhy CBOC subnosných - a) datový signál, b) pilotní signál.

1.3 Ostatní navigační satelitní systémy

Mezi další navigační satelitní systémy s globálním pokrytím (GNSS) je možné zařadit systém Glonass, který je vlastněn a spravován Ruskou federací, nebo čínský systém BeiDou-3. [13]

Ze systémů s lokálním pokrytím, pro které již nelze použít označení GNSS, lze zmínit například čínský BeiDou-1 a BeiDou-2, japonský QZSS a indický IRNSS. [13]

2 Vývoj softwarového přijímače

Tato kapitola se zabývá návrhem a simulací jednotlivých funkčních bloků softwarového přijímače potřebných pro příjem navigačních signálů v pásmu L1/E1. Veškerý návrh a simulace byly provedeny v prostředí MATLAB. Vzorčky navigačních signálů použitých během návrhu přijímače pro otestování správnosti řešení byly získány pomocí softwarového rádia Airspy R2 a programu SDR#. Softwarové rádio bohužel umožňovalo volbu pouze mezi dvěma vzorkovacími frekvencemi, a sice 2,5 MHz a 10 MHz. Pro L1 C/A signál byla dostačující vzorkovací frekvence 2,5 MHz, kdy připadají více než dva vzorky na jeden chip C/A kódu. V případě E1-OS signálu by pro zaznamenání subnosné BOC(6, 1), která je nejrychlejší složkou signálu, byly potřeba alespoň dva vzorky na jednu periodu této subnosné, čemuž odpovídá vzorkovací frekvence o hodnotě $(2 \cdot 6 \cdot 1,023) \text{ MHz} = 12,276 \text{ MHz}$, a proto byla pro tento signál zvolena vyšší dostupná, i když ne zcela ideální, vzorkovací frekvence 10 MHz. Výstupem programu SDR# je komplexní signál v základním pásmu uložený v souboru ve formátu WAV nesoucím dva kanály určené pro složky I a Q.

2.1 Nalezení navigačního signálu

Při zahájení příjmu je nejdříve potřeba nalézt přibližnou hodnotu frekvenčního posuvu nosné vlivem Dopplerova jevu a aktuální fázi rozprostíracího kódu. Tento proces se v anglicky psané literatuře označuje jako *acquisition*. Počeštěný název používaný v této práci je akvizice. [2]

Akvizici družice je možné vnímat jako hledání korelačního maxima přijaté a lokální rozprostírací sekvence ve 2D prostoru, jehož osy jsou tvořeny frekvenčním posuvem nosné frekvence a fází rozprostírací sekvence. V případě, že se v tomto prostoru nachází maximum přesahující definovanou hranici, v signálu byla nalezena družice s fází dané rozprostírací sekvence a frekvenčním posuvem odpovídajícím souřadnicím tohoto maxima. Fází rozprostírací sekvence je potřeba rozmítat v celém rozsahu s krokem o maximální hodnotě 1 chip. Frekvenční rozmítání je vhodné provádět v rozsahu $\pm 10 \text{ kHz}$ s krokem maximálně 500 Hz, přičemž v mnoha případech, jak bude patrné v některých následujících kapitolách, je potřeba volit i menší hodnoty tohoto kroku. [2]

2.1.1 Akvizice s postupným hledáním

Základní metodou akvizice je postupný výpočet všech korelací odpovídajících všem možným kombinacím fáze rozprostíracího kódu a frekvenčního posuvu v rámci definovaného rozsahu a kroku. Jedná se sice o jednoduchou, ale za to výpočetně dosti neefektivní, metodu. Například pro vyzkoušení všech 1023 možných fází C/A kódu a frekvenčních posuvů ve výše uvedeném intervalu je potřeba vypočítat korelaci pro celkem $1023 \cdot \left(2 \cdot \frac{10 \text{ kHz}}{500 \text{ Hz}} + 1\right) = 41943$ kombinací. [2]

2.1.2 Akvizice s paralelním hledáním fáze kódu

Pro akvizici dále existují metody založené na FFT (*Fast Fourier Transform*) vedoucí k částečné paralelizaci. Jedna z těchto metod, která je použita i v této práci, umožňuje pro jednotlivé frekvenční posuvy paralelně prohledat všechny fáze rozprostíracího kódu. Například pro rozsah ± 10 kHz s krokem 500 Hz vychází pro akvizici pouze $\left(2 \cdot \frac{10 \text{ kHz}}{500 \text{ Hz}} + 1\right) = 41$ kombinací. Metoda je o něco složitější, ale výpočetně mnohem méně náročná než metoda popsaná v předchozí podkapitole. [2]

Pro vzájemnou cyklickou korelaci $r_{xy}(n)$ dvou diskrétních komplexních funkcí x, y délky N s periodickým opakováním a vzájemným posunutím n platí rovnice 2.1. [14]

$$r_{xy}(n) = \frac{1}{N} \sum_{m=0}^{N-1} x^*(m) y(m+n) \quad (2.1)$$

Korelaci v rovnici 2.1 je možné v případě cyklicity vyjádřit rovněž pomocí diskrétní konvoluce vztahem 2.2. [14]

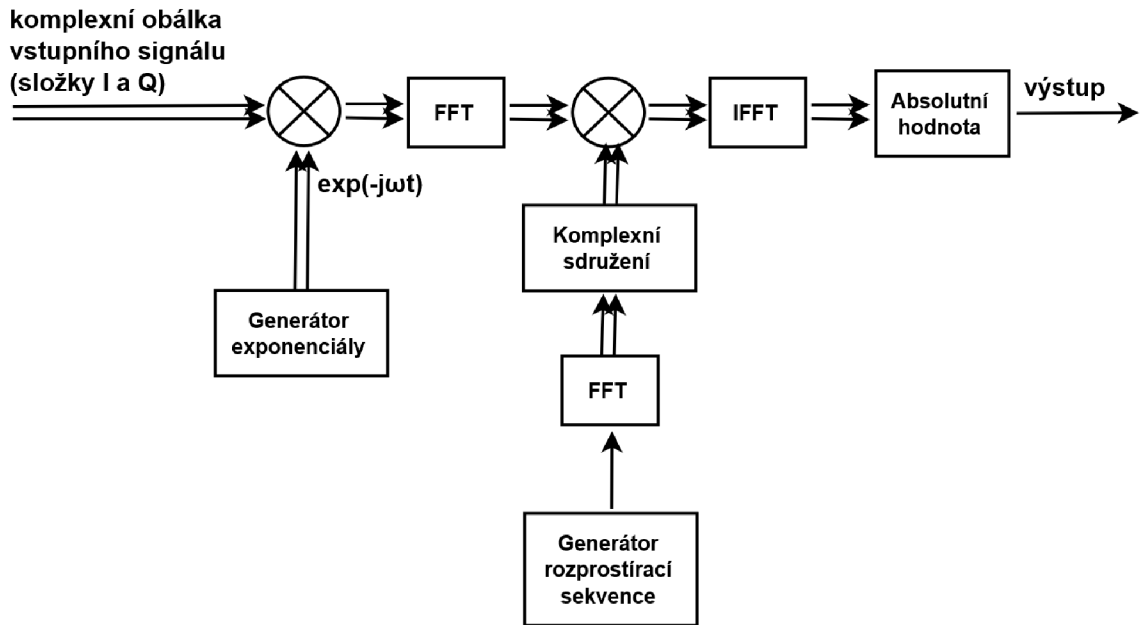
$$r_{xy}(n) = \frac{1}{N} (x^*(-m) * y(m))(n) = \frac{1}{N} \sum_{m=0}^{N-1} x^*(-m) y(-m+n) \quad (2.2)$$

Konečné výsledky obou vztahů 2.1 a 2.2 se liší pouze znaménkem u indexu m , což má však v případě cyklicity vliv pouze na pořadí sčítanců. Jelikož konvoluce dvou funkcí v časové oblasti představuje násobení těchto funkcí v oblasti frekvenční, lze korelaci $r_{xy}(n)$ ve frekvenční oblasti vyjádřit vztahem 2.3. [14]

$$F\{r_{xy}(n)\} = \frac{1}{N} \cdot F\{x^*(-m)\} \cdot F\{y(m)\} = \frac{1}{N} \cdot (F\{x(m)\})^* \cdot F\{y(m)\} \quad (2.3)$$

Na základě vztahu 2.3 lze již snadno získat požadovanou cyklickou korelaci pomocí inverzní Fourierovy transformace, jak je ukázáno ve vztahu 2.4 [14], ze kterého rovněž vychází i metoda akvizice popisovaná v této podkapitole, jejíž blokové schéma je na obrázku 2.1 [2].

$$r_{xy}(n) = \frac{1}{N} \cdot F^{-1}\{(F\{x(m)\})^* \cdot F\{y(m)\}\} \quad (2.4)$$



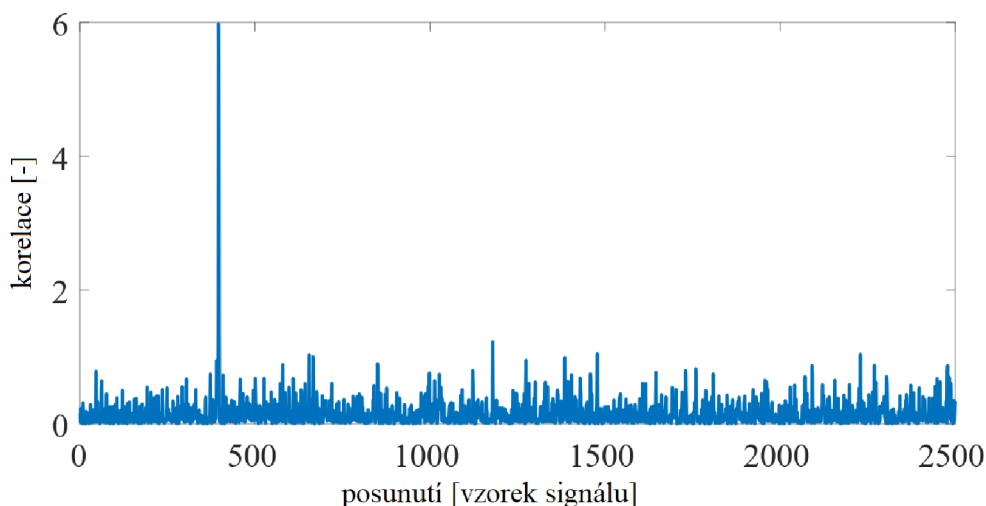
Obr. 2.1 Akvizice s paralelním hledáním fáze rozprostírací sekvence.

Komplexní obálka vstupního navigačního signálu je nejprve vynásobena exponenciálou kompenzující frekvenční posuv. Následně je pomocí rychlé Fourierovy transformace (FFT) vypočítána vzájemná cyklická korelace vstupního signálu a rozprostírací sekvence ve frekvenční oblasti. Graf vzájemné cyklické korelace v časové oblasti je nakonec získán rychlou inverzní Fourierovou transformací (IFFT) a výpočtem absolutní hodnoty. Do výpočtu musí vstupovat signál o délce odpovídající celočíselnému počtu period rozprostírací sekvence, přičemž větší počet period odpovídá výraznějšímu korelačnímu maximu. Postup popsany v tomto odstavci se opakuje pro všechny frekvenční posuvy v daném intervalu a s daným krokem. [2]

2.1.2.1 Akvizice signálu GPS

V případě L1 C/A signálu může být výsledek akvizice negativně ovlivněn výskytem bitových přechodů ve zkoumaném signálu. Z tohoto důvodu bylo potřeba provést akvizici pro několik na sebe navazujících úseků signálu o takové délce, aby byla jistota, že alespoň jeden z těchto úseků neobsahuje bitové přechody. Simulacemi v MATLABu se osvědčilo provádět akvizici pro dva navazující úseky signálu s délkou jednoho úseku 8 až 10 milisekund, což je 8 až 10 period C/A sekvence. Na jeden bit připadá 20 period C/A sekvence, což znamená, že minimálně jeden korelační graf nebude ovlivněn bitovým přechodem. Pro spolehlivé nalezení dostupných družic bylo provedeno rozmitání frekvenčního posuvu v rozsahu ± 10 kHz s krokem 125 Hz. Relativně nízká hodnota kroku rozmitání byla zvolena jednak za účelem nalezení i družic se slabým signálem, a jednak kvůli vlastnostem smyček frekvenční synchronizace, které jsou popsány v pozdějších kapitolách této práce. Krok rozmitání fáze rozprostíracího kódu je v případě popisované metody dán reciprokou hodnotou počtu vzorků signálu

připadajících na jeden čip, což je v případě získaného simulačního signálu GPS popsaného v úvodu kapitoly $\frac{1023000 \text{ chip} \cdot \text{s}^{-1}}{2500000 \text{ s}^{-1}} = 0,4092 \text{ chip}$. Příklad korelačního grafu s nalezenou družicí v daném frekvenčním posuvu pro L1 C/A signál je na obrázku 2.2.



Obr. 2.2 Akvizice L1 C/A signálu.

Na obrázku 2.2 je výrazné korelační maximum pro posunutí o 396 vzorků. Ve vztahu 2.1 je toto posunutí značeno indexem n , rozprostírací sekvence funkcí $x(m)$ a vstupní signál funkcí $y(m)$. Znamená to tedy, že ve zkoumaném úseku signálu se na pozici odpovídající n -tému vzorku nachází počátek rozprostírací sekvence.

2.1.2.2 Akvizice signálu Galileo

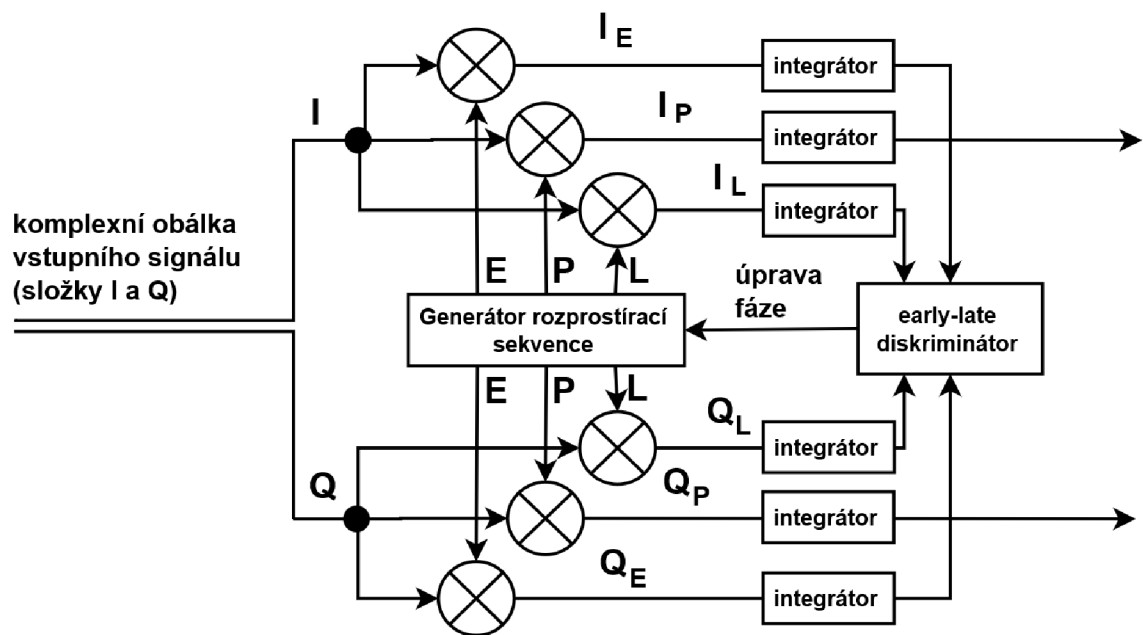
Výhodou E1-OS signálu systému Galileo oproti signálu L1 C/A je absence bitových přechodů v pilotním signálu, na kterém je akvizice prováděna. Nevýhodou naopak je dlouhá doba trvání jedné periody sekundárního kódu, která činí 100 milisekund, a vyšší vzorkovací frekvence, což vede k mnohem většímu počtu vzorků potřebnému k provedení akvizice.

Akvizice byla prováděna nad jednou periodou sekundárního kódu s frekvenčním rozmitáním shodným s akvizicí L1 C/A signálu. Použitý přístup akvizice je relativně jednoduchý a pro účely této práce dostačující, avšak ne zcela optimální. Pro účinnou akvizici by bylo vhodné použití pokročilejších metod, jejichž studium však není předmětem této práce. Na rozdíl od L1 C/A signálu bylo navíc potřeba během akvizice odstranit subnosnou CBOC, jejíž časový průběh je na obrázku 1.5, a která je synchronní s primárním kódem, čehož bylo dosaženo vynásobením rozprostíracího kódu touto subnosnou před operací FFT. Vzhledem k nedostatečné vzorkovací frekvenci signálu použitého pro simulaci, musela být složka BOC(6, 1) zanedbána. Vlastnosti korelačního grafu signálu E1-OS jsou srovnatelné s grafem na obrázku 2.2.

2.2 Sledování rozprostíracího kódu

Vlivem Dopplerova posuvu a prakticky nedosažitelné shodnosti taktovacích frekvencí generátorů rozprostíracích kódů na přijímací a vysílací straně není chipová rychlost přijaté a lokálně generované rozprostírací sekvence shodná, a proto je potřeba během příjmu dohlížet na jejich synchronnost. [2]

Ke sledování fáze rozprostírací sekvence byla použita DLL (*Delay Lock Loop*) smyčka s tak zvaným early-late diskriminátorem. Blokové schéma DLL smyčky je na obrázku 2.3. [2]



Obr. 2.3 Blokové schéma DLL smyčky.

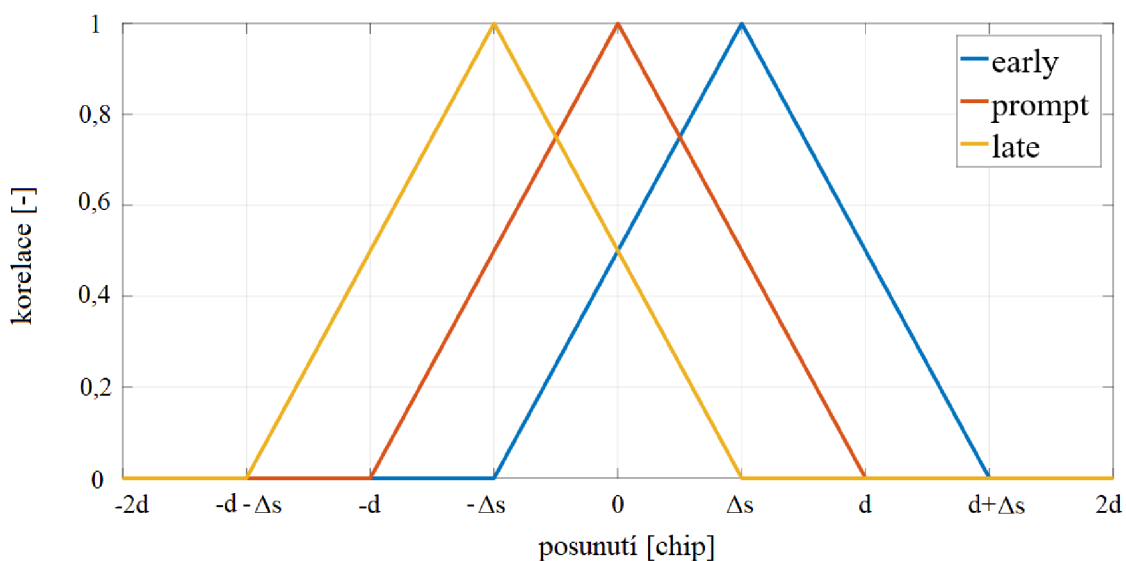
DLL smyčka na obrázku 2.3 generuje celkem 3 verze rozprostírací sekvence [2]:

- P (prompt), sekvence, jejíž fáze je udržovaná synchronní s fází přijímané sekvence,
- E (early), sekvence používaná pro úpravu fáze sekvence P, vzhledem k sekvenci P má fázi posunutou o definovanou část chipu Δs v kladném směru,
- L (late), sekvence používaná pro úpravu fáze sekvence P, vzhledem k sekvenci P má fázi posunutou o definovanou část chipu Δs v záporném směru.

Násobičky následované integrátory tvoří společně korelátoři pro jednotlivé kombinace složek vstupního signálu a verzí rozprostírací sekvence. Diskriminátor vyhodnocuje rozdíl míry korelace pro sekvence E a L, na základě kterého odhaduje fázový rozdíl přijímané sekvence a lokální P sekvence, která je použita pro

derozprostírání datových bitů. Odhadnutý fázový rozdíl vstupuje do generátoru rozprostíracích sekvencí a slouží pro úpravu fáze sekvencí E, P a L. [2]

Princip fungování diskriminátoru vychází ze zjednodušeného grafu vzájemné korelace rozprostíracích sekvencí, podle kterého má korelace v závislosti na vzájemném posunutí sekvencí lineární charakter pro rozdíl fází v intervalu $\pm d$ chipů, mimo tento interval má nulovou hodnotu. Graf korelací pro sekvence E, P a L, které jsou vzájemně posunuty o hodnotu Δs , je na obrázku 2.4, přičemž na vodorovné ose je vyneseno rozdíl fáze přijímané sekvence a sekvence P. [2]



Obr. 2.4 Grafy korelací rozprostíracích sekvencí.

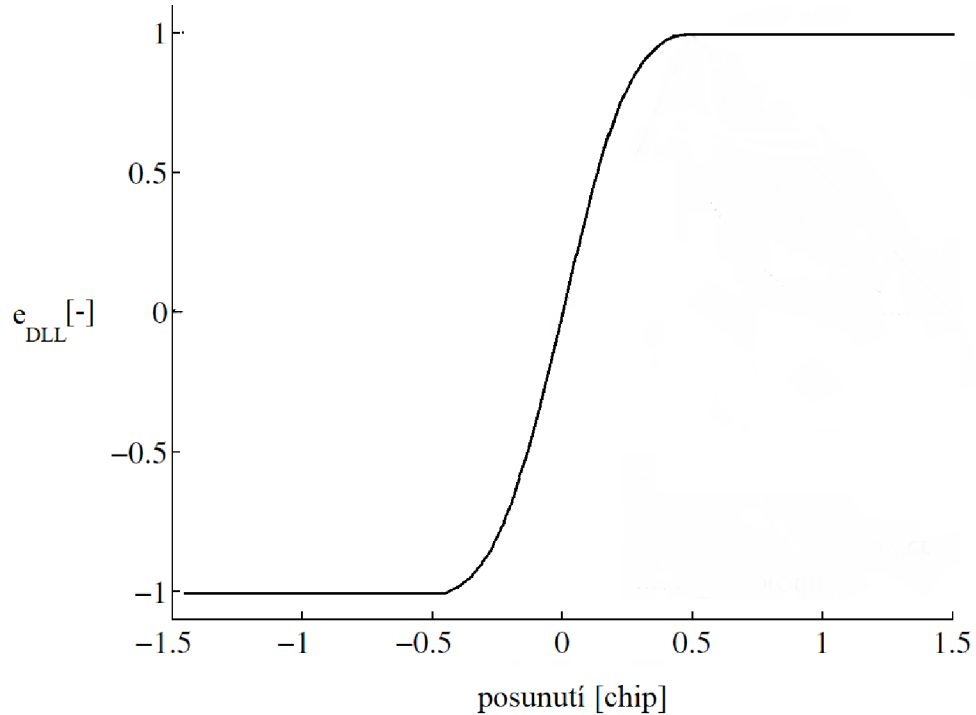
Z obrázku 2.4 plyne, že rozdíl míry korelace mezi E a L sekvencemi je přímo úměrný rozdílu fáze přijímané sekvence a sekvence P. Hodnota parametru d byla zjištěna simulací v MATLABu, přičemž pro signál L1 C/A platí přibližně $d = 1 \text{ chip}$ a pro signál E1-OS platí $d = 0,3 \text{ chipu}$. Pro hodnotu vzájemného posunutí sekvencí E, P a L se simulacemi osvědčilo zvolit přibližně poloviční hodnotu vůči hodnotě d . Konkrétně bylo zvoleno $\Delta s = 0,1875$ pro E1-OS a $\Delta s = 0,5$ pro L1 C/A.

Pro odhad rozdílu fáze se v diskriminátoru vychází z chybového signálu e_{DLL} , pro jehož vyjádření byl zvolen normalizovaný rozdíl výkonů na výstupech korelatorů E a L podle vztahu 2.5 [2]. Graf závislosti chybového signálu na rozdílu fáze přijímané a lokální sekvence je pro signál L1 C/A na obrázku 2.5 [2]. Mezi výhody zvoleného řešení patří možnost práce s různě silnými signály díky normování a funkčnost diskriminátoru i v případě nesynchronní lokální nosné frekvence díky zahrnutí obou složek I a Q. Naopak drobnou nevýhodou je ne zcela lineární průběh chybového signálu v intervalu $\pm \Delta s$ chipu, jak je možné vidět na obrázku 2.5. Linearizace by bylo možné dosáhnout odmocněním výrazů vyjadřujících ve vztahu 2.5 výkony ve větvích E a L, což však není provedeno kvůli vysoké výpočetní náročnosti odmocniny.

$$e_{DLL} = \frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}, \quad (2.5)$$

kde význam proměnných je následující:

- e_{DLL} je chybový signál,
- I_E, I_L, Q_E, Q_L jsou výkony na výstupu korelátorů podle obrázku 2.3.



Obr. 2.5 Závislost chybového signálu na rozdílu fáze přijímané a lokální sekvence pro signál L1 C/A.

Odhad fázového posunutí v chipech byl z chybového signálu získán vynásobením konstantou úměrnosti K_{DLL} . Hodnota této konstanty odpovídá reciproké hodnotě derivace chybového signálu ze vztahu 2.5 v bodě odpovídajícímu nulovému posunutí.

Pro výpočet derivace byla ve vztahu 2.5 nejprve provedena substituce výkonů vyjádřených pomocí I a Q složek podle vztahu 2.6.

$$e_{DLL} = \frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)} = \frac{E^2 - L^2}{E^2 + L^2}, \quad (2.6)$$

kde proměnné E, L odpovídají korelačním křivkám early, late na obrázku 2.4. Tyto korelační křivky byly vyjádřeny složením tří lineárních funkcí podle vztahů 2.7 a 2.8.

$$E = \begin{cases} \delta + (1 - \mu), & \delta \in \langle -0,5; 0,5 \rangle \\ 0, & \delta \in (-\infty; -0,5) \cup (1,5; \infty) , \\ -\delta + (1 + \mu), & \delta \in \langle 0,5; 1,5 \rangle \end{cases} \quad (2.7)$$

$$L = \begin{cases} -\delta + (1 - \mu), & \delta \in \langle -0,5; 0,5 \rangle \\ 0, & \delta \in (-\infty; -1,5) \cup (0,5; \infty) , \\ \delta + (1 + \mu), & \delta \in \langle -1,5; -0,5 \rangle \end{cases} \quad (2.8)$$

kde δ vyjadřuje posunutí v chipch a μ je odstup mezi sekvencemi. Pro výpočet derivace vztahu 2.6 pro nulové posunutí byly do tohoto vztahu dosazeny vztahy 2.7 a 2.8 pro $\delta = 0$, výsledek po úpravě ukazuje vztah 2.9.

$$e_{DLL} = \frac{2\delta - 2\delta\mu}{\delta^2 + \mu^2 - 2\mu + 1} \quad (2.9)$$

Derivace vztahu 2.9 je po úpravě dána vztahem 2.10.

$$\frac{de_{DLL}}{d\delta} = -\frac{2\delta^2 - 6\mu^2 + 6\mu - 2\delta^2\mu + 2\mu^3 - 2}{(\delta^2 + \mu^2 - 2\mu + 1)^2} \quad (2.10)$$

Dosazením $\delta = 0$ a $\mu = \frac{1}{2}$ pro L1 C/A a $\mu = 0,1875$ pro E1-OS do vztahu 2.10 byla určena hodnota derivace, neboli směrnice tečny, křivky na obrázku 2.5 pro bod odpovídající nulovému posunutí, platí tedy $e'_{DLL}(0) = 4$ pro L1 C/A a $e'_{DLL}(0) = \frac{32}{13}$ pro E1-OS, z čehož pro hodnotu konstanty úměrnosti popsané na předchozí straně plyne $K_{DLL} = \frac{1}{e'_{DLL}(0)} = \frac{1}{4}$ pro L1 C/A a $K_{DLL} = \frac{1}{e'_{DLL}(0)} = \frac{13}{32}$ pro E1-OS.

Rozdíl fáze $\Delta\varphi_{DLL}$, která je generátorem rozprostíracích sekvencí přičítána k jejich aktuální fázi, je dána vztahem 2.11. Jednotkou rozdílu fáze $\Delta\varphi_{DLL}$ je chip.

$$\Delta\varphi_{DLL} = K_{DLL} \cdot e_{DLL} \quad (2.11)$$

Popsaná sledovací smyčka byla implementovaná v MATLABu a její funkčnost ověřena pro oba zkoumané navigační signály. Sledovací smyčka bez problému sledovala fázi rozprostíracího kódu pro všechny akvizici nalezené družice s výjimkou několika družic se slabým signálem, kde docházelo místy ke ztrátě synchronizace. Tento problém byl vyřešen experimentálním zmenšením hodnoty konstanty K_{DLL} na jednu desetinu pro GPS a na jednu čtvrtinu pro Galileo, tedy $K_{DLL,exp,GPS} = \frac{1}{40}$ a $K_{DLL,exp,Galileo} = \frac{13}{128}$. Verze sledovacích smyček s upravenými konstantami $K_{DLL,exp}$ již byly schopny správně sledovat rozprostírací kódy i u družic se slabým signálem, aniž by byla nějak negativně ovlivněna jejich sledovací schopnost zbývajících družic. Doba

integrace E a L sekvencí byla zvolena tak, aby odpovídala periodě C/A sekvence u GPS a periodě primární sekvence u Galilea.

2.3 Sledování nosného kmitočtu

V této podkapitole jsou rozebrány metody frekvenční a fázové synchronizace nosného kmitočtu. Frekvenční synchronizace je potřebná k nalezení přesného frekvenčního posuvu nosné frekvence, který je akvizicí odhadnut pouze přibližně. Fázová synchronizace má za úkol jednak sledovat fázi nosné frekvence za účelem koherentní demodulace, a jednak kompenzovat pomalé změny nosné frekvence způsobené změnou rychlosti družice vzhledem k přijímači.

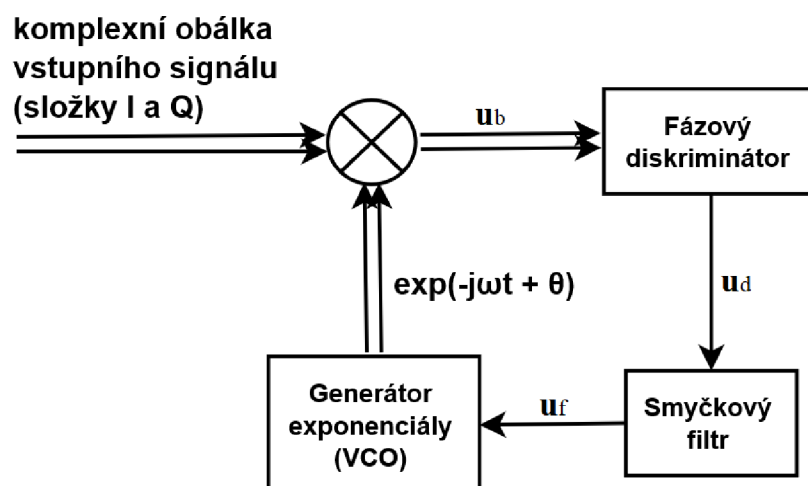
Vzhledem k tomu, že většina níže popisovaných metod frekvenční synchronizace vychází z Costasovy smyčky, která je primárně určena pro synchronizaci fáze, jsou metody fázové synchronizace popsány jako první navzdory obrácenému pořadí synchronizací v přijímacím řetězci.

2.3.1 Fázová synchronizace

V rámci sledování fáze nosného kmitočtu byly vytvořeny dvě možná řešení popsaná v následujících podkapitolách.

2.3.1.1 Costasova smyčka – popis

Costasova smyčka je díky odolnosti vůči symbolovým přechodům na nosné frekvenci a použitelnosti pro různé modulační metody jedna z nejpoužívanějších synchronizačních smyček. V této práci byla použita Costasova smyčka pro BPSK modulaci ve verzi určené pro komplexní vstupní signál, jejíž blokové schéma je na obrázku 2.6 [16].



Obr. 2.6 Blokové schéma Costasovy smyčky pro komplexní BPSK signál.

Costasova smyčka na obrázku 2.6 se snaží ladit frekvenci exponenciály na výstupu jejího generátoru tak, aby imaginární složka signálu u_b byla, pokud možno, nulová. Jinými slovy se snaží udržovat exponenciálu ve fázi, případně protifázi, se vstupním signálem. Z reálné složky signálu u_b je v takovém případě možné dekódovat příchozí symboly. [16]

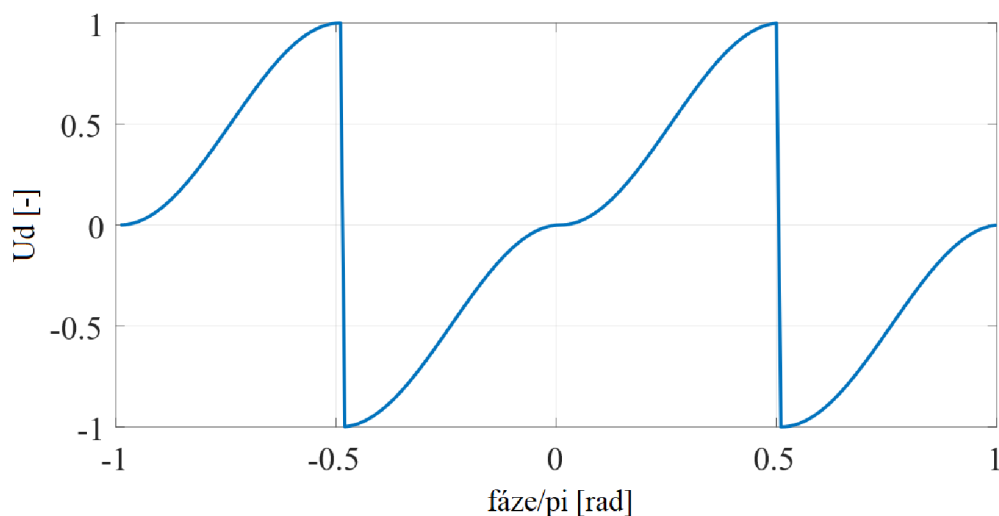
Fázový diskriminátor má za úkol generovat chybový signál u_d úměrný fázi signálu u_b , avšak s ohledem na možný výskyt symbolových přechodů, což je možné slovně popsat jako převod fází mimo interval $(-90^\circ; +90^\circ)$ do tohoto intervalu přičtením nebo odečtením 180° , například na fázi o hodnotě 110° musí být nahlíženo jako na fázi o hodnotě -70° [16]. V odborné literatuře byla nalezena řešení fázových diskriminátorů, která buď umožňovala práci pouze s normalizovaným signálem, nebo byla výpočetně příliš náročná (například využití arctg), a proto byl vytvořen diskriminátor odhadující fázi na základě normovaného výkonu imaginární složky signálu u_b . Použitý fázový diskriminátor je vyjádřen rovnicí 2.12.

$$u_d = K_d \cdot \operatorname{sgn}(I) \cdot \left[\frac{\operatorname{sgn}(Q) \cdot Q^2}{I^2 + Q^2} \right], \quad (2.12)$$

kde význam proměnných je následující:

- I a Q jsou označení pro reálnou a imaginární část signálu u_b
- K_d je zisk fázového diskriminátoru.

Polarita proměnné Q ve vztahu 2.12 určuje polaritu fáze, kterou je třeba kompenzovat a polarita proměnné I zohledňuje symbolové přechody. Na obrázku 2.7 je graf závislosti chybového signálu u_d na fázi vstupního signálu pro $K_d = 1$. V grafu lze vidět, že diskriminátor skutečně generuje shodný chybový signál pro fáze vzájemně vzdálené o π radiánů (180°).



Obr. 2.7 Chybový signál fázového diskriminátoru.

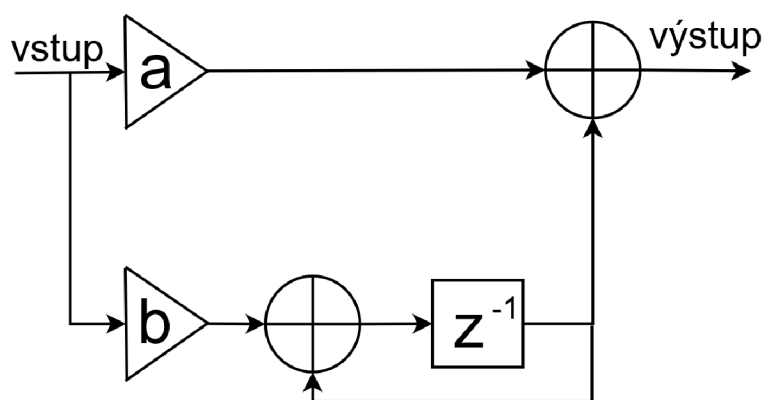
Chybový signál u_d je filtrován smyčkovým filtrem na signál u_f , který nastavuje frekvenci generované exponenciály podle vztahu 2.13. [16]

$$\omega = 2\pi K_0 u_f, \quad (2.13)$$

kde význam proměnných je následující:

- K_0 představuje zisk generátoru exponenciály, jednotkou je radián/sekunda,
- u_f je chybový signál filtrovaný smyčkovým filtrem,
- ω je úhlový kmitočet generované exponenciály.

Smyčkový filtr má charakter dolní propusti. Blokové schéma smyčkového filtru je na obrázku 2.8 [15].



Obr. 2.8 Smyčkový filtr.

Jak lze vidět na obrázku 2.8, smyčkový filtr se skládá z 2 paralelních větví, z nichž jedna (s koeficientem a) reaguje na okamžité změny vstupního chybového signálu, a druhá (s koeficientem b) kumuluje vstupní chybový signál pomocí integrátoru a udržuje tak na výstupu i při nulovém vstupu hodnotu nastavující frekvenci exponenciály. Míra vlivu obou větví se ladí pomocí koeficientů a , b .

2.3.1.2 Costasova smyčka – návrh

V této podkapitole je popsán obecný postup, jakým byly použité Costasovy smyčky navrženy. Z informací uvedených v předchozí podkapitole plyne, že návrh Costasovy smyčky spočívá ve stanovení 4 parametrů - K_a , K_0 , a , b .

Na začátku návrhu byly zvoleny hodnoty tak zvaného činitele tlumení ξ [-] a maximálního frekvenčního rozdílu pro rychlé zachycení f_L [Hz]. Činitel tlumení určuje míru překmitů frekvence exponenciály na počátku synchronizace. S rostoucí hodnotou ξ klesá míra překmitů, ale také roste doba potřebná pro fázovou synchronizaci. Doporučovanou hodnotou činitele tlumení je $\xi = 0,7$ [2]. Parametr f_L definuje maximální frekvenční rozdíl (v anglické literatuře označováno termínem *lock range frequency*), pro který dojde k rychlé fázové synchronizaci smyčky, čímž se rozumí doba

v řádu nižších jednotek počtu period frekvenčního rozdílu. Překročí-li frekvenční rozdíl hodnotu f_L , synchronizace smyčky trvá mnohem delší dobu. V případě Costasovy smyčky pro reálné vstupní signály je navíc definován rozsah frekvenčního rozdílu (v anglické literatuře označován termínem *pull-in range frequency*), při jehož překročení není smyčka schopna provést synchronizaci, což je způsobeno fázovým zpožděním, které do systému vnáší filtry pro odstranění vyšších frekvenčních složek po převedení pásmového signálu do základního pásma. Hodnotu f_L lze v úhlové míře (jednotkou je rad/sekunda) vyjádřit vztahem 2.14. [16]

$$\omega_L = 2\pi f_L \quad (2.14)$$

Po zvolení hodnot ξ a f_L byla určena tak zvaná přirozená úhlová frekvence smyčky ω_n [rad/s] a šumová šířka pásma B_L [Hz] podle vztahů 2.15 a 2.16. [16][2]

$$\omega_n = \frac{\omega_L}{2 \cdot A_{FD} \cdot \xi}, \quad (2.15)$$

kde A_{FD} [-] je amplituda fázového diskriminátoru pro $K_d = 1$. Pro fázový diskriminátor definovaný vztahem 2.12 platí $A_{FD} = 1$.

$$B_L = \frac{\omega_n \cdot (4\xi^2 + 1)}{8\xi} \quad (2.16)$$

Šumová šířka pásma určuje přibližnou šířku pásma šumu přítomného ve vstupním signálu, který se nepodaří odstranit smyčkovým filtrem. Čím menší je šumová šířka pásma, tím je smyčka odolnější vůči nízkému poměru signál-šum a tím je přesnější i její fázová synchronizace. Zároveň však klesá maximální frekvenční rozdíl f_L , což je jeden z důvodů, proč nejsou smyčky fázové synchronizace vhodné pro synchronizaci signálů s velkým frekvenčním posuvem. [2]

Pro výpočet požadovaných parametrů podle následujících vztahů bylo potřeba zvolit hodnotu jednoho z nich, a sice hodnotu zisku fázového diskriminátoru, pro který bylo zvoleno $K_d = 1$. Dále bylo potřeba si určit hodnotu první časové konstanty smyčkového filtru, pro kterou byla zvolena doporučená hodnota $\tau_1 = 200 \text{ ms}$. [16]

Zisk frekvence generátoru exponenciály K_0 byl určen podle vztahu 2.17. [16]

$$K_0 = \frac{\omega_n^2 \cdot \tau_1}{K_d} \quad (2.17)$$

Pro určení zbývajících dvou parametrů smyčkového filtru, a , b , byla nejprve vypočítána druhá časová konstanta smyčkového filtru τ_2 podle vztahu 2.18. [16]

$$\tau_2 = \frac{\omega_L \tau_1}{A_{FD} K_0 K_d} \quad (2.18)$$

Koeficienty smyčkového filtru byly vypočítány pomocí vztahů 2.19 a 2.20 za použití vzorkovací periody T_s , která odpovídá periodě vzorků vstupujících do smyčky. [16]

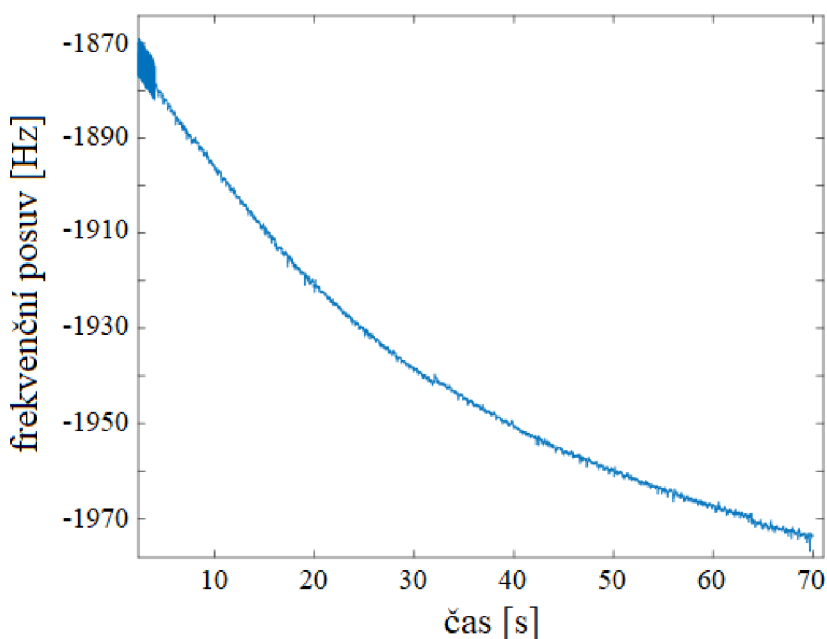
$$a = \frac{\tau_2}{\tau_1} \quad (2.19)$$

$$b = \frac{T_s}{\tau_1} \quad (2.20)$$

2.3.1.3 Costasova smyčka - simulace

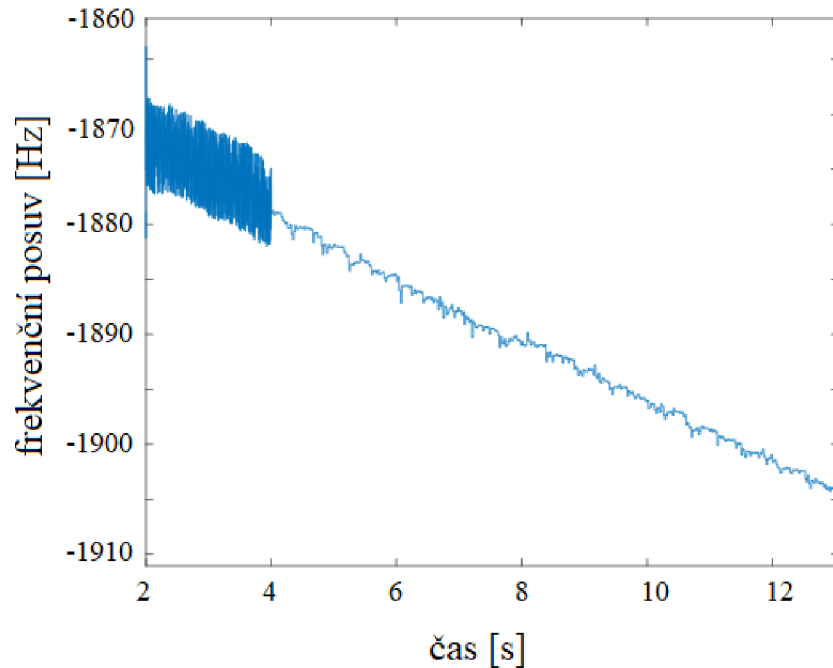
Jelikož vztahy použité pro návrh Costasovy smyčky vychází ze zjednodušených modelů, výsledný návrh nemusí zcela odpovídat vstupním požadavkům. Sledovací schopnosti navržených smyček byly proto ověřeny simulacemi v MATLABu pro oba zkoumané navigační signály.

V případě L1 C/A signálu systému GPS byly navrženy dvě Costasovy smyčky s maximálním frekvenčním rozdílem $f_L = 4 \text{ Hz}$. Jedna smyčka byla navržena pro sledování v okamžiku, kdy ještě není známa poloha bitových přechodů se vzorkovací periodou odpovídající periodě C/A sekvence, $T_s = 1 \text{ ms}$. Druhá smyčka byla navržena pro sledování signálu s již provedenou bitovou synchronizací pro vzorkovací periodu odpovídající době trvání jednoho bitu, $T_s = 20 \text{ ms}$. Výše zmíněna hodnota f_L byla zvolena s ohledem na, pokud možno, co nejmenší šířku pásma pronikajícího šumu a zároveň, aby smyčka měla dostatečnou sledovací schopnost pro navigační signál s neustále se měnícím nosným kmitočtem. Sledovací schopnosti smyčky byly zkoumány pomocí detektoru fázové synchronizace, který je popsán v jedné z následujících kapitol. Sledování nosného kmitočtu signálu L1 C/A je na obrázku 2.9.



Obr. 2.9 Frekvenční posuv sledovaný Costasovou smyčkou, L1 C/A.

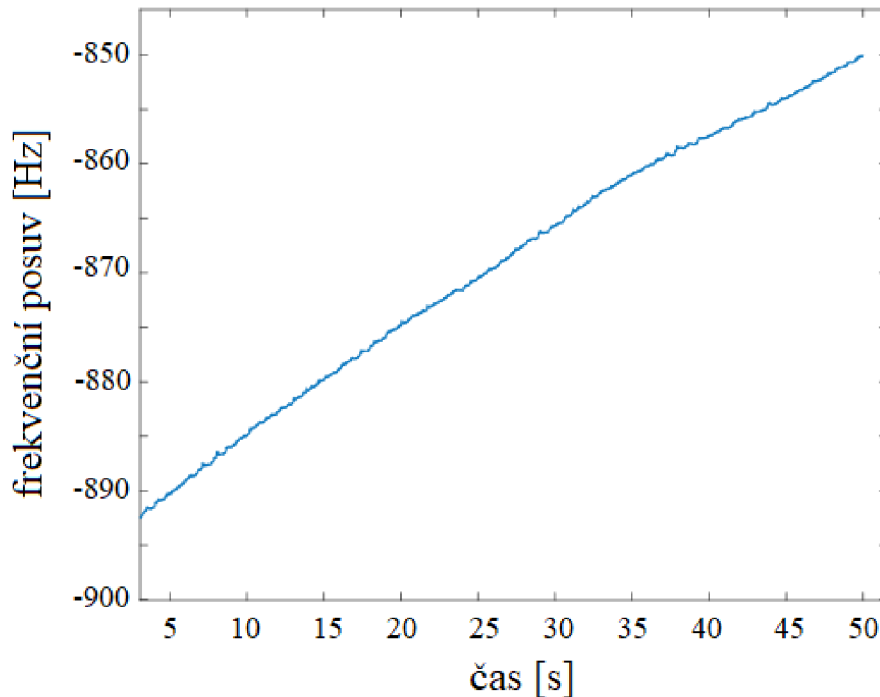
Na obrázku 2.9 je na počátku sledování znát o něco větší míra šumu v odhadované frekvenci, což je způsobeno kratší vzorkovací periodou v době před bitovou synchronizací. Detail sledování frekvence při přechodu na větší vzorkovací periodu po bitové synchronizaci je na obrázku 2.10.



Obr. 2.10 Frekvenční posuv sledovaný Costasovou smyčkou, L1 C/A, detail.

Na obrázku 2.10 lze vidět, že maximální výchylky frekvence od pravděpodobné hodnoty skutečného frekvenčního posuvu v zašuměné části grafu skutečně přibližně odpovídají nastavené hodnotě f_L . Pro názornost byla pro obrázky 2.9 a 2.10 záměrně vybrána družice se slabým signálem, protože pouze u těchto signálů je dostatečně nízký poměr signál-šum, který způsobí rozkmit v plném rozsahu.

Costasova smyčka pro E1-OS signál systému Galileo byla ze stejných důvodů navržena rovněž s maximálním frekvenčním rozdílem $f_L = 4 \text{ Hz}$ a vzorkovací periodou $T_s = 20 \text{ ms}$, což je experimentálně určená hodnota, při které bylo simulacemi v MATLABu dosaženo nejlepších sledovacích schopností. Vzhledem k tomu, že sledování bylo provedeno na pilotním signálu se známou sekvencí sekundárního kódu, na rozdíl od GPS byla navržena pouze jedna verze smyčky. Sledování nosného kmitočtu E1-OS signálu je na obrázku 2.11, v tomto případě pro družici s kvalitním signálem.



Obr. 2.11 Frekvenční posuv sledovaný Costasovou smyčkou, E1-OS.

2.3.1.4 Smyčka fázového závěsu

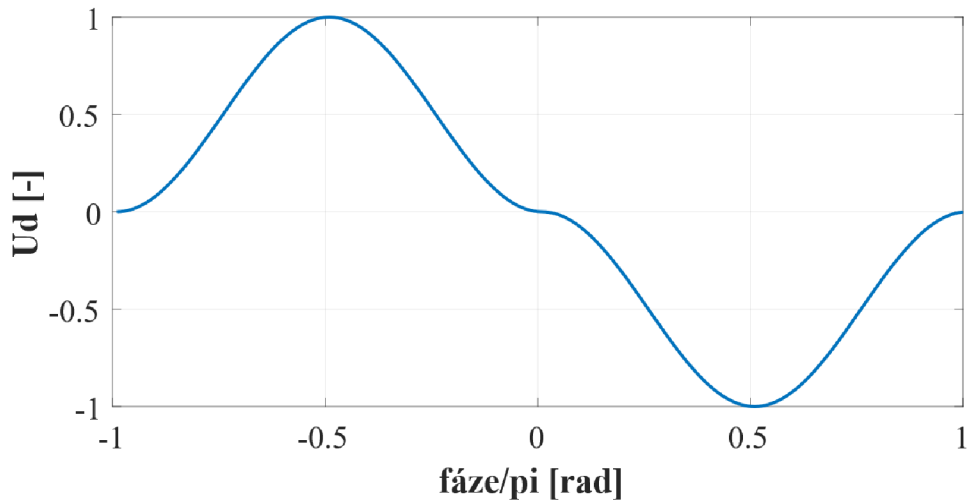
Smyčka fázového závěsu, též označována zkratkou PLL (*Phase Locked Loop*), představuje jednu z dalších metod pro fázovou synchronizaci. Na rozdíl od Costasovy smyčky však není odolná vůči symbolovým přechodům. [17]

V rámci této práce byla použita PLL smyčka, jejíž struktura je shodná se strukturou Costasovy smyčky na obrázku 2.6 s výjimkou fázového diskriminátoru, který bylo potřeba, stejně jako u Costasovy smyčky, navrhnout tak, aby byl schopný pracovat s různě silnými signály a zároveň byl výpočetně efektivní. Vytvořený fázový diskriminátor odhaduje fázi vstupního signálu na základě normalizovaného výkonu v imaginární složce signálu a je popsán vztahem 2.21, přičemž význam proměnných je shodný s významy ve vztahu 2.12.

$$u_d = K_d \cdot \left[\frac{\text{sgn}(Q) \cdot Q^2}{I^2 + Q^2} \right] \quad (2.21)$$

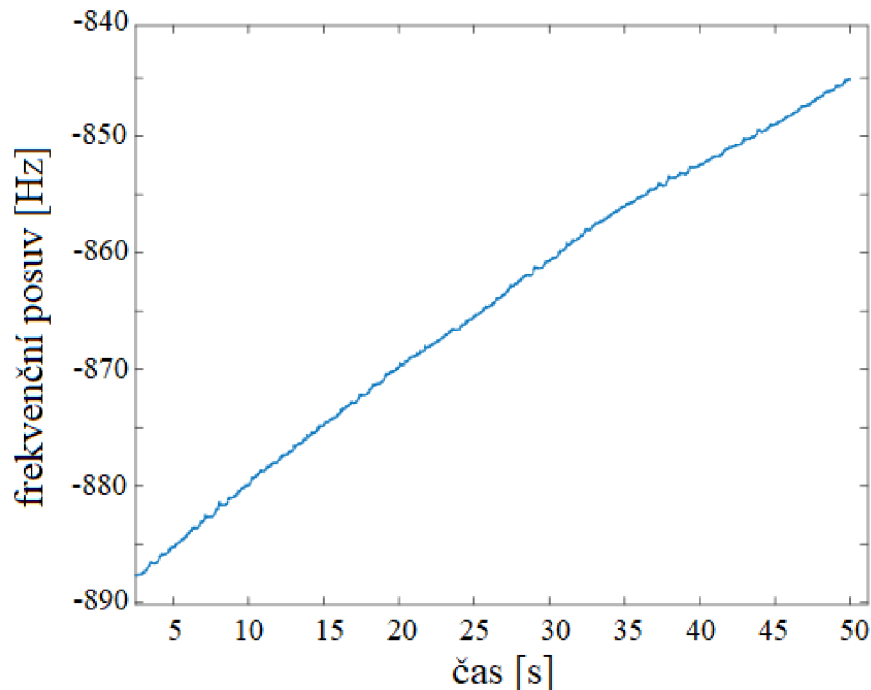
Vztah 2.21 je shodný se vztahem 2.12 až na zohlednění polaritě reálné složky signálu, která právě souvisí s bitovými přechody. Na obrázku 2.12 je graf závislosti chybového signálu u_d na fázi vstupního signálu pro $K_d = 1$, přičemž lze z grafu vyčíst, že i v případě použité PLL smyčky může být vstupní signál teoreticky i v protifázi s lokálně generovanou exponenciálou, avšak taková situace je jednak prakticky

vyloučena díky přítomnému šumu a měnícímu se frekvenčnímu posuvu, a jednak nemá zásadní vliv na dekódování dat v případě zohlednění fázové nejistoty.



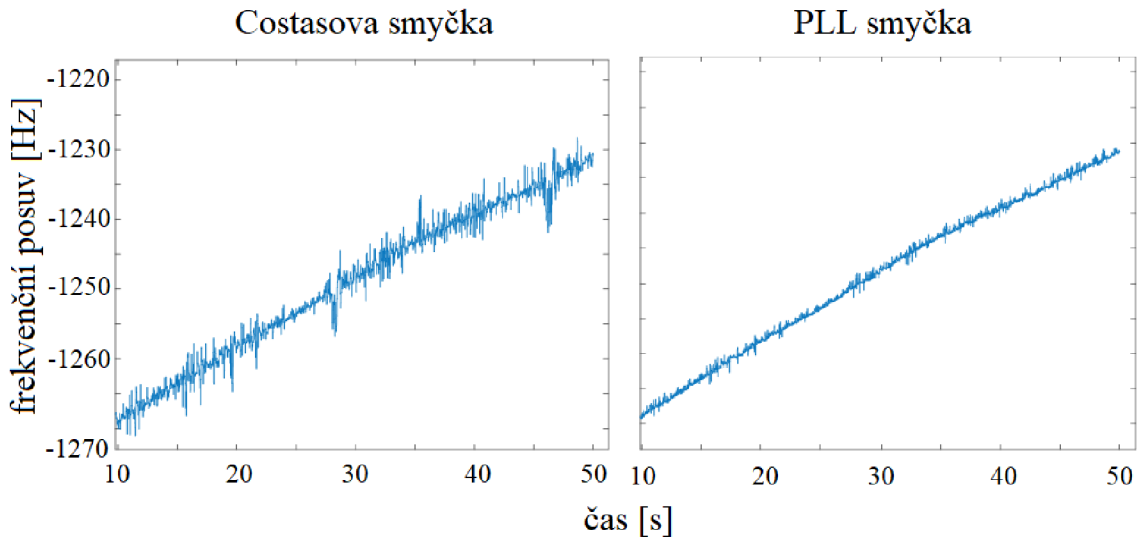
Obr. 2.12 Chybový signál fázového diskriminátoru PLL.

Jelikož PLL smyčka není odolná vůči bitovým přechodům přítomným v L1 C/A signálu, byla správnost popsaného řešení ověřena simulacemi pouze pro sledování fáze pilotního signálu E1-OS se známou fází sekundárního kódu. Smyčka byla navržena pomocí stejných vztahů jako v případě Costasovy smyčky. Postačující hodnota maximálního frekvenčního rozdílu byla nastavena $f_L = 1,5 \text{ Hz}$, což je méně než poloviční hodnota ve srovnání s Costasovou smyčkou. Sledování nosného kmitočtu E1-OS pro družici s kvalitním signálem je na obrázku 2.13.



Obr. 2.13 Frekvenční posuv sledovaný PLL smyčkou, E1-OS.

Porovnání sledovacích schopností Costasovy smyčky a PLL smyčky pro družici s kvalitou signálu odpovídající hranici dekódovatelnosti dat je na obrázku 2.14, na kterém lze vidět, že PLL smyčka má díky menší potřebné hodnotě f_L , respektive menší šumové šířce pásma, lepší odolnost vůči nízkému poměru signál-šum.



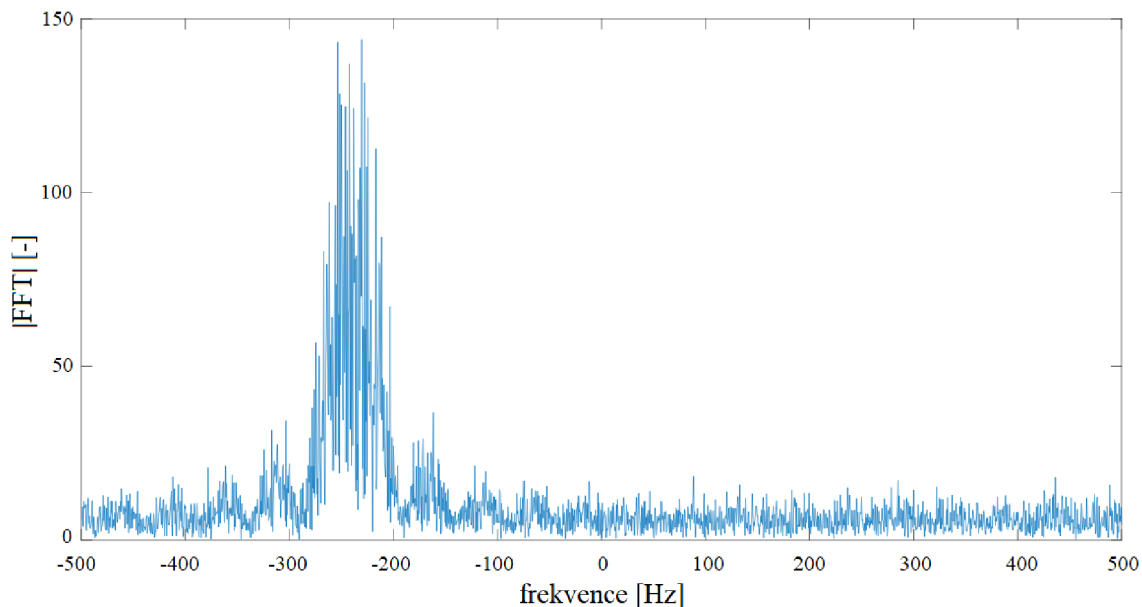
Obr. 2.14 Porovnání Costasovy smyčky a PLL smyčky.

2.3.2 Frekvenční synchronizace

Navigační signály na výstupu akvizice mohou být zatíženy značným frekvenčním posuvem, který nejsou smyčky fázové synchronizace schopny kompenzovat jednak kvůli potřebné velké šumové šířce pásma, a jednak kvůli riziku nestability pro velké hodnoty f_L . Z tohoto důvodu bylo nalezeno několik metod frekvenční synchronizace, které jsou popsány v následujících podkapitolách.

2.3.2.1 Synchronizace pomocí FFT

První myšlenkou, jak přesněji odhadnout skutečnou nosnou frekvenci, byla prostá frekvenční analýza dostatečně dlouhého navigačního signálu. Pro frekvenční analýzu byly použity vzorky signálu po částečné frekvenční kompenzaci na základě informace získané pomocí akvizice a po derozprostření požadovanou rozprostírací sekvencí. Simulacemi v MATLABu se osvědčilo využít ze získaného frekvenčního spektra přibližně 20 nejsilnější frekvenčních složek, z nichž se frekvenční posuv odhadnul jako aritmetický průměr frekvencí, které těmto složkám náležely. Předání takto získaného frekvenčního posuvu smyčkám fázové synchronizace vedlo k úspěšnému sledování fáze nosné frekvence. Příklad frekvenčního spektra navigačního signálu L1 C/A s frekvenčním posuvem přibližně -240 Hz je na obrázku 2.15.



Obr. 2.15 Frekvenční spektrum L1 C/A signálu s frekvenčním posuvem.

Analýza byla prováděna pomocí FFT nad 2048 vzorky pro L1 C/A signál a nad 512 vzorky pro E1-OS signál, čímž bylo pro oba signály dosaženo vzájemné vzdálenosti sousedních složek ve spektru méně než $0,5 \text{ Hz}$ pro vzorkovací periodu signálu odpovídající periodě C/A sekvence, respektive periodě primární sekvence. Rozsah zjištěného frekvenčního posuvu závisí na vzorkovací periodě a je tedy $\pm 500 \text{ Hz}$ pro L1 C/A a $\pm 125 \text{ Hz}$ pro E1-OS.

Nevýhodou metody je její nevhodnost pro zpracování signálu v reálném čase kvůli výpočetní náročnosti a horší odolnost vůči nízkému poměru signál-šum oproti ostatním popisovaným metodám frekvenční synchronizace. Naopak výhodou je poměrně velký rozsah zjištěného frekvenčního posuvu a odolnost vůči symbolovým přechodům.

2.3.2.2 Synchronizace pomocí derivace fáze

Jak je známo, frekvence signálu odpovídá časové derivaci fáze tohoto signálu, čehož bylo využito při vytvoření metody frekvenční synchronizace popsané v této podkapitole.

Jedná se o synchronizační smyčku, která byla vytvořena na základě Costasovy smyčky, s kterou se liší pouze řešením fázového diskriminátoru, jehož výstup je úměrný rozdílu fází aktuálního a předchozího vzorku vstupujícího do diskriminátoru. Fázový diskriminátor je definován vztahem 2.22, přičemž význam proměnných je shodný s významy ve vztahu 2.12 a dolní index n označuje pořadí vzorku.

$$u_d = K_d \cdot \left[\text{sgn}(I_n) \cdot \left(\frac{\text{sgn}(Q_n) \cdot Q_n^2}{I_n^2 + Q_n^2} \right) - \text{sgn}(I_{n-1}) \cdot \left(\frac{\text{sgn}(Q_{n-1}) \cdot Q_{n-1}^2}{I_{n-1}^2 + Q_{n-1}^2} \right) \right] \quad (2.22)$$

Stejně jako u Costasovy smyčky, k návrhu smyčky je potřeba určit čtyři již známé parametry - K_0 , K_d , a , b , a to způsobem shodným s Costasovou smyčkou. Jediným drobným rozdílem v návrhu je význam proměnné f_L , která v případě frekvenční synchronizace nepředstavuje frekvenční rozmezí vstupního signálu, ale je úměrná rychlosti, s jakou je smyčka schopná se synchronizovat.

S Costasovou smyčkou je také shodný způsob nahlížení na fázi signálu a její převádění do intervalu $\langle -90^\circ; +90^\circ \rangle$ kvůli odolnosti vůči bitovým přechodům. Intervalu $\langle -90^\circ; +90^\circ \rangle$ odpovídá v diskriminátoru interval $\langle -1; +1 \rangle$, přičemž na činitele rozdílu ve vztahu 2.22 je nahlíženo jako na proměnné, jež mohou nabývat hodnot právě v tomto intervalu, což má za následek, že například platí $80^\circ - (-20^\circ) = -80^\circ$, kdy aktuální fáze signálu je 80° a fáze předchozího vzorku signálu je -20° , situace je diskriminátorem vyhodnocena jako pohyb fázoru v záporném směru o 80° za dobu trvání odpovídající vzorkovací periodě signálu vstupujícího do smyčky. Z uvedeného příkladu plyne pro maximální správně zjistitelnou hodnotu frekvenčního posuvu $f_{off,max}$ vztah 2.23, kde T_s je vzorkovací perioda vstupního signálu.

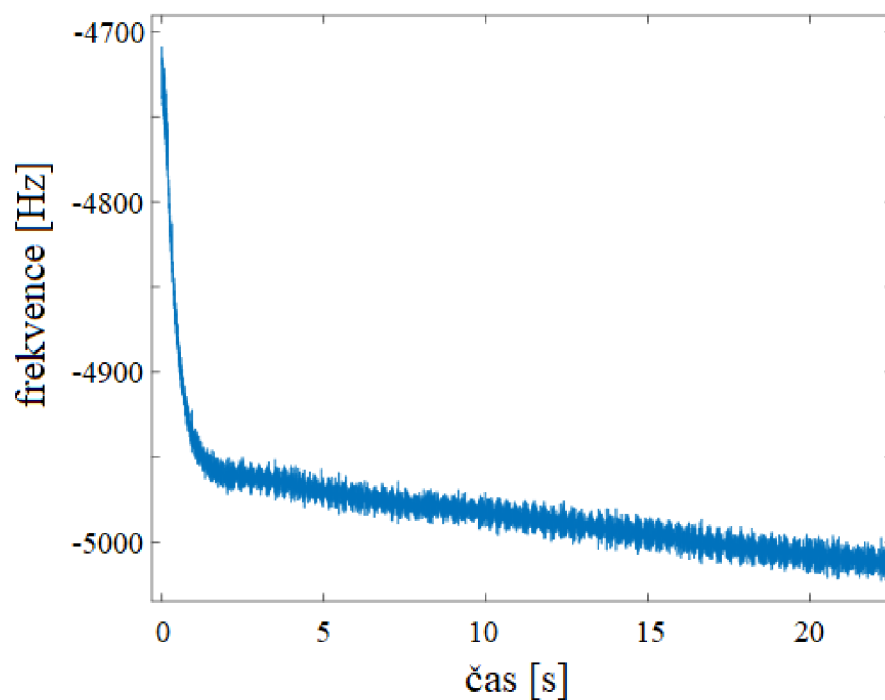
$$f_{off,max} = \pm \frac{1}{4T_s} \quad (2.23)$$

Volba hodnoty f_L představuje kompromis mezi rychlostí synchronizace a šumovou šířkou pásma. Hodnoty f_L byly pro oba zkoumané navigační signály určeny experimentálně simulacemi v MATLABu a to tak, aby k synchronnosti frekvencí došlo přibližně do 1 sekundy, z čehož pro L1 C/A signál vyšlo $f_L = 15 \text{ Hz}$ a pro E1-OS signál $f_L = 11 \text{ Hz}$. Jelikož byla odhadovaná frekvence na výstupu smyčky zatížena určitou mírou šumu, která by mohla komplikovat fázovou synchronizaci následující smyčky (Costasova nebo PLL), byl pro odstranění šumu v signálu předávajícím informaci o frekvenčním posuvu navržen filtr dolní propusti. Dolní propust byla navržena v MATLABu jako filtr prvního řádu Butterworthovy aproximace s experimentálně určenou časovou konstantou $\tau = 0.02 \text{ s}$.

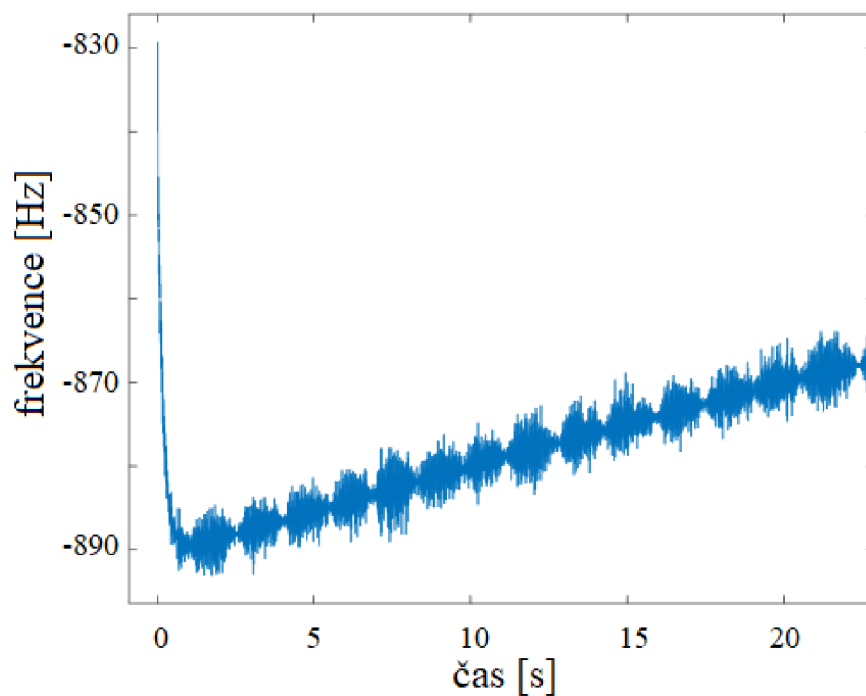
Po získání hodnoty frekvenčního posuvu výše uvedeným způsobem a předání této hodnoty smyčkám fázové synchronizace byla fáze zkoumaných signálů úspěšně sledována.

Příklady průběhů frekvenční synchronizace pro oba zkoumané signály jsou na obrázcích 2.16 a 2.17, ze kterých je zřejmé, že počáteční frekvenční posuv je odstraněn během 1 – 2 sekund a poté už probíhá frekvenční sledování měnícího se nosného kmitočtu.

Vzorkovací perioda signálu vstupujícího do smyčky odpovídá v případě GPS jedné periodě L1 C/A sekvence a v případě Galilea jedné periodě primární sekvence, z čehož podle vztahu 2.23 plynou pro maximální možné frekvenční posuvy hodnoty $\pm 250 \text{ Hz}$ pro GPS a $\pm 62,5 \text{ Hz}$ pro Galileo.



Obr. 2.16 Frekvenční synchronizace s derivací fáze, L1 C/A.



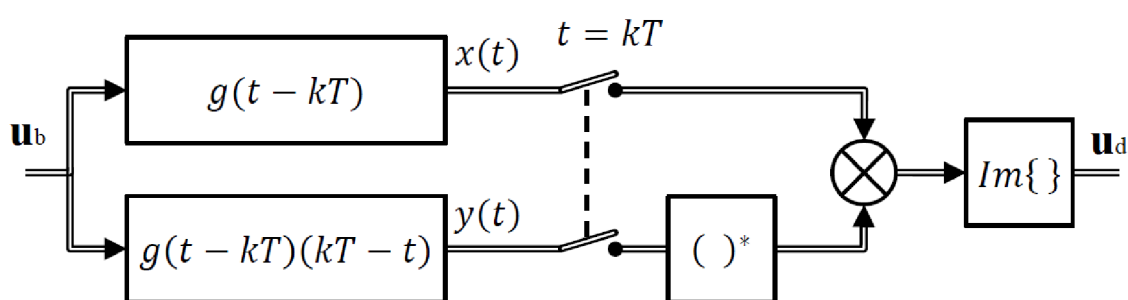
Obr. 2.17 Frekvenční synchronizace s derivací fáze, E1-OS.

Drobnou nevýhodou popisované metody je ne zcela maximální odolnost vůči nízkému poměru signál-šum. Mezi výhody patří nízká výpočetní náročnost, odolnost

vůči bitovým přechodům a schopnost sledovat signál E1-OS bez znalosti fáze sekundárního kódu.

2.3.2.3 Frekvenční estimátor ML

Další možností frekvenční synchronizace je použití frekvenčního estimátoru ML (*Maximum Likelihood*). Jedná se o synchronizační smyčku, jež je odvozena na základě kritéria ML, čímž se rozumí, že výstupem je odhadovaná frekvence, která ze všech možností tvoří s největší pravděpodobností nosnou frekvenci signálu [18]. Obecné blokové schéma smyčky odpovídá Costasově smyčce na obrázku 2.6, zásadní rozdíl však spočívá ve vnitřní struktuře fázového diskriminátoru, který je znázorněn na obrázku 2.18 [18].



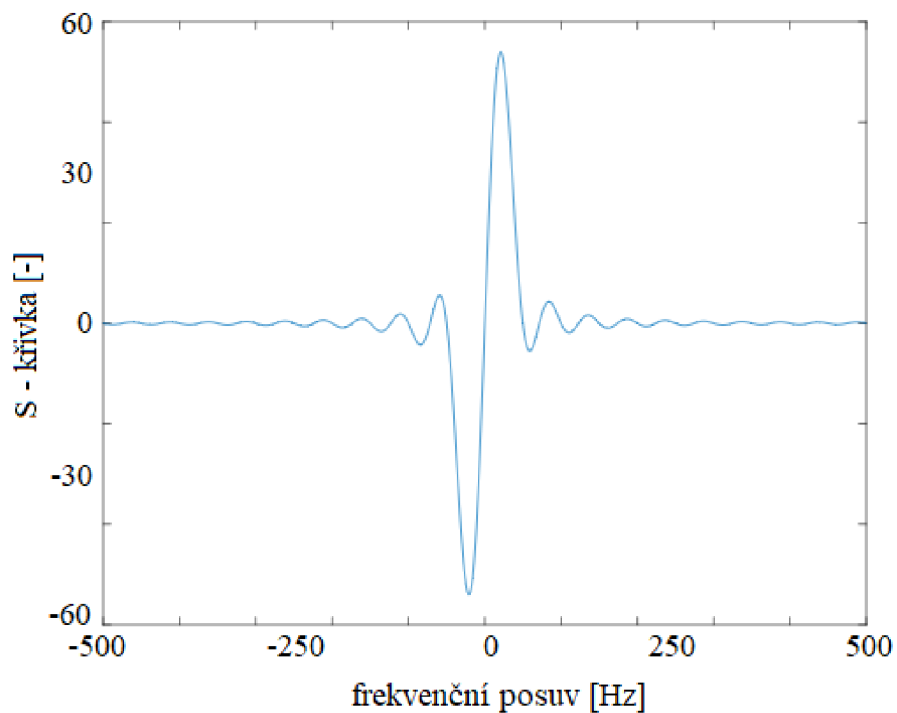
Obr. 2.18 Fázový diskriminátor frekvenčního estimátoru ML.

Výraz $g(t - kT)$ na obrázku 2.18 představuje impulzní charakteristiku přizpůsobeného filtru na přijímací straně a $g(t - kT) \cdot (kT - t)$ impulzní charakteristiku jeho frekvenčně derivované verze. Výstup druhého ze zmíněných filtrů je komplexně sdružen a následně jsou výstupy obou filtrů vynásobeny. Chybovým výstupním signálem u_d je imaginární část tohoto součinu. Graf závislosti chybového signálu u_d na frekvenčním posuvu vstupního signálu u_b je označován podle tvaru jako S – křivka. [19]

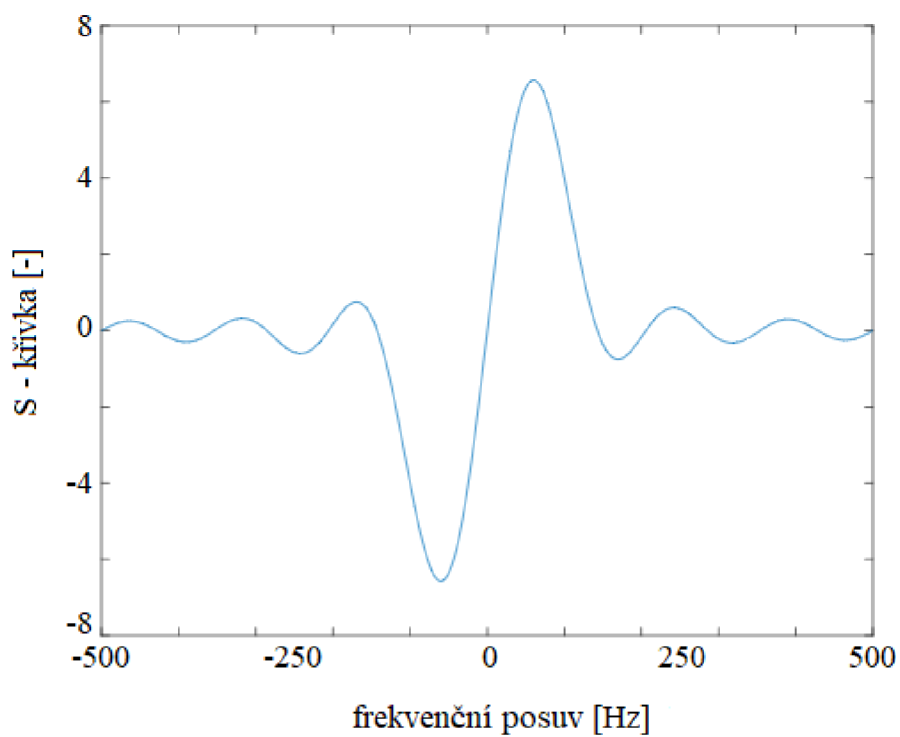
Vzhledem k tomu, že zkoumané navigační signály nepoužívají přizpůsobený filtr, impulzní charakteristikám $g(t - kT)$ byly přiřazeny koeficienty o hodnotě 1 s délkou odpovídající počtu derozprostřených vzorků signálů použitých pro výpočet chybového signálu. Výraz $(kT - t)$ zjednodušeně řečeno představuje lineární posloupnost bodů s nulovou střední hodnotou a procházející počátkem souřadnic. Délka posloupnosti odpovídá délce impulzní charakteristiky $g(t - kT)$.

Simulacemi v MATLABu se ukázalo, že s rostoucím počtem derozprostřených vzorků jednak připadajících na symbol, a jednak použitých pro výpočet chybového signálu, roste i odolnost smyčky vůči symbolovým přechodům a nízkému poměru signál-šum. Na druhou stranu bylo také zjištěno, že s rostoucím počtem derozprostřených vzorků použitých pro výpočet chybového signálu se zmenšuje rozsah zjistitelného frekvenčního posuvu, jak lze vidět na příkladech S – křivek na obrázcích

2.19 a 2.20 pro L1 C/A signál. Rozsah zjistitelného frekvenčního posuvu se také zmenšuje s rostoucí vzorkovací periodou vzorků na vstupu.



Obr. 2.19 S - křivka pro délku impulsní charakteristiky 20, L1 C/A.

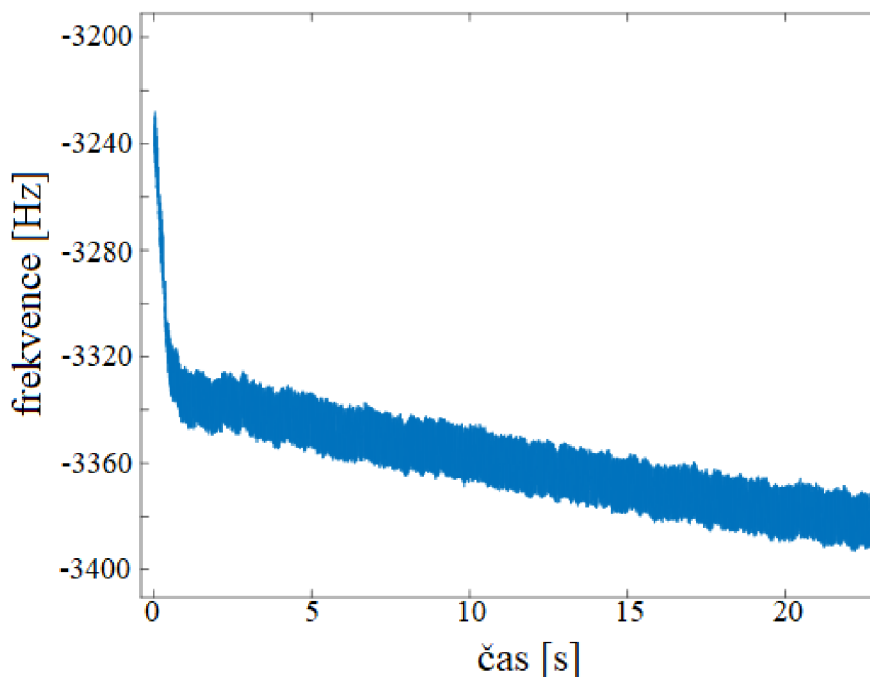


Obr. 2.20 S - křivka pro délku impulsní charakteristiky 7, L1 C/A.

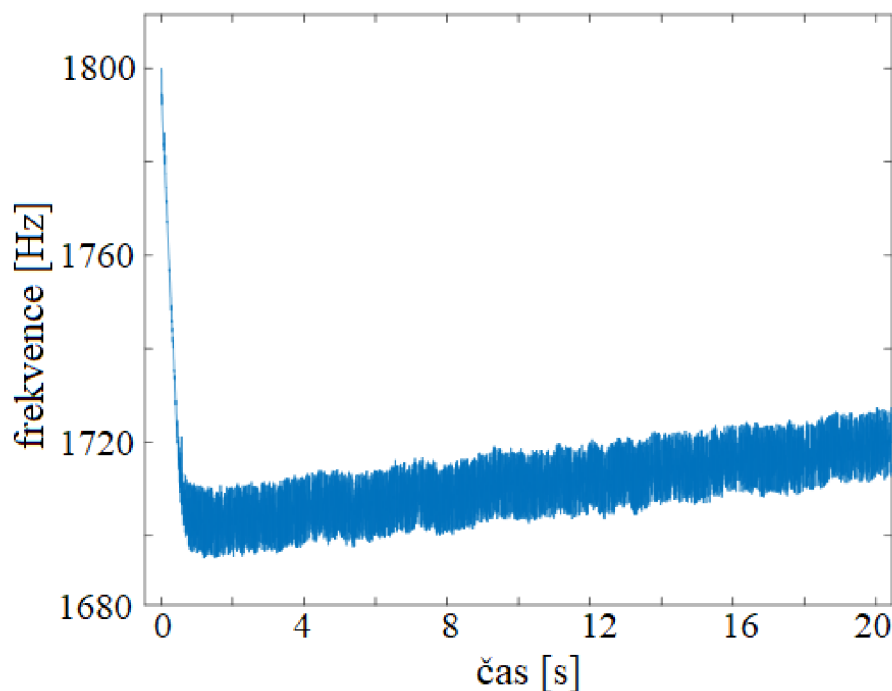
Frekvenční estimátor ML byl pro L1 C/A signál navržen s hodnotou $f_L = 10 \text{ Hz}$, vzorkovací periodou $T_s = 1 \text{ ms}$ a s impulsní charakteristikou $g(t - kT)$ délky 7. V případě E1-OS signálu byl estimátor navržen s hodnotou $f_L = 7,5 \text{ Hz}$, vzorkovací periodou $T_s = 4 \text{ ms}$ a s impulsní charakteristikou $g(t - kT)$ délky 2. Výstup fázového diskriminátoru je výkonově normován.

Výše uvedené vstupní parametry pro návrh smyček byly určeny experimentálně simulacemi v MATLABu jako kompromis mezi rychlostí synchronizace, rozsahem zjistitelného frekvenčního posuvu a odolností vůči nízkému poměru signál-šum. Návrh smyček byl proveden stejným postupem jako v případě Costasovy smyčky. Navržené smyčky spolehlivě provedly frekvenční synchronizaci pro všechny nalezené družice s frekvenčním posuvem $\pm 100 \text{ Hz}$ přibližně během jedné sekundy. Stejně jako v případě synchronizace pomocí derivace fáze, informace o frekvenčním posuvu signálu byla před odesláním do Costasovy smyčky filtrována Butterworthovou dolní propustí prvního řádu s časovou konstantou 0,02 sekund. Příklady průběhů frekvenční synchronizace a jejího následného sledování jsou pro oba zkoumané signály na obrázcích 2.21 a 2.22.

Výhodou frekvenčního ML estimátoru je oproti předchozím řešením teoreticky větší odolnost vůči nízkému poměru signál-šum [19]. Nevýhodou vůči metodě s derivací fáze je nutnost znát fázi sekundárního kódu v případě E1-OS, protože jeden chip sekundární sekvence je rozprostřen pouze jednou periodou primární sekvence, což z pohledu ML estimátoru znamená, že po derozprostření primární sekvencí připadá na každý symbol pouze jeden vzorek signálu. Tak nízký počet vzorků na symbol vede k vysokému počtu skokových změn fáze o 180° , kvůli kterým fázový diskriminátor generuje chybový signál o vysoké hodnotě, kterou smyčkový filtr není schopen odstranit.



Obr. 2.21 Sledování frekvence s ML estimátorem, L1 C/A.



Obr. 2.22 Sledování frekvence s ML estimátorem, E1-OS.

2.3.2.4 Kvadrikorelátor

Poslední popisovanou možností frekvenční synchronizace je použití tak zvaného kvadrikorelátoru. Jedná se o sledovací smyčku obecným blokovým schématem a návrhem opět shodnou s Costasovou smyčkou. Fázový diskriminátor je definován vztahem 2.24, přičemž význam proměnných je shodný s významy ve vztahu 2.12. [17]

$$u_d = K_d \cdot \left(\frac{dI}{dt} \cdot Q \right) \quad (2.24)$$

Podle vztahu 2.24 je chybový signál na výstupu diskriminátoru úměrný součinu imaginární složky a časové derivace složky reálné. Platí-li pro složky I a Q vztahy 2.25 a 2.26, je možné výstupní chybový signál vyjádřit vztahem 2.27, přičemž ω_{off} značí frekvenční posuv v úhlové míře. [17]

$$I = \sin(\omega_{off} \cdot t) \quad (2.25)$$

$$Q = \cos(\omega_{off} \cdot t) \quad (2.26)$$

$$u_d = \frac{1}{2} \cdot \omega_{off} \cdot [1 + \cos(2 \cdot \omega_{off} \cdot t)] \quad (2.27)$$

Vzhledem k tomu, že signál u_d je filtrován smyčkovým filtrem, lze uvažovat pouze jeho stejnosměrnou část a vyjádřit ho vztahem 2.28.

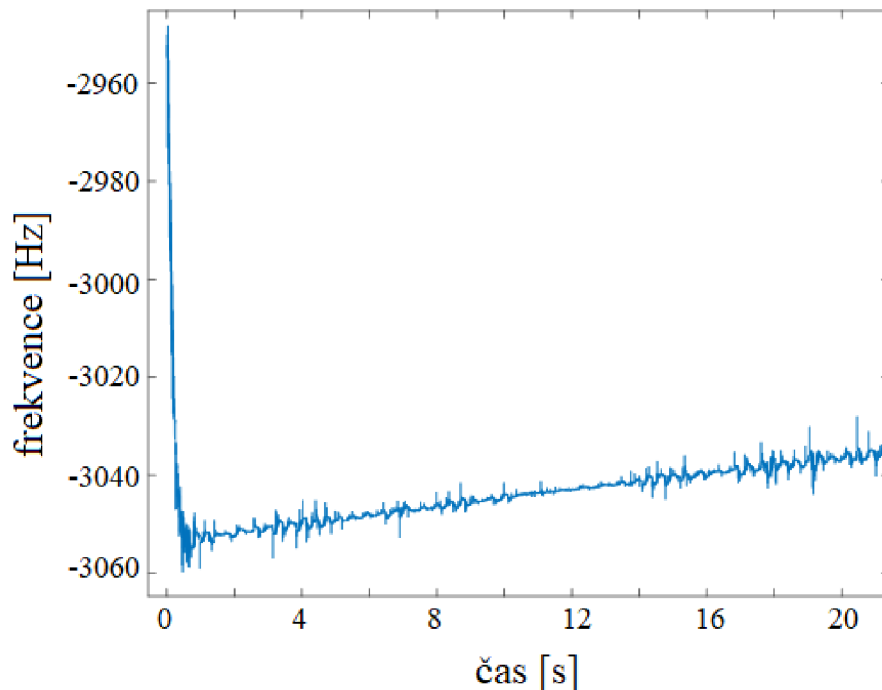
$$u_d = \frac{1}{2} \cdot \omega_{off} \quad (2.28)$$

Popisovaná metoda není odolná vůči symbolovým přechodům, a proto ji je možné použít pouze v případě synchronizace na pilotním signálu E1-OS systému Galileo. [17]

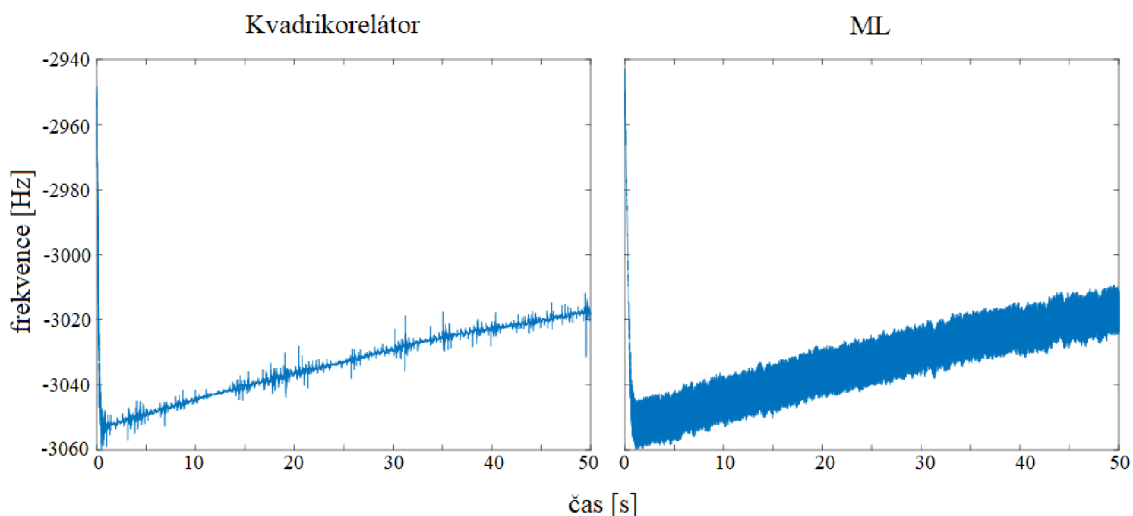
Signál u_d byl normován do intervalu $\langle -1; +1 \rangle$ vydělením jeho maximální možnou hodnotou $u_{d,max} = \frac{1}{2} \cdot \omega_{off,max} = \frac{1}{2} \cdot \frac{\pi}{T_p}$, kde T_p je perioda primární rozprostírací sekvence E1-OS.

Kvadríkorelátör byl pro signál E1-OS navržen s hodnotou $f_L = 25 \text{ Hz}$ a vzorkovací periodou $T_s = 4 \text{ ms}$.

Výše uvedené vstupní parametry pro návrh smyčky byly určeny experimentálně simulacemi v MATLABu jako kompromis mezi rychlostí synchronizace a odolností vůči nízkému poměru signál-šum. Návrh smyčky byl proveden stejným postupem jako v případě Costasovy smyčky. Navržená smyčka spolehlivě provedla frekvenční synchronizaci pro všechny nalezené družice s frekvenčním posuvem $\pm 100 \text{ Hz}$ přibližně během jedné sekundy. Stejně jako v případě synchronizace pomocí derivace fáze, informace o frekvenčním posuvu signálu byla před odesláním do Costasovy smyčky filtrována Butterworthovou dolní propustí prvního řádu s časovou konstantou 0,02 sekund. Příklad průběhu frekvenční synchronizace s následným sledováním frekvence a porovnání s ML estimátorem jsou pro E1-OS signál na obrázcích 2.23 a 2.24.



Obr. 2.23 Sledování frekvence kvadríkorelátorem, E1-OS.



Obr. 2.24 Porovnání Kvadríkorelátoru a ML estimátoru, E1-OS.

Jak lze vidět na obrázku 2.24, výhodou kvadríkorelátoru vůči ML estimátoru je větší odolnost vůči nízkému poměru signál-šum při srovnatelné rychlosti synchronizace. Naopak nevýhodou kvadríkorelátoru je nutná absence symbolových přechodů, což je důvod, proč nemůže být použit pro signál L1 C/A.

2.3.3 Volba metody

Co se týče fázové synchronizace, pro L1 C/A signál je potřeba použít Costasovu smyčku kvůli přítomnosti bitových přechodů. Pro fázovou synchronizaci E1-OS signálu lze použít navíc i PLL smyčku, která se jeví jako optimálnější řešení díky větší odolnosti vůči nízkému poměru signál-šum.

Z hlediska frekvenční synchronizace je, vzhledem k nejvyšší odolnosti vůči nízkému poměru signál-šum, neoptimálnějším řešením kvadríkorelátor, který však není odolný vůči bitovým přechodům, a proto jeho použití připadá v úvahu pouze pro E1-OS signál. Ze zbývajících metod, které jsou schopny sledovat i signál s bitovými přechody, má nejvyšší odolnost vůči nízkému poměru signál-šum ML estimátor, jehož použití se jeví jako optimální volba pro L1 C/A signál.

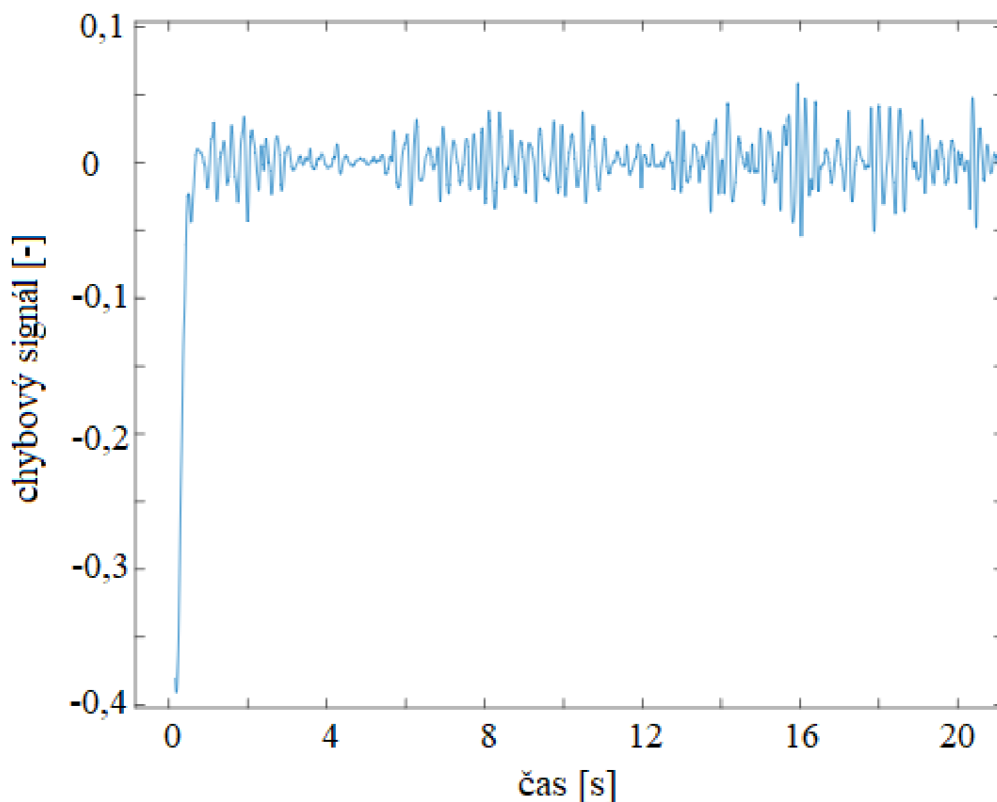
2.4 Detekce synchronizace

V této podkapitole jsou popsány způsoby, jakými je prováděna kontrola fázové a frekvenční synchronizace.

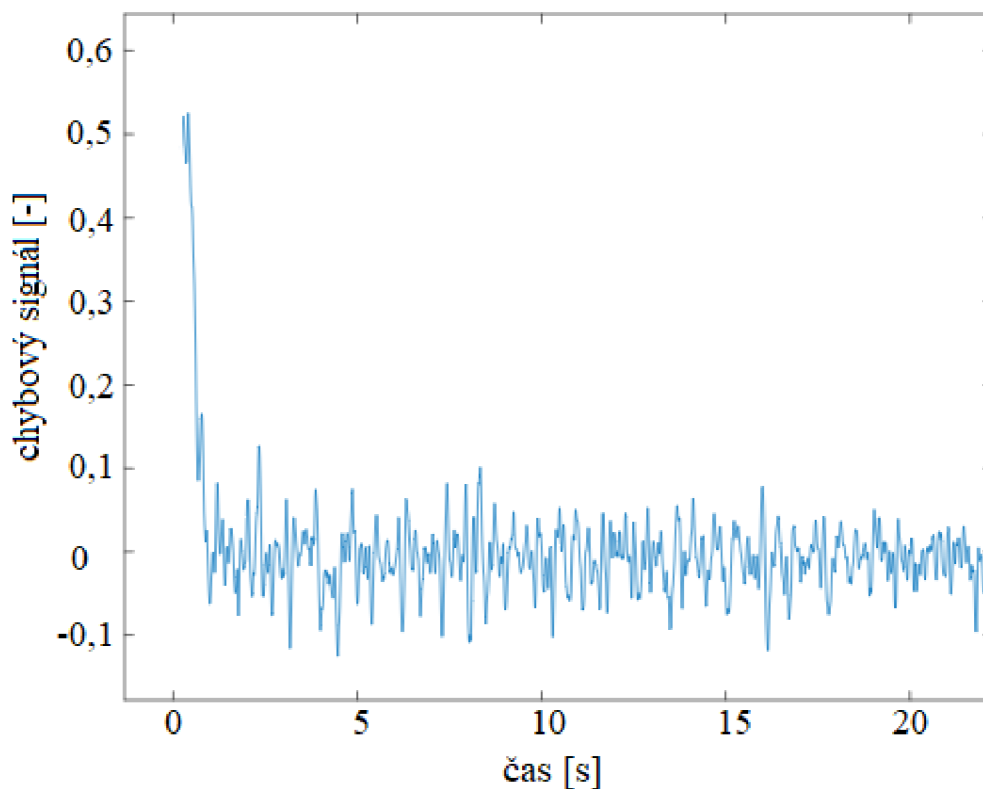
2.4.1 Detekce frekvenční synchronizace

K úspěšnému přechodu ze smyčky frekvenční synchronizace ke smyčce fázové synchronizace je potřeba vyhodnotit, kdy je přijímaná nosná frekvence již dostatečně synchronní s její lokálně generovanou verzí.

Pro detekci frekvenční synchronizace byl použit filtrovaný chybový signál na výstupu fázového diskriminátoru. Simulacemi v MATLABu se pro filtraci osvědčila dolní propust tvořena kaskádou čtyř Butterworthových dolních propustí prvního řádu s časovou konstantou 0,02 sekund. Popisovaný filtr by bylo možné vytvořit i jako jednu dolní propust čtvrtého řádu, avšak takové řešení je v případě rekurzivních filtrů, mezi které se řadí i Butterworthova aproximace, mnohem náchylnější k nestabilitě vlivem drobných změn koeficientů způsobených konečnou přesností datových typů. Příklady časových průběhů filtrovaného chybového signálu během frekvenční synchronizace jsou pro zkoumané navigační signály na obrázcích 2.25 a 2.26. Frekvenční synchronnosti je dosaženo v okamžiku, kdy chybový signál má nulovou střední hodnotu. Simulacemi v MATLABu se ukázalo jako dostačující tento stav detektovat jako okamžik první změny polarity filtrovaného chybového signálu.



Obr. 2.25 Detekce frekvenční synchronizace, E1-OS.



Obr. 2.26 Detekce frekvenční synchronizace, L1 C/A.

Z obrázků 2.25 a 2.26 plyne, že frekvenční synchronizace je pro oba zkoumané signály provedena přibližně během jedné sekundy. V těchto případech byl použit kvadrikorelátor pro E1-OS a ML estimátor pro L1 C/A.

2.4.2 Detekce fázové synchronizace

Fázovou synchronnost přijímané a lokálně generované nosné frekvence je potřeba průběžně kontrolovat za účelem její včasné obnovy při její případné ztrátě ve sledovacích smyčkách.

Pro určení míry fázové synchronizace bylo použito vyjádření fáze komplexního signálu v základním pásmu jako $\cos(2\alpha)$, kde α je fáze signálu. [20] Pro zmíněný výraz platí všeobecně známý vztah 2.29, který je možné vyjádřit pomocí složek I a Q vztahem 2.30, protože tyto složky představují kosinus a sinus fáze signálu.

$$\cos(2\alpha) = (\cos x)^2 - (\sin x)^2 \quad (2.29)$$

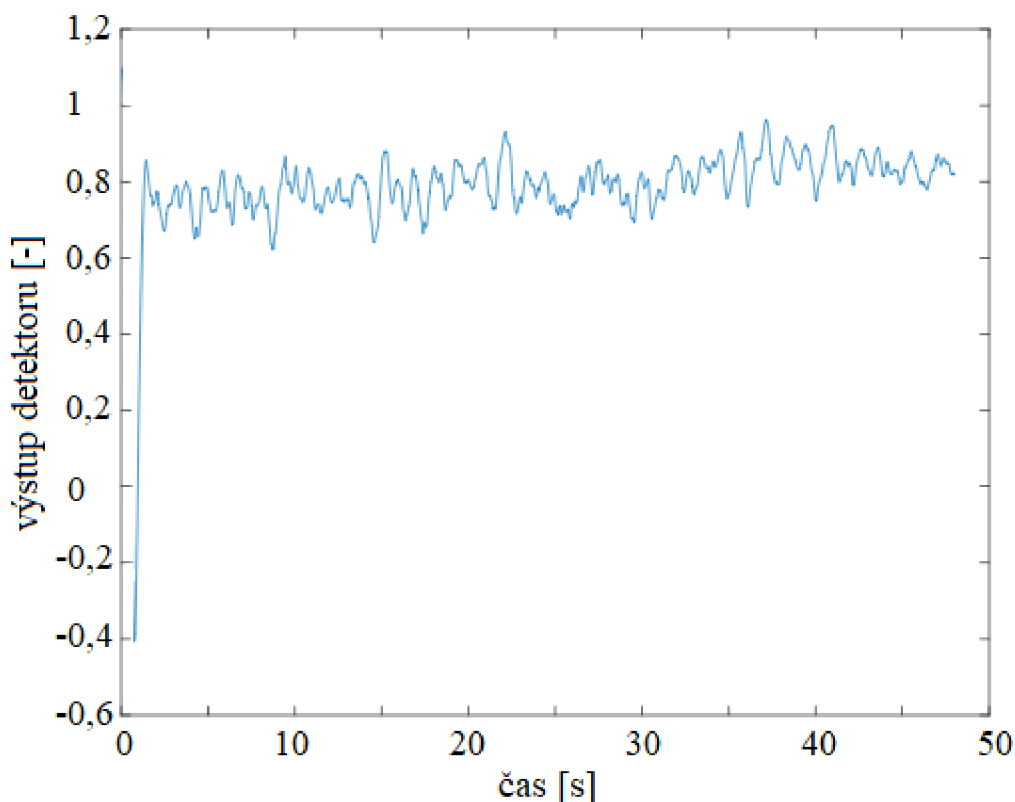
$$\cos(2\alpha) = I^2 - Q^2 \quad (2.30)$$

Vzhledem k různé velikosti vstupního signálu byl výraz ve vztahu 2.30 normován a složky I a Q byly nahrazeny sumou těchto složek po dobu trvání 20 milisekund, což je

kompromis mezi akceptovatelnou mírou šumu a rychlostí aktualizace této informace. Výsledkem je vztah 2.31. [20]

$$\cos(2\alpha) = \frac{[\sum_{20ms} I]^2 - [\sum_{20ms} Q]^2}{[\sum_{20ms} I]^2 + [\sum_{20ms} Q]^2} \quad (2.31)$$

Z uvedených vztahů plyne, že v případě fázové synchronnosti nosných frekvencí se popisovaný parametr blíží hodnotě 1. Pro potlačení míry šumu v tomto parametru se simulacemi v MATLABu ukázalo jako vhodné výstup detektoru filtrovat dvěma kaskádně řazenými Butterworthovými dolními propustmi prvního řádu s časovou konstantou 0,02 sekund. Příklad časového průběhu výsledného parametru je na obrázku 2.27.



Obr. 2.27 Detekce fázové synchronizace, E1-OS.

Z grafu na obrázku 2.27 lze vidět, že sledovaný signál není přesně ve fázi, přičemž hodnota parametru 0,8 odpovídá přibližnému rozdílu fází 18° . Tento fakt je důsledkem snahy o, pokud možno, nejmenší šumovou šířku pásma sledovací smyčky. Taková smyčka již pouze pomalu kompenzuje menší fázové rozdíly, jak je vidět na pomalu rostoucí tendenci zmiňovaného grafu. Při simulacích v MATLABu nebyl shledán žádný negativní vliv této synchronizační nedokonalosti na sledování nosné frekvence ani na dekodování dat.

2.5 Bitová synchronizace

Jak již bylo v této práci zmíněno, datové bity L1 C/A signálu jsou rozprostřeny 20 periodami rozprostírací sekvence, z čehož plyne, že při počáteční fázi příjmu není jisté, která z těchto 20 period rozprostírací sekvence představuje začátek bitu.

V ideálním případě by stačilo čekat na změnu polaritu výstupu odpovídajícího korelátoru a danou periodu prohlásit za začátek bitu, avšak v reálném signálu může vlivem velké míry šumu docházet k náhodným změnám polaritu i mimo bitové přechody, což by mohlo vést k chybné bitové synchronizaci.

Metoda použitá v této práci spočívá ve vyhodnocování četnosti změn polaritu pro všech 20 period v daném časovém intervalu, po kterém je za skutečný začátek nového bitu prohlášena ta perioda rozprostírací sekvence, jejíž pozici odpovídá výrazně vyšší četnost změn polaritu vzhledem ke všem ostatním pozicím. V případě, že žádná pozice rozprostírací sekvence nevyhovuje daným kritériím, následuje další vyhodnocovací interval, dokud není dosaženo bitové synchronizace. [1]

Pro L1 C/A signál se simulacemi v MATLABu osvědčilo použít pro vyhodnocování úsek signálu o délce odpovídající jedné sekundě a jako kritérium pro pozici bitového přechodu, aby četnost změn polaritu dané pozice byla alespoň 1,5x vyšší než četnosti všech ostatních pozic. U většiny družic došlo k bitové synchronizaci do 2 sekund, u několika družic se slabým signálem trvala maximálně 5 sekund.

Pro E1-OS signál se není třeba bitovou synchronizací zabývat, protože každý symbol datového signálu je rozprostřen právě jednou periodou rozprostírací sekvence, která je synchronní s primární sekvencí pilotního signálu. V případě E1-OS signálu je vhodnější mluvit o symbolech, protože bity přenášené kanálem jsou kódované pomocí konvolučního kodéru. [12]

2.6 Kanálové dekódování

Posledním krokem před samotnou analýzou dat z družic je kanálové dekódování přijatých bitů (symbolů), které spočívá v odstranění zabezpečovací bitů a kontrole bezchybnosti přijatých dat.

2.6.1 Signál L1 C/A

Navigační data tohoto signálu jsou vysílány v rámcích o délce 1500 bitů a obsahujících 5 subrámců o délce 300 bitů. Každý subrámeček se skládá z 10 slov o délce 30 bitů. Každé slovo se skládá z 24 datových bitů a 6 paritních bitů na konci. Každý subrámeček začíná slovem TLM (*telemetry*), jehož prvních 8 datových bitů je tvořeno preambulí, jež je tvořena sekvencí bitů 10001011. Tato preambule slouží pro detekci začátku subrámečku, přičemž kvůli fázové nejistotě je potřeba v signálu hledat i její bitově negovanou verzi. Subrámeček je s jistotou nalezen, jestliže se na jeho začátku i za jeho koncem nachází stejná preambule. [2]

Získání užitečných dat a kontrola parity je provedena pomocí exkluzivního součtu podle vzorců v tabulce 2.1, kde D_n značí bit na pozici n v aktuálně přijatém slově, D_n^* značí bit na pozici n v předchozím přijatém slově a d_n značí bit na pozici n po odstranění fázové nejistoty [6].

Tabulka 2.1 Kontrola parity, L1 C/A.

d_1	=	$D_1 \oplus D_{30}^*$
d_2	=	$D_2 \oplus D_{30}^*$
d_3	=	$D_3 \oplus D_{30}^*$
•		•
•		•
•		•
•		•
D_{24}	=	$d_{24} \oplus D_{30}^*$
D_{25} $\oplus d_{23}$	=	$D_{29}^* \oplus d_1 \oplus d_2 \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_{10} \oplus d_{11} \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{17} \oplus d_{18} \oplus d_{20}$
D_{26} $\oplus d_{24}$	=	$D_{30}^* \oplus d_2 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7 \oplus d_{11} \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{18} \oplus d_{19} \oplus d_{21}$
D_{27} d_{22}	=	$D_{29}^* \oplus d_1 \oplus d_3 \oplus d_4 \oplus d_5 \oplus d_7 \oplus d_8 \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{19} \oplus d_{20} \oplus d_{22}$
D_{28} $\oplus d_{23}$	=	$D_{30}^* \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_6 \oplus d_8 \oplus d_9 \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{17} \oplus d_{20} \oplus d_{21}$
D_{29} $d_{22} \oplus d_{24}$	=	$D_{30}^* \oplus d_1 \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_7 \oplus d_9 \oplus d_{10} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{17} \oplus d_{18} \oplus d_{21} \oplus d_{22} \oplus d_{24}$
D_{30}	=	$D_{29}^* \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_8 \oplus d_9 \oplus d_{10} \oplus d_{11} \oplus d_{13} \oplus d_{15} \oplus d_{19} \oplus d_{22} \oplus d_{23} \oplus d_{24}$

Kontrola parity je úspěšná, jestliže přijaté paritní bity D_{25} až D_{30} jsou shodné s paritními bity vypočítanými podle tabulky 2.1. [6]

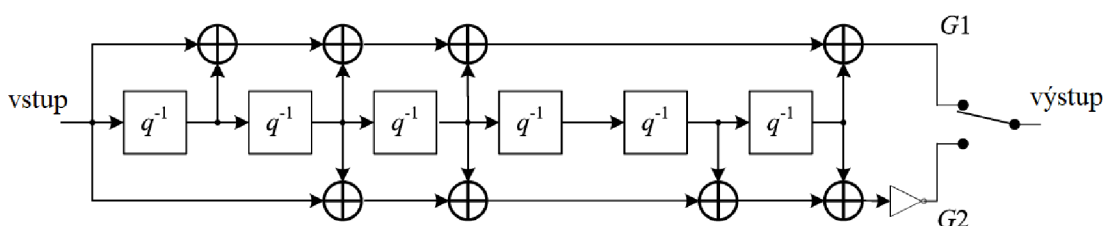
Kanálové dekódování bylo pro signál L1 C/A výše uvedeným způsobem simulováno v MATLABu, přičemž byly pomocí preamble nalezeny všechny subrámcce a paritní kontrola všech slov byla úspěšně provedena pro všechny sledované družice. Nalezené družice, společně s ID přečtených subrámců a jejich hodnotami TOW (*Time Of Week*), jsou uvedeny v tabulce 2.2. Informace uvedené v tabulce 2.2 byly získány ze slova HOW (*Hand Over Word*), které následuje po slově TLM. TOW udává časovou hodnotu od začátku týdne v 6-ti sekundovém násobku, jinými slovy rozdíl hodnot TOW sousedních subrámců musí být vždy 1 [2]. V tabulce 2.2 lze vidět, že u všech nalezených družic rozsah hodnot TOW odpovídá počtu přečtených subrámců, což potvrzuje správnost fungování celého přijímacího řetězce. U družic s hodnotou ID 7 a 10 bylo přečteno o jeden subrámeček méně, protože se jednalo o družice se slabým signálem a bitová synchronizace trvala o několik sekund déle.

Tabulka 2.2 Nalezené družice, L1 C/A.

ID družice	Počet subrámců	Hodnoty TOW
4	11	43970, 43971, ..., 43980
7	10	43971, 43972, ..., 43980
10	10	43971, 43972, ..., 43980
15	11	43970, 43971, ..., 43980
16	11	43970, 43971, ..., 43980
21	11	43970, 43971, ..., 43980
26	11	43970, 43971, ..., 43980
29	11	43970, 43971, ..., 43980

2.6.2 Signál E1-OS

Navigační informace vysílané signálem E1-OS jsou rozděleny do úseků nazývaných substránky. Každá substránka obsahuje celkem 250 symbolů, z nichž prvních 10 symbolů tvoří synchronizační preambuli tvořenou posloupností 0101100000 a zbývajících 240 symbolů je výstup konvolučního kodéru s poměrem $\frac{1}{2}$ a polynomy $G1 = 171o$ a $G2 = 133o$. V 240 symbolech je zakódováno celkem 120 bitů. [12] Blokové schéma použitého konvolučního kodéru je na obrázku 2.28 [12].



Obr. 2.28 Blokové schéma konvolučního kodéru, E1-OS.

Před odesláním je nad symboly každé substránky provedeno blokové prokládání s rozměry 30 sloupců x 8 řádků (zápis do sloupců a čtení z řádků). Synchronizační preambule je z procesu konvolučního kódování a prokládání vynechána. [12]

Na základě prvního bitu obsaženého v substránce se substránky dělí na sudé – 0 a liché – 1. Sudá a lichá substránka tvoří dohromady jednu stránku o délce 240 bitů, přičemž sudá substránka je vysílána jako první. Každá stránka obsahuje několik segmentů, z nichž nejdůležitější je datový segment o délce 128 bitů a CRC segment obsahující 24 zabezpečovacích bitů pro datový segment. Generující polynom $G(x)$ pro CRC ochranu je vyjádřen pomocí vztahů 2.32 a 2.33. [12]

$$G(x) = (1 + x)P(x) \quad (2.32)$$

$$P(x) = x^{23} + x^{17} + x^{13} + x^{12} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^3 + 1 \quad (2.33)$$

Navigační data byla z přijatých symbolů získána pomocí MATLABu za použití inverzních operací k výše uvedeným operacím kanálového kódování.

Prvním krokem bylo nalezení počátků jednotlivých substránek pomocí synchronizační preamble. Nalezená substránka musela obsahovat synchronizační preamble jednak na jejím začátku, a jednak za jejím koncem, přičemž musel být brán zřetel na fázovou nejistotu.

Po odstranění synchronizační preamble bylo provedeno inverzní prokládání, které spočívalo v zápisu symbolů do řádků o délce 30 a jejich následném vyčítání ze vzniklého bloku po sloupcích o délce 8.

O něco složitějším krokem, který následoval, bylo dekodování symbolů z konvolučního kodéru, k čemuž byl v MATLABu vytvořen Viterbiho dekodér. Dekodérem jsou pro každý možný stav konvolučního kodéru vyhodnocovány Hammingovy vzdálenosti přijaté dvojice symbolů a dvojice symbolů, která by měla být přijata v případě, že se konvoluční kodér nachází po odeslání aktuálně přijaté symbolové dvojice právě v tomto stavu, a to pro obě možnosti předchozích stavů, ze kterých mohl kodér při odeslání symbolů přejít do stavu aktuálního. Stavem konvolučního kodéru se rozumí hodnota jeho posuvného registru bez uvažování bitu na jeho vstupu, který je kódován při odeslání aktuální dvojice symbolů. Hodnotě kódovaného bitu fakticky odpovídá nejvyšší bit aktuálního stavu kodéru. Takto vypočítané Hammingovy vzdálenosti se nazývají metriky větví. Pro každý stav zůstává zachována ta větev (přechod z předchozího stavu), pro niž je součet její metriky a metriky cesty daného předchozího stavu menší, přičemž v případě stejných hodnot součtů metrik je vybrána náhodně jedna z nich, a metrice cesty zkoumaného současného stavu je přiřazena tato hodnota součtu. Výsledkem popisovaného procesu je tak zvaný mřížový diagram jednotlivých cest tvořených různými sekvencemi stavů posuvného registru. [21]

Po přijetí každé symbolové dvojice je na výstupu Viterbiho dekodéru 1 bit, který byl na straně konvolučního kodéru s největší pravděpodobností kódován N kroků zpět, kde N je délka jeho posuvného registru. Kvůli tomuto zpoždění se na konci řetězce bitů kódovaných vysílačem nachází N nulových bitů (anglicky označovaných jako *tail bits*) umožňujících kompletní kódování posledního bitu nesoucího nějakou informaci. Při dekodování bitu na straně přijímače dekodér vybere v mřížovém diagramu nejpravděpodobnější aktuální stav, což je stav s nejnižší metrikou cesty, a postupuje N kroků zpět po cestě mřížového diagramu, která ústí do tohoto stavu, na pozici, které odpovídá stav, jehož nejvyšší bit je s největší pravděpodobností dekodovaný bit. [21]

Posledním krokem byla po sestavení stránek z dekódovaných bitů CRC kontrola datového segmentu pomocí vestavěné funkce v MATLABu za použití generujícího polynomu $G(x)$ definovaného vztahy 2.32 a 2.33.

Během výše popisovaného kanálového dekódování signálu E1-OS v MATLABu byly bez chyby přečteny všechny stránky všech nalezených družic s výjimkou jedné jediné stránky u družice číslo 21 s velmi nízkou kvalitou signálu. U této družice byla zmíněná stránka zatížena příliš velkým počtem chyb, které Viterbiho dekodér nebyl schopen opravit, následkem čehož nebyla úspěšná ani CRC kontrola datových segmentů těchto stránek.

Nalezené družice společně s počty přečtených stránek, výsledky CRC kontroly a počty symbolových chyb v signálovém úseku, které byly odhadnuty pomocí metrik ve Viterbiho dekodéru, jsou uvedeny v tabulce 2.3.

Tabulka 2.3 Nalezené družice, E1-OS.

ID družice	Počet stránek	CRC	Počet symbolových chyb
4	23	vždy úspěšné	2
5	23	vždy úspěšné	19
9	23	vždy úspěšné	0
11	22	vždy úspěšné	4
14	23	vždy úspěšné	3
21	19	vždy úspěšné s výjimkou jedné stránky	19

V tabulce 2.3 lze vidět, že CRC kontrola družic proběhla úspěšně, i přesto že došlo k několika symbolovým chybám, které musely být opraveny Viterbiho dekodérem, což potvrzuje jeho správnou funkčnost. Úspěšná CRC kontrola všech družic potvrzuje správnou funkci celého přijímacího řetězce.

3 Implementace softwarového přijímače na FPGA

Tato kapitola se zabývá hardwarovou implementací metod frekvenčního a fázového sledování signálů navigačních družic, jejichž vývoj byl předmětem předchozí kapitoly.

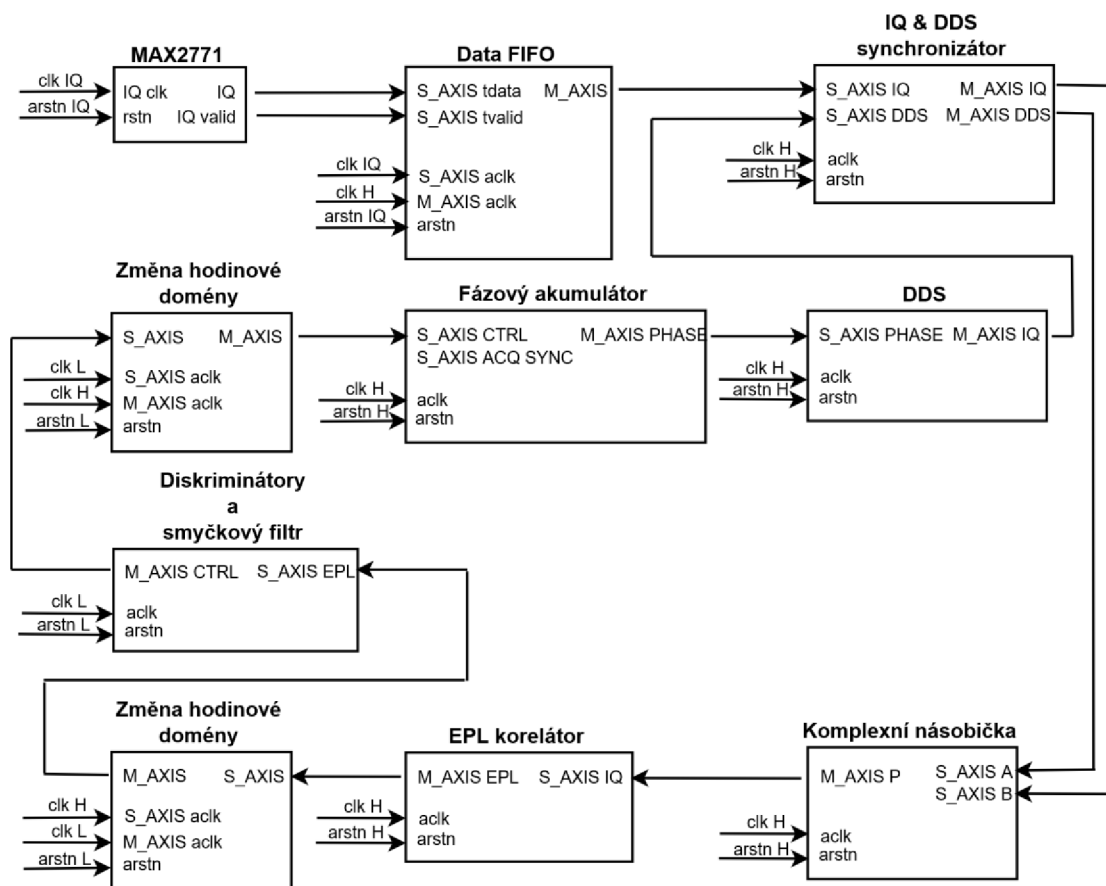
Pro implementaci byla použita vývojová deska Zybo Z7-20 od společnosti Digilent. Vývojová deska je osazena integrovaným obvodem Zynq XC7Z020-1CLG400C od společnosti Xilinx. Tato řada integrovaných obvodů obsahuje, kromě samotného FPGA (*Field Programmable Gate Array*), i procesorovou část skládající se z dvoujádrového mikroprocesoru architektury ARM A9, obvodů pro generování hodinových signálů a celé řady periférií pro komunikační rozhraní. Procesorová část byla využita pouze okrajově a to pro účely testování a generování hodinových signálů pro FPGA.

Pro návrh na hardwarové části bylo použito vývojové prostředí Vivado 2019.1, a pro vytvoření testovací aplikace pro procesorovou část bylo použito vývojové prostředí SDK 2019.1. Obě vývojová prostředí jsou od společnosti Xilinx.

Pro hardwarový popis byl zvolen jazyk VHDL (*VHSIC Hardware Description Language*). Kromě funkčních bloků napsaných v jazyce VHDL bylo v návrhu použito i několik volně dostupných IP (*Intellectual Property*) jader od společnosti Xilinx.

3.1 Struktura softwarového přijímače

Přijímač byl implementován ve formě dvou, na sobě zcela nezávislých, multikanálových přijímacích modulů, přičemž jeden modul zajišťuje zpracování signálu L1 C/A a druhý modul má na starosti signál E1-OS. Oba přijímací moduly mají totožné blokové schéma, které je na obrázku 3.1. Komunikace mezi jednotlivými funkčními bloky je řešena přes AXI4-Stream rozhraní, přičemž data, jež jsou výsledkem části signálového zpracování, které má daný funkční blok na starosti, jsou přenášena po sběrnici označované jako TDATA a doplňující informace týkající se například čísla kanálu nebo čísla zpracovávané družice jsou přenášena po sběrnici TUSER. Sběrnice TDATA a TUSER jsou součástí rozhraní AXI4-Stream. Jako zdroj vstupního signálu slouží komponenta MAX2771. Výsledné symboly po deozproštění by bylo možné získat z bloku obsahujícího diskriminátory a smyčkové filtry v případě, že by na popisovanou část navazovala část zpracovávající navigační informace družic.



Obr. 3.1 Blokové schéma přijímacího modulu

Komponenta označená jako MAX2771 generuje na výstupu IQ vzorky přijímaného navigačního signálu. Navzorkovaný signál vstupuje do vyrovnávací FIFO (*First In First Out*) paměti, která na svém výstupu zprostředkovává vstupní signál již přes AXIS (*AXI4-Stream*) rozhraní, pomocí kterého spolu komunikují všechny zbývající komponenty, přičemž zkratka S_AXIS označuje rozhraní typu SLAVE a zkratka M_AXIS rozhraní typu MASTER. Fázový akumulátor na svém výstupu generuje fázi komplexní exponenciály, která je následně generována komponentou DDS (*Direct Digital Synthesis*). Komponenta s názvem IQ&DDS synchronizátor zaručuje, že do komplexní násobičky vstupují odpovídající dvojice vzorků vstupního signálu a komplexní exponenciály. Na výstupu komplexní násobičky se nachází vstupní navigační signál po provedené frekvenční, případně i fázové, synchronizaci. Synchronizovaný signál vstupuje do EPL (*Early, Prompt, Late*) korelátoru, který provádí derozprostření signálu, který je následně používán pro korekci frekvence komplexní exponenciály a fáze rozprostírací sekvence v bloku obsahujícím diskriminátory synchronizačních smyček a smyčkový filtr.

Všechny zmíněné funkční bloky byly navrženy pro paralelní zpracování signálů od více družic, přičemž jejich maximální počet se odvíjí od taktovací frekvence systému a

vzorkovací frekvence vstupního navigačního signálu. Více na toto téma je uvedeno v kapitole 3.2.

Z blokového schématu na obrázku 3.1 je patrné, že přijímací modul je taktován celkem třemi různými kmitočty, které jsou generovány procesorovou částí. Frekvence IQ clk odpovídá vzorkovací frekvenci vstupního navigačního signálu. Frekvencí clk H je taktována vysokofrekvenční část přijímacího modulu, která je tvořena všemi zbývajících funkčními bloky s výjimkou bloku diskriminátorů a smyčkového filtru, který je taktován nižší frekvencí označované jako clk L, a který je aktivní pouze při dokončení procesu derozprostírání. Důvodem použití dvou taktovacích frekvencí (clk H a clk L) jsou především budoucí optimalizace softwarového přijímače, kdy je například možné snížit využití hardwarových prostředků snížením latence některých částí na úkor snížení maximální taktovací frekvence. Typickým příkladem je manuální nastavení latence děličky diskriminátorů. Jako další, méně významný, důvod pro použití nižších taktovacích frekvencí v místech, kde je to možné, je méně časově náročný proces implementace ve vývojových nástrojích. Ke každému hodinovému signálu náleží jeden resetovací signál, který je s daným hodinovým signálem synchronní. Resetovací signály jsou generovány procesorovou částí za použití IP jader Processor System Reset a v blokovém schématu na obrázku 3.1 jsou označeny jako arstn IQ, arstn H a arstn L. Aktivní úroveň resetovacích signálů odpovídá logické nule.

Podrobnější popisy jednotlivých funkčních bloků jsou pro oba navigační signály uvedeny v následujících podkapitolách.

3.1.1 MAX2771

Název funkčního bloku je odvozen z názvu integrovaného obvodu, jehož použití je plánováno pro příjem a analogově digitální převod signálů navigačních družic. Avšak v době vzniku této práce nebyl zmíněný integrovaný obvod použit a pro odladění hardwarové implementace algoritmů popsaných v kapitole 2 byly použity soubory se vzorky signálů ve formátu wav, které sloužily pro simulace v MATLABu.

Vzhledem k tomu, že integrovaný obvod má na výstupu IQ vzorky s bitovou hloubkou 1 až 2 bity, byla pomocí MATLABu snížena bitová hloubka signálů ve zmíněných souborech na 1 bit, kde logická 0 odpovídá hodnotě +1 a logická 1 odpovídá hodnotě -1 stejně, jako tomu je i u integrovaného obvodu v případě nastavení výstupního formátu na dvojkový doplněk. Úprava bitové hloubky na 2 bity by měla smysl pouze u E1-OS signálu v případě použití i subnosné BOC(6, 1), která však musela být zanedbána kvůli nedostatečné vzorkovací frekvenci, jak již bylo zmíněno v kapitole 2. Důvodem použití 2 bitů v případě kombinace obou BOC subnosných je skutečnost, že v takovém případě již nehraje roli pouze polarita signálu, ale i jeho hodnota, jak lze vidět na obrázku 1.5.

Upravené soubory byly uloženy na SD kartu, která byla v případě potřeby vložena do SD slotu na vývojové desce a čtena za použití knihovny xillffs od společnosti Xilinx. Knihovna je určena pro práci se souborovým systémem typu FAT (*File Allocation*

Table). Čtení vzorků ze souboru je prováděno procesorovou částí a přes rozhraní DMA (*Direct Memory Access*) posíláno na výstup IQ funkčního bloku synchronně s náběžnou hranou hodinového signálu IQ clk, který svou frekvencí odpovídá vzorkovací frekvenci signálu. Platnost výstupních dat je indikována logickou 1 na výstupu IQ valid. DMA přenos je zajištěn IP blokem s názvem AXI Direct Memory Access, který čte vzorky signálu z DRAM (*Dynamic Random Access Memory*) paměti přes AXI HP0 rozhraní procesorové části na základě informací uložených v takzvaných deskriptorech obsahujících počáteční adresu a délku bloku dat a další doplňující informace. Pro nepřetržitý přenos dat byly vytvořeny dva střídající se deskriptory. Jeden deskriptor je vždy používán DMA rozhraním a druhý aktualizován procesorovou částí. Stejně jako vzorky samotného signálu, i deskriptory jsou uloženy v DRAM paměti a zmíněným IP blokem jsou čteny přes AXI GP0 rozhraní. Rozhraní typu GP (*General Purpose*) jsou na rozdíl od rozhraní typu HP (*High Performace*) určeny především pro nízkorychlostní a nízkoobjemové datové přenosy. Použité AXI HP/GP porty nejsou koherentní s lokálními paměťmi typu cache, a proto bylo nutné po každém zápisu do DRAM paměti provést manuální synchronizaci těchto dvou pamětí.

3.1.2 Data FIFO

Pro vyrovnávací FIFO paměť vzorků signálu byl použit IP blok s názvem AXI4-Stream Data FIFO. Vyrovnávací paměť slouží jednak pro přechod z hodinové domény clk IQ na hodinovou doménu clk H, a jednak pro ukládání vzorků signálu v okamžicích, kdy je prováděn výpočet a aktualizace fáze rozprostírací sekvence a frekvence komplexní exponenciály, a tudíž pozastaveno zpracování příchozích signálových vzorků.

Při volbě velikosti vyrovnávací paměti je potřeba vycházet z teoreticky nejdelší možné doby výpočtu a aktualizace zmíněných parametrů. Při určení této doby se vycházelo z nepravděpodobného, avšak teoreticky možného případu, kdy by v jednom okamžiku došlo k dokončení derozprostírání aktuálně přijímaného symbolu u všech aktivních kanálů, a tím pádem i k potřebě provést zmíněné operace pro všechny aktivní kanály před zahájením zpracování dalších příchozích vzorků signálu.

Minimální počet IQ vzorků signálu, které vyrovnávací paměť musí být schopna uložit, je možné určit vztahy 3.1 až 3.3 v závislosti na dostupných parametrech.

$$N_{IQ,min} = \frac{N_{ch,max}T_{ch}}{T_{IQ}}, \quad (3.1)$$

$$N_{IQ,min} = f_{IQ}N_{ch,max}(T_L n_{ch,L} + T_H n_{ch,H}), \quad (3.2)$$

$$N_{IQ,min} = f_{IQ}N_{ch,max} \left(\frac{n_{ch,L}}{f_L} + \frac{n_{ch,H}}{f_H} \right), \quad (3.3)$$

kde význam proměnných je následující:

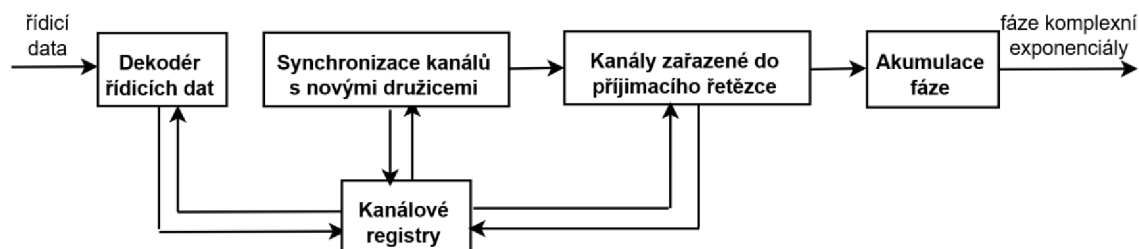
- $N_{IQ,min}$ je minimální počet IQ vzorků signálu,
- $N_{ch,max}$ je maximální počet paralelně zpracovávaných kanálů v daném přijímacím modulu,

- T_{ch} je doba výpočtu a aktualizace parametrů pro jeden kanál,
- T_{IQ} je vzorkovací perioda přijímaného signálu,
- f_{IQ} je vzorkovací frekvence přijímaného signálu, odpovídá taktovacímu kmitočtu clk IQ na obrázku 3.1,
- T_L je perioda taktovacího kmitočtu nízkofrekvenční části,
- f_L je frekvence taktovacího kmitočtu nízkofrekvenční části, odpovídá taktovacímu kmitočtu clk L na obrázku 3.1,
- T_H je perioda taktovacího kmitočtu vysokofrekvenční části,
- f_H je frekvence taktovacího kmitočtu vysokofrekvenční části, odpovídá taktovacímu kmitočtu clk H na obrázku 3.1,
- $n_{ch,L}$ je maximální počet period T_L potřebných pro provedení zmíněných operací v nízkofrekvenční části,
- $n_{ch,H}$ je maximální počet period T_H potřebných pro provedení zmíněných operací ve vysokofrekvenční části.

V praxi je vhodné vypočítanou hodnotu $N_{IQ,min}$ navýšit o přiměřenou hodnotu, aby bylo zaručeno, že nikdy nedojde k úplnému naplnění této paměti, přičemž při použití zmíněného IP bloku je potřeba zvolit hodnotu odpovídající mocnině 2. Vzhledem k tomu, že výsledky hardwarové implementace v této práci budou dále předmětem optimalizací, které budou mít vliv na výsledné hodnoty parametrů ve vztazích 3.1 až 3.3, nemá smysl tyto vztahy řešit pro konkrétní hodnoty těchto parametrů. Nicméně pro účely této práce byla pro signály obou zkoumaných navigačních systémů dostačující hodnota $N_{IQ,min} = 2048$. Každá dvojice IQ vzorků byla uložena v nižších bitech jednoho bajtu vyrovnávací paměti, z čehož plyne, že nastavená velikost vyrovnávací paměti v bajtech se rovnala hodnotě $N_{IQ,min}$.

3.1.3 Fázový akumulátor

Jak již název napovídá, hlavním úkolem této komponenty je generování fáze komplexní exponenciály pro frekvenční a fázovou synchronizaci. Zjednodušené blokové schéma komponenty je na obrázku 3.2.



Obr. 3.2 Zjednodušené blokové schéma fázového akumulátoru

Veškerá konfigurace fázového akumulátoru, s výjimkou případné korekce synchronnosti počátečních vzorků pro akvizici, je prováděna po sběrnici TDATA rozhraní S_AXIS CTRL. Jedná se o zavádění nových družic nalezených v procesu akvizice a naopak vyřazování družic se slabým signálem, u nichž došlo ke ztrátě synchronizace. U aktivních kanálů, kterým byla přiřazena dostupná družice, je prostřednictvím zmíněného rozhraní prováděna úprava fázového inkrementu a úprava fáze rozprostírací sekvence. Funkci řídicí jednotky, která tyto konfigurace provádí, plní v popisované hardwarové implementaci blok obsahující diskriminátory a smyčkové filtry, jak je možné vidět v blokovém schématu na obrázku 3.1. Vytvořený konfigurační protokol je rozebrán v popisu řídicí komponenty v kapitole 3.1.9.

Hodnoty fázového akumulátoru jsou odesílány po sběrnici TDATA rozhraní M_AXIS PHASE společně s doplňujícími informacemi obsaženými ve sběrnici TUSER a v signálu TLAST. Sběrnice TDATA má šířku 8 bitů, přičemž její minimální hodnota 0 odpovídá fázi 0 radiánů a maximální hodnota 255 odpovídá fázi 2π radiánů. Šířka sběrnice TDATA, a s ní i minimální fázový rozdíl, byl zvolen s ohledem na počet bitů, do kterých je kvantována reálná a imaginární složka komplexní exponenciály, a který činí 5 bitů. Vzhledem k nelineárnímu průběhu harmonických funkcí, kterými jsou složky komplexní exponenciály tvořeny, bylo vhodné šířku fáze zvolit o několik bitů větší. Fáze jednotlivých kanálů jsou postupně a periodicky vysílány, přičemž fáze posledního kanálu je indikována aktivní úrovní signálu TLAST. Doplňující informace jsou obsaženy ve sběrnici TUSER, jejíž položky jsou popsány v tabulce 3.1. Názvy položek jsou shodné s názvy použitými v příslušném HDL (*Hardware Description Language*) kódu.

Tabulka 3.1 Položky M_AXIS PHASE TUSER sběrnice fázového akumulátoru

Název položky	Počet bitů	Základní popis
sat_ID	5	Číslo družice.
chan_ID	4	Číslo kanálu.
ms_elapsed_flag ₁	1	Logická 1 indikuje, že byla odeslána kompletní perioda C/A kódu (1 milisekunda).
ms4_elapsed_flag ₂	1	Logická 1 indikuje, že byla odeslána kompletní perioda primárního kódu (4 milisekundy).
acq_sync_flag	1	Logická 1 indikuje, že údaje obsažené na sběrnících TDATA a TUSER přísluší k počátečnímu vzorku úseku signálu určeného pro akvizici.
dummy_flag	1	Logická 1 indikuje, že ve fázovém akumulátoru není aktivní žádný kanál a vysílané informace slouží pouze k příjmu navigačního signálu. Podrobnější informace v kapitole 3.1.5.

dump_DDS_flag	1	Logická 1 indikuje, že fázový akumulátor čeká na aktualizaci informací týkajících se fázového inkrementu a fáze rozprostírací sekvence, a že vysílané informace slouží pouze pro vyprázdnění DDS komponenty. Podrobnější informace v kapitole 3.1.4.
CA_phase ₁	12	Fáze C/A kódu (-1022...1022,5) vzorku signálu dané družice, ke kterému náleží aktuálně vysílané informace. 1 znaménkový bit, 10 bitů pro celočíselnou část, 1 bit pro zlomkovou část.
prim_phase ₂	17	Fáze primárního kódu (-4091...4091,9375) vzorku signálu dané družice, ke kterému náleží aktuálně vysílané informace. 1 znaménkový bit, 12 bitů pro celočíselnou část, 4 bity pro zlomkovou část.
sec_phase ₂	5	Fáze sekundárního kódu (0...24) vzorku signálu dané družice, ke kterému náleží aktuálně vysílané informace.

₁ údaj platný pouze pro signál L1 C/A

₂ údaj platný pouze pro signál E1-OS

Položky sběrnice TUSER popsané v tabulce 3.1 slouží pro správnou funkci následujících funkčních bloků v přijímacím řetězci a významy většiny z nich budou objasněny při popisu těchto bloků v následujících kapitolách.

Fázový akumulátor potřebuje při zavádění nově nalezených družic znát počátky signálových úseků použitých pro akvizici, aby mohl určit okamžik, kdy začít vysílat fázi nově nalezené družice tak, aby akvizicí získaná fáze rozprostírací sekvence odpovídala aktuální fázi této sekvence v přijímaném signálu. V případě popisované hardwarové implementace, kdy byly použity soubory s relativně krátkými úseky signálu, a kdy byly pro počáteční vzorky těchto signálů známy výsledky akvizice, která byla provedena v MATLABu při simulacích v kapitole 2, stačilo interní čítač vzorků pro akvizici na začátku příjmu resetovat na nulu. V případě příjmu v reálném čase, který může trvat mnohem delší dobu, je potřeba akvizici provádět průběžně a v tomto případě bude potřeba k signálovým vzorkům periodicky vkládat značky indikující počátky úseků signálu určených pro akvizici. Kdyby komponenta, která bude v budoucnu vytvořena pro vkládání těchto značek, fungovala zcela bezchybně a značky vkládala nepřetržitě po celou dobu příjmu, teoreticky by u fázového akumulátoru stačil pouze počáteční reset čítače na nulu a jeho obnova s definovanou periodou.

Pro zajištění synchronnosti počátečních vzorků akvizice za všech podmínek bylo pro budoucí použití vytvořeno rozhraní S_AXIS ACQ SYNC určené pro případnou korekci okamžiků těchto počátečních vzorků. Informace o rozdílu bude vyhodnocována na základě rozdílu poloh akvizičních značek signálu a značek, které jsou periodicky vkládány fázovým akumulátorem na sběrnici TUSER v podobě položky nazvané acq_sync_flag (viz. tabulka 3.1). Rozdíl přijatý na sběrnici TDATA rozhraní S_AXIS

ACQ SYNC je odečten od aktuální hodnoty interního čítače značek pro akvizici. Důvodem, proč pro tuto korekci bylo vytvořeno zvláštní rozhraní, je skutečnost, že může nastat taková situace, že korekci bude potřeba provést v okamžiku, kdy nebude aktivní žádný kanál družice, což je případ, kdy hlavní konfigurační rozhraní S_AXIS CTRL není aktivní.

Chod jednotlivých kanálů uvnitř fázového akumulátoru je řízen kanálovými registry, jež jsou vypsané v tabulce 3.2, přičemž jejich názvy jsou shodné s názvy použitými v příslušném HDL kódu. Významy registrů budou postupně blíže vysvětleny v textu následujícím za uvedenou tabulkou.

Tabulka 3.2 Kanálové registry fázového akumulátoru

Název registru	Počet bitů	Základní popis
enabled	1	Logická 1 indikuje, že kanál je aktivní.
sat_ID	5	Číslo družice.
acq_sync_state	výčtový typ, 5 položek	Stav synchronizace nově nalezené družice – NA (<i>Not Available</i>), acquired, sync_in_progress, synchronized, in_channel_ring.
ms_elapsed ₁	1	Logická 1 indikuje, že byla odeslána kompletní perioda C/A kódu (1 milisekunda).
ms4_elapsed ₂	1	Logická 1 indikuje, že byla odeslána kompletní perioda primárního kódu (4 milisekundy).
phase_increment	28 ₁ /30 ₂	Fázový inkrement fázového akumulátoru.
phase_accumulator	28 ₁ /30 ₂	Fázový akumulátor.
CA_phase ₁	75	Fáze C/A kódu (-1022 až 1022,999...). 1 znaménkový bit, 10 bitů pro celočíselnou část, 64 bitů pro zlomkovou část.
prim_phase ₂	77	Fáze primárního kódu (-4091 až 4091,999...). 1 znaménkový bit, 12 bitů pro celočíselnou část, 64 bitů pro zlomkovou část.
sec_phase ₂	5	Fáze sekundárního kódu (0...24)
channel_ring_prev	4	Číslo kanálu, jehož fáze byla vysílána před aktuálně vysílaným kanálem.
channel_ring_next	4	Číslo kanálu, jehož fáze bude vysílána po aktuálně vysílaném kanálu.

₁ údaj platný pouze pro signál L1 C/A

₂ údaj platný pouze pro signál E1-OS

Po zapnutí přijímače není před provedením akvizice aktivní žádný kanál, což je ve všech kanálech indikováno logickou 0 zapsanou v registru enabled a hodnotou NA zapsanou v registru acq_sync_state. V tomto okamžiku fázový akumulátor periodicky vysílá na rozhraní M_AXIS PHASE pouze prázdné pakety s aktivní dummy_flag

položkou. Jakmile je akvizicí nalezena nějaká družice, příslušná řídicí jednotka pošle přes rozhraní S_AXIS CTRL příkaz k zavedení nové družice do přijímacího řetězce, společně s potřebnými informacemi o dané družici (číslo kanálu, číslo družice, přibližný frekvenční posuv a fáze rozprostírací sekvence). Fázový akumulátor tyto informace uloží do příslušných kanálových registrů daného kanálu, registr enabled nastaví do logické 1 a do registru stavu synchronizace, acq_sync_state, nastaví hodnotu acquired. Nyní fázový akumulátor dále posílá prázdné pakety a čeká na okamžik počátečního vzorku pro akvizici. Jakmile nastane tento okamžik, fázový akumulátor ví, že v aktuálně přijímaném vzorku signálu má daná družice fázi rozprostírací sekvence shodnou s fází obdrženu z procesu akvizice, změní stav synchronizace na hodnotu sync_in_progress a s každým příchozím vzorkem signálu inkrementuje fázi rozprostírací sekvence o odpovídající hodnotu, která vychází z chipové rychlosti této sekvence a vzorkovací frekvence signálu. V okamžiku, kdy hodnota fáze rozprostírací sekvence dosáhne své maximální hodnoty, je stav synchronizace změněn na hodnotu synchronized. V následujícím přijatém vzorku signálu se nachází počátek symbolu a v tomto okamžiku dojde k přepnutí do konečného stavu synchronizace – stavu nazvaného in_channel_ring, což je stav, kdy je daná družice zařazena do periodicky vysílaného řetězce fází kompletně synchronizovaných družic. Když dojde u dané družice k vyslání fází odpovídajících kompletní periodě rozprostírací sekvence, je registr ms_elapsed/ms4_elapsed nastaven do aktivní úrovně, komponenta začne vysílat prázdné pakety s aktivní položkou dump_DDS_flag a čeká na aktualizaci fázového inkrementu a fáze rozprostírací sekvence z řídicí jednotky. Po aktualizaci těchto parametrů komponenta dále pokračuje v inkrementaci a posílání fází do přijímacího řetězce.

V tabulce 3.2 je možné si všimnout poněkud velkého počtu bitů určeného pro desetinnou část rozprostíracích sekvencí. Důvodem jejich použití jsou vzorkovací frekvence navigačních signálů, které neodpovídají celočíselnému násobku chipové rychlosti rozprostíracích sekvencí. Násobek musí být dále roven mocnině čísla 2, jinak má změna fáze rozprostíracích sekvencí, odpovídající jedné vzorkovací periodě, nekonečnou zlomkovou část ve dvojkové soustavě, kvůli čemuž docházelo na pomezí jednotlivých chipů k jejich chybnému adresování a výrazně zhoršenému příjmu družic se slabším signálem. Největší náchylnost na tuto nepřesnost vykazovala DLL smyčka, jejíž early/late sekvence mají sami o sobě již sníženou korelaci s příchozí sekvencí. Při budoucím použití komponenty s možností nastavení vzorkovací frekvence bude šířka této zlomkové části výrazně menší.

Bitová hloubka registrů, ve kterých je uložen fázový inkrement a fázový akumulátor byla zvolena tak, aby výslednou přesností, se kterou je možné nastavit frekvenci komplexní exponenciály, nebyla snížena přesnost synchronizačních smyček. Smyčky fázové synchronizace jsou u kvalitních signálů schopny dosáhnout přesnosti pohybující se v řádech desetin Hertzů, a proto byl pro fázový akumulátor zvolen požadavek dosáhnout přesnosti v řádu setin Hertzů, čemuž odpovídá šířka 28 bitů pro signál

L1 C/A a šířka 30 bitů pro signál E1-OS. Bitové šířky byly vypočítány pomocí vztahu 3.4 [23].

$$B = \log_2 \frac{f_s}{\Delta f}, \quad (3.4)$$

kde význam proměnných je následující:

- B je šířka fázového akumulátoru v bitech,
- f_s je vzorkovací frekvence signálu v Hertzích,
- Δf je výsledná přesnost v Hertzích, se kterou lze frekvenci nastavit.

Řídicí jednotka může fázovému akumulátoru kdykoliv poslat příkaz na okamžité vyřazení jakéhokoliv kanálu z přijímacího řetězce. Informace o pořadí vysílaných kanálů je uložena v registrech `channel_ring_prev` a `channel_ring_next`, jejichž hodnoty jsou u vyřazeného kanálu odpovídajícím způsobem přesunuty do registrů sousedních vysílaných kanálů. Do registru `channel_ring_next` kanálu, jehož číslo odpovídá hodnotě registru `channel_ring_prev` vyřazovaného kanálu, je zapsána hodnota registru `channel_ring_next` vyřazovaného kanálu. Do registru `channel_ring_prev` kanálu, jehož číslo odpovídá hodnotě registru `channel_ring_next` vyřazovaného kanálu, je zapsána hodnota registru `channel_ring_prev` vyřazovaného kanálu.

Fázový akumulátor má v sobě zabudovaný FIFO zásobník pro příkazy z řídicí jednotky o hloubce 16, což je hodnota odpovídající maximálnímu možnému počtu aktivních kanálů. Díky tomuto zásobníku nemusí řídicí jednotka čekat na postupné zpracovávání zaslaných příkazů.

3.1.4 DDS

Pro generování komplexní exponenciály byl použit IP blok s názvem DDS Compiler. IP blok byl nastaven na generování reálné a imaginární složky komplexní exponenciály ve dvojkovém doplňku s bitovou hloubkou 5 bitů. Jako vstupní parametr byla nastavena fáze s bitovou hloubkou 8 bitů. Dále byla komponenta nastavena tak, aby společně se vstupní fází přenášela se stejnou latencí i signály TUSER a TLAST na výstup.

Bitová hloubka složek komplexní exponenciály byla zvolena tak, aby kvantizační šum, který je v ní obsažen, měl minimální vliv na celkový šum obsažený ve výsledném signálu po frekvenční/fázové kompenzaci. Výsledná hodnota byla zvolena na základě simulací v MATLABu, kde byl počítán poměr signál-šum dvou signálů. Prvním z nich byl harmonický signál kvantovaný do jednoho bitu. Druhým signálem byl signál, který vznikl vynásobením zmíněného jednobitového harmonického signálu druhým harmonickým signálem, jehož bitová hloubka byla postupně zvyšována, dokud mezi oběma poměry signál-šum nebylo dosaženo zanedbatelného rozdílu.

Při odeslání poslední fáze z fázového akumulátoru před aktualizací parametrů bylo zjištěno, že tyto poslední hodnoty fáze již neprojdou až na výstup, dokud se na vstupu DDS neobjeví další hodnoty, k čemuž však již nedorazí, protože následující funkční

bloky potřebují kompletní periodu rozprostírací sekvence s položkou `ms_elapsed_flag/ms4_elapsed_flag` v aktivní úrovni u posledního vzorku periody k tomu, aby mohlo dojít k výpočtu a aktualizaci nových hodnot fázového inkrementu a fáze rozprostírací sekvence. Kvůli této vlastnosti DDS IP byla vytvořena již zmíněná položka `dump_DDS_flag` (viz. tabulka 3.1), kterou fázový akumulátor nastaví do aktivní úrovně a dále posílá prázdné pakety i v době zastaveného zpracování vstupních vzorků. Následující funkční bloky automaticky ignorují všechny pakety, které mají tuto položku v aktivní úrovni.

3.1.5 IQ & DDS synchronizátor

V přijímacím řetězci se může objevovat s časem se měnící počet aktivních kanálů, což znamená, že se s časem může měnit i počet po sobě jdoucích hodnot komplexní exponenciály, které musí být vynásobeny s jedním příchozím vzorkem navigačního signálu. K této synchronizaci signálových vzorků a vzorků komplexní exponenciály slouží komponenta s názvem IQ & DDS synchronizátor.

Komponenta provede duplikaci každého signálového vzorku přečteného ze vstupního rozhraní `S_AXIS IQ` na výstupní rozhraní `M_AXIS IQ` v počtu odpovídajícím počtu kanálů, které jsou aktuálně periodicky vysílány fázovým akumulátorem. Počet aktivních kanálů je vyhodnocován pomocí signálu `TLAST`, kterým je označen poslední kanál v periodicky vysílaném řetězci kompletně synchronizovaných kanálů. Vzorky komplexní exponenciály jsou převáděny ze vstupního rozhraní `S_AXIS DDS` na výstupní rozhraní `M_AXIS DDS` s výjimkou vzorků s aktivní `TUSER` položkou `dump_DDS_flag`. Obsah signálů `TUSER` a `TLAST` je rovněž převáděn na výstupní rozhraní pro jejich další využití následujícími funkčními bloky.

Pro případy, kdy není v přijímači aktivní žádný kanál, a tudíž nejsou do přijímacího řetězce vysílány žádné platné hodnoty fázového akumulátoru, potažmo komplexní exponenciály, bylo potřeba zajistit, aby synchronizátor nečekal na příchozí vzorky komplexní exponenciály, a tudíž neblokoval příjem vzorků signálu, které jsou právě potřeba pro provedení akvizice a synchronizaci nalezených kanálů. Tento problém byl vyřešen vytvořením `TUSER` položky `dummy_flag` (viz. tabulka 3.1), kterou v popisované situaci fázový akumulátor nastaví do aktivní úrovně a posílá prázdné pakety společně s aktivním `TLAST` signálem, což vede k tomu, že s každým takto odeslaným prázdným paketem je synchronizátorem přijat vzorek navigačního signálu, ovšem s informací, že tento vzorek není určen pro zpracování synchronizačními smyčkami.

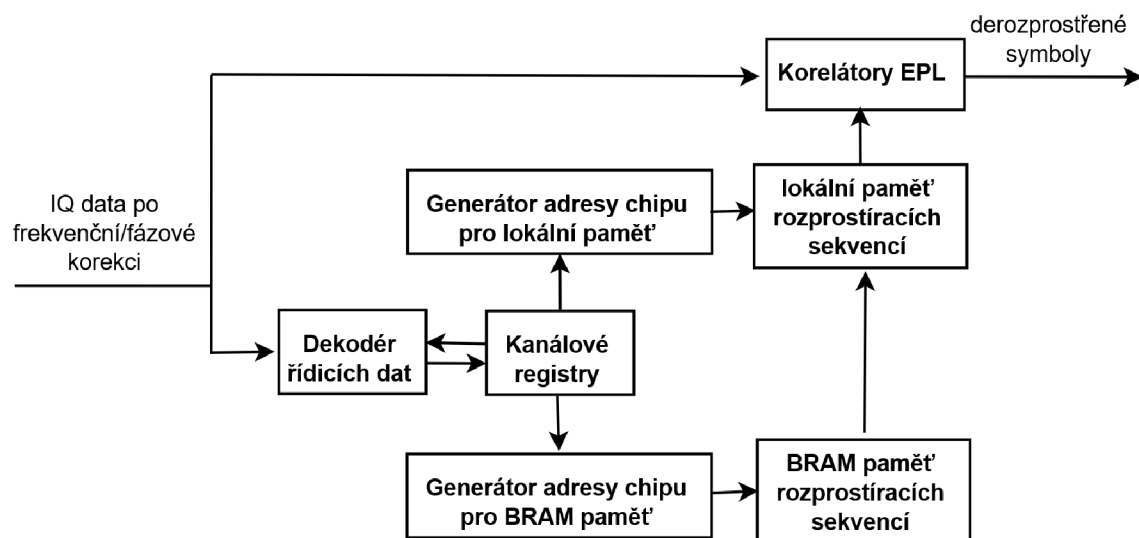
3.1.6 Komplexní násobička

Vzorky na výstupních rozhraních IQ & DDS synchronizátoru je potřeba mezi sebou vynásobit k provedení frekvenční/fázové kompenzace. Jelikož se jedná o operaci násobení dvou komplexních čísel, bylo pro tento účel použito již hotové IP jádro

s názvem Complex Multiplier. Vstup i výstup komponenty je tvořen AXIS rozhraními. Komponenta je nastavena tak, aby přeposílala signály TUSER a TLAST se stejnou latencí, s jakou provádí operaci násobení. K násobení jsou použity vestavěné násobičky v podobě DSP48 bloků.

3.1.7 EPL korelátor

Jak název napovídá, tento funkční blok má na starosti derozprostírání přijímaného signálu, a to pro všechny 3 vzájemně posunuté rozprostírací sekvence (Early, Prompt, Late). Zjednodušené blokové schéma komponenty je na obrázku 3.3.



Obr. 3.3 Zjednodušené blokové schéma EPL korelátoru

Rozprostírací sekvence jsou uloženy v BRAM (*Block Random Access Memory*) paměti. BRAM paměť byla vytvořena vložením IP jádra s názvem Block Memory Generator, pomocí kterého byla nakonfigurována jako ROM (*Read Only Memory*) paměť s délkou slova 1 bit a délkou adresové sběrnice 15 bitů pro signál L1 C/A a 17 bitů pro signál E1-OS. Na každé adrese paměti se nachází právě 1 chip rozprostírací sekvence. Horních 5 bitů adresy slouží pro výběr jedné z 32 družic a zbývající bity pro výběr konkrétního chipu. V případě signálu E1-OS je v paměti uložena pouze primární sekvence. Sekundární sekvence signálu E1-OS je uložena v distribuované paměti FPGA jako konstanta o délce 25 bitů. Číslo družice a fáze rozprostírací sekvence náležící k přijatému vzorku je komponentou přečtena z příslušných TUSER položek vstupního S_AXIS IQ rozhraní. Z tabulky 3.1 je patrné, že fáze primární sekvence přenášená po TUSER sběrnici má pro desetinnou část vyhrazeno více bitů, než je tomu u C/A sekvence. Důvodem je menší vzájemné posunutí EPL sekvencí u signálu E1-OS, což vyžaduje i vyšší přesnost při určení fází těchto sekvencí.

Derozprostírání je v jednom okamžiku prováděno pro všechny 3 sekvence, jejichž adresy aktuálních chipů se vzájemně mohou lišit, přičemž chip E sekvence může být

pro následující příchozí vzorky použit u zbývajících dvou sekvencí, které jsou vzhledem k sekvenci E časově zpožděny. Aby se předešlo opakovanému čtení BRAM paměti, která má v dané konfiguraci latenci 2 hodinových cyklů, ze stejné adresy, byla v komponentě vytvořena pro všechny kanály malá lokální paměť rozprostíracích sekvencí o délce 8 bitů. Tato lokální paměť upravuje čtením hlavní BRAM paměti svůj obsah v závislosti na aktuální adrese sekvence P a umožňuje okamžité čtení všech 3 sekvencí pro daný kanál.

Příchozí vzorky s TUSER položkou `dummy_flag` v aktivní stavu jsou komponentou automaticky zahazovány.

V případě signálu E1-OS je prováděno také odstranění BOC subnosné, a to násobením lokálně generovanou subnosnou, jejíž fáze je určena desetinnou částí fáze rozprostírací sekvence. Jak již bylo dříve uvedeno, v době vzniku této práce je složka BOC(6, 1) zanedbána.

Jakmile dojde k ukončení derozprostírání aktuálně přijímaného symbolu, což je indikováno TUSER položkou `ms_elapsed_flag/ms4_elapsed_flag` v aktivní úrovni, jsou hodnoty EPL akumulátorů odeslány přes výstupní rozhraní M_AXIS EPL do bloku tvořeného diskriminátory a smyčkovým filtrem. EPL akumulátory jsou následně resetovány na nulovou hodnotu. Hodnoty EPL akumulátorů, z nichž je každý tvořen reálnou a imaginární částí, jsou společně s informací o číslu družice a číslu kanálu posílány po sběrnici TDATA, sběrnice TUSER není u rozhraní M_AXIS_EPL použita.

Šířka jedné části (reálná nebo imaginární) každého akumulátoru je u signálu L1 C/A 18 bitů a u signálu E1-OS 22 bitů. Při určení těchto šířek se vycházelo z teoretické úvahy, že by s každým příchozím vzorkem byla k akumulátoru přičtena maximální možná hodnota, která se může na výstupu komplexní násobičky objevit, přičemž by znaménko této hodnoty bylo po derozprostření daným chipem vždy stejné. Maximální, případně minimální, hodnota složek komplexní exponenciály je při kvantování do n bitů $\pm(2^{n-1} - 2)$, z čehož pro $n = 5$ plyne hodnota ± 14 [23]. Maximální hodnoty jedné ze složek výsledku komplexního násobení může být dosaženo v případě, že absolutní hodnoty složek komplexní exponenciály jsou shodné a IQ hodnota příchozího vzorku navigačního signálu je taková, že při násobení nedojde v rámci jedné ze složek k součtu hodnot s rozdílnými znaménky. Absolutní hodnoty složek komplexní exponenciály jsou shodné například pro její fázi rovnající se hodnotě 45° , kdy jsou obě složky rovny hodnotě $(2^{5-1} - 2) \cdot \frac{\sqrt{2}}{2} \doteq 10$. Jestliže je příchozí vzorek navigačního signálu například u signálu L1 C/A roven hodnotě $1 + 1j$, je na výstupu komplexní násobičky hodnota $20j$. V případě budoucího nastavení GNSS přijímače MAX2771 na jednobitové kvantování signálu L1 C/A a dvoubitové kvantování signálu E1-OS, je maximální možnou hodnotou jedné ze složek komplexního násobení hodnota 20 pro signál L1 C/A a hodnota 40 pro signál E1-OS. Celková maximální možná hodnota v akumulátoru je dále závislá na vzorkovací frekvenci, která určuje počet hodnot, které se k akumulátoru přičtou během jedné periody rozprostírací sekvence. U signálu L1 C/A se do budoucna jeví jako vhodná hodnota 4,092 MHz, což je první hodnota vzorkovací frekvence, která

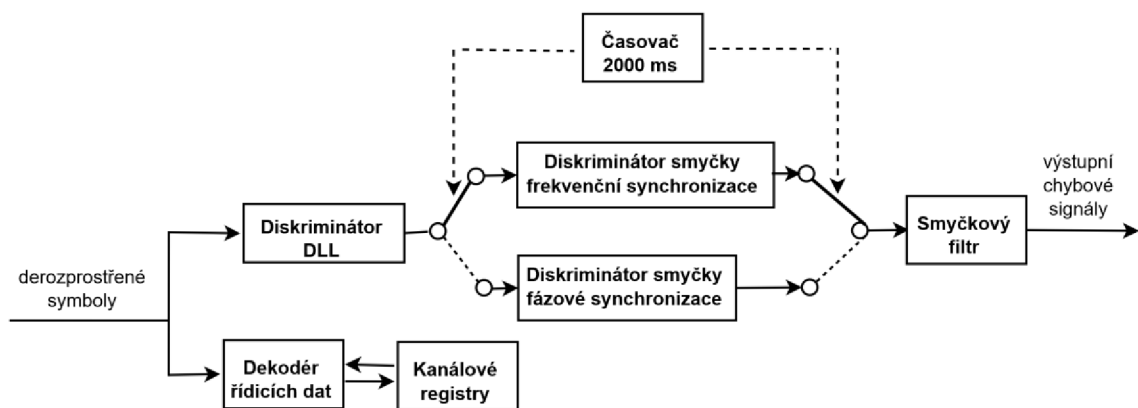
je vyšší než současně používaná vzorkovací frekvence, a která splňuje podmínky pro předejití problémům s nekonečnou zlomkovou částí inkrementu fáze rozprostírací sekvence, jak je uvedeno v kapitole 3.1.3. U signálu E1-OS se ze stejných důvodů dá do budoucna uvažovat o hodnotě 12,276 MHz, která navíc umožní použití složky BOC(6, 1). Maximální, případně minimální, hodnotou, kterou tím pádem musí být každý akumulátor schopen uchovat je u signálu L1 C/A $\pm 20 \cdot 0,001s \cdot 4,092MHz = \pm 81840$ a u signálu E1-OS $\pm 40 \cdot 0,004s \cdot 12,276MHz = \pm 1964160$, což jsou hodnoty, pro jejichž uložení ve dvojkovém doplňku jsou potřeba bitové šířky uvedené na začátku tohoto odstavce. Časové konstanty 0,001s a 0,004s jsou periody rozprostíracích sekvencí.

3.1.8 Změna hodinové domény

Jak již bylo dříve zmíněno, komponenta tvořena diskriminátory a smyčkovým filtrem je taktována hodinovým signálem clk L, na rozdíl od ostatních komponent, které jsou taktovány hodinovým signálem clk H. Proto byla potřeba daná AXIS rozhraní, která jsou taktována rozdílnými hodinovými signály, mezi sebou synchronizovat, k čemuž bylo použito IP jádro s názvem AXI4-Stream Clock Converter. IP jádro není třeba nějak konfigurovat s výjimkou počtu synchronizačních stupňů, který byl nastaven na hodnotu 2, což je v praxi běžně používaný počet.

3.1.9 Diskriminátory a smyčkový filtr

Posledním funkčním blokem v přijímacím řetězci je blok obsahující diskriminátor smyčky DLL, diskriminátory smyček frekvenční a fázové synchronizace a smyčkové filtry. Popisovaná komponenta rovněž provádí násobení konstantami K_0 a K_d popisovanými v kapitole 2 a v případě signálu L1 C/A provádí i bitovou synchronizaci. Na začátku příjmu jsou komponentou inicializovány kanály s dostupnými družicemi na základě informací získaných z akvizice provedené v MATLABu. Zjednodušené blokové schéma komponenty je na obrázku 3.4.



Obr. 3.4 Zjednodušené blokové schéma bloku diskriminátorů a smyčkového filtru

Komponenta obsahuje jednu děličku, která se používá pro normalizaci chybových signálů, a která je společná pro všechny diskriminátory. Vzhledem k poměrně velkému množství hardwarových prostředků, které jsou k její realizaci potřeba a vzhledem k její poměrně nízké vytíženosti by nebylo výhodné ji instanciovat pro každý diskriminátor zvlášť. Pro vytvoření děličky bylo použito IP jádro s názvem Divider Generator. Pro bitovou šířku vstupních operandů byla zvolena hodnota 36, což je automaticky i bitová šířka výstupního podílu. Pro zlomkovou část byla zvolena šířka 16 bitů. Zvolené bitové šířky byly kompromisem mezi hardwarovou náročností komponenty, její latencí a výsledné přesnosti. Zvoleným typem algoritmu při instanciaci děličky byl algoritmus s názvem High Radix, což byl, vzhledem ke zvoleným bitovým šířkám, doporučený typ algoritmu, který navíc využívá z velké části DSP48 bloky, kterých je na FPGA dostatek.

Při přijetí nových hodnot EPL akumulátorů přes rozhraní S_AXIS EPL je nejprve proveden výpočet chybového signálu pro korekci fáze rozprostírací sekvence, následně je proveden výpočet chybového signálu pro frekvenční, případně fázovou synchronizaci, a nakonec jsou všechny chybové signály společně s číslem kanálu a číslem družice odeslány přes rozhraní S_AXIS CTRL do fázového akumulátoru. Bitová synchronizace u signálu L1 C/A je prováděna paralelně s uvedenými operacemi, protože jako jediná z operací nepotřebuje k výpočtu sdílenou děličku.

Hardwarová implementace DLL diskriminátoru odpovídá vztahu 2.5. Konstanta úměrnosti byla u signálu L1 C/A změněna z hodnoty $\frac{1}{40}$, popisované v kapitole 2.2, na hodnotu $\frac{1}{32}$, aby ji bylo možné realizovat jednoduchým bitovým posuvem. Konstantu úměrnosti $\frac{13}{128}$ u signálu E1-OS se během hardwarové implementace osvědčilo změnit na hodnotu $\frac{1}{16}$. Pro výpočty druhých mocnin a operace sčítání a odečítání byly použity násobičky a sčítačky, jež jsou součástí DSP48 bloků. U signálu E1-OS, jehož akumulátory mají šířku 22 bitů, bylo potřeba po výpočtu druhé mocniny zanedbat nejnižší bity z důvodu příliš velké výsledné bitové šířky vzhledem k nakonfigurované děličce, použité pro normování chybového signálu. U družic se slabým signálem, jejichž hodnoty akumulátorů ani zdaleka neodpovídají maximální hodnotě, by ovšem prosté oříznutí nejnižších bitů znamenalo vnesení poměrně velké chyby, a proto byl u všech 4 akumulátorů (reálná a imaginární část akumulátorů E a L) před oříznutím bitů proveden, pokud možno, co největší bitový posuv vlevo tak, aby vzájemné poměry mezi jednotlivými akumulátory zůstaly zachovány beze změny a zároveň se eliminoval vliv oříznutí nejnižších bitů.

Pro frekvenční synchronizaci u signálu L1 C/A byl na základě výsledků dosažených v kapitole 2 zvolen ML estimátor s parametry shodnými s těmi, které byly použity při simulacích v MATLABu. Při výpočtu chybového signálu byly opět použity především bloky DSP48. ML estimátor v dané konfiguraci potřebuje znát 7 posledních IQ hodnot akumulátoru P, což při použití všech 16 kanálů, představuje téměř 5000 bitů, a proto pro uložení těchto hodnot byla použita dvouportová BRAM paměť, kdy jeden port slouží pro čtení reálné složky a druhý port pro čtení složky imaginární, díky čemuž čtení

vzorků zabere o polovinu méně času. Stejně jako u DLL diskriminátoru, i zde je před normováním provedeno zesílení hodnot pomocí bitových posuvů vlevo.

Frekvenční synchronizace u signálu E1-OS je prováděna pomocí kvadrikorelátoru, který u tohoto signálu vykazoval při simulacích v MATLABu nejlepší vlastnosti. Parametry smyčky byly zvoleny shodně s parametry v MATLABu. Jak lze vidět ve vztahu 2.24, kvadrikorelátor potřebuje pro výpočet derivace reálné složky znát hodnotu předchozího vzorku. Jelikož se jedná pouze o jednu 22-bitovou hodnotu, je tato hodnota uložena v jednom z kanálových registrů. Při výpočtu byla použita jedna násobička tvořena bloky DSP48. Před normováním byly opět použity bitové posuvy vlevo.

Fázová synchronizace je automaticky zapnuta po uplynutí 2000 milisekund počátečního příjmu daného kanálu, hodnota doby příjmu je uložena v kanálovém registru s názvem `freq_tracking_time_ms`. Po uplynutí této doby je přepnut kanálový registr s názvem `freq_locked` do aktivní úrovně, což signalizuje, že počínaje tímto okamžikem má být pro daný kanál místo smyčky frekvenční synchronizace použita smyčka fázové synchronizace. Hodnota akumulátoru ve smyčkovém filtru, která je uložena v kanálovém registru s názvem `LF_acc`, není po přepnutí resetována, a tudíž slouží jako počáteční hodnota frekvenčního posuvu pro fázovou synchronizaci. Pro fázovou synchronizaci je u signálu L1 C/A použita Costasova smyčka a u signálu E1-OS smyčka PLL. Hardwarová implementace obou zmíněných smyček odpovídá vztahům 2.12 a 2.21, které je definují, a ze kterých lze vidět, že diskriminátory obou smyček jsou prakticky shodné s výjimkou násobení polaritou reálné složky ($\text{sgn}(I)$), která je u PLL smyčky vynechána. I v tomto případě byly pro výpočet chybového signálu použity bloky DSP48 a před normováním bitové posuvy vlevo. U signálu L1 C/A je doba integrace zpočátku rovna 1 milisekundě a po provedení bitové synchronizace je prodloužena na 20 milisekund, což znamená, že aktualizace frekvenčního posuvu je prováděna jedenkrát během 20 přichozích vzorků. U signálu E1-OS je díky známé sekundární sekvenci hned zpočátku nastavena doba integrace na 20 milisekund, z čehož plyne, že aktualizace frekvenčního posuvu je prováděna jednou během 5 přichozích vzorků. Přichozí vzorky P akumulátorů jsou proto dále akumulovány do kanálových registrů s názvy `P_I_20ms_acc` a `P_Q_20ms_acc`.

Pro urychlení výpočtů bylo násobení koeficienty K_0, K_d a koeficienty smyčkového filtru, a, b , sloučeno do násobení dvěma koeficienty K_a, K_b , které představují součin koeficientů K_0, K_d , příslušného koeficientu smyčkového filtru a koeficientu pro převod úhlového kmitočtu na fázový inkrement pro DDS, K_{DDS} . Výpočet koeficientu K_{DDS} vychází ze vztahu 3.5 určeného pro výpočet fázového inkrementu [23].

$$\Delta\theta = \frac{f \cdot 2^B}{f_s}, \quad (3.5)$$

kde význam jednotlivých proměnných je následující:

- $\Delta\theta$ je fázový inkrement,
- f je požadovaná frekvence,

- f_s je vzorkovací frekvence,
- B je šířka fázového akumulátoru v bitech.

Chybový signál na výstupu diskriminátoru je po vynásobení koeficienty K_0, K_d v radiánech za sekundu, což znamená, že po vydělení chybového signálu hodnotou 2π může být výsledná hodnota dosazena za proměnnou f ve vztahu 3.5. Tím pádem pro koeficienty $K_{DDS,L1}$ a $K_{DDS,E1}$ signálů L1 C/A a E1-OS plynou vztahy $K_{DDS,L1} = \frac{2^{B_{L1}}}{2\pi \cdot f_{s,L1}} = \frac{2^{28}}{2\pi \cdot 2500000 \text{ Hz}} \doteq \frac{107,37}{2\pi} \text{ s}$ a $K_{DDS,E1} = \frac{2^{B_{E1}}}{2\pi \cdot f_{s,E1}} = \frac{2^{30}}{2\pi \cdot 10000000 \text{ Hz}} \doteq \frac{107,37}{2\pi} \text{ s}$, kde $f_{s,L1}$ a $f_{s,E1}$ jsou vzorkovací frekvence navigačních signálů a B_{L1} a B_{E1} jsou bitové šířky fázových akumulátorů. Vzhledem k tomu, že v tomto případě má koeficient stejnou hodnotu pro oba signály, bylo možné zavést obecné označení K_{DDS} , pro které platí $K_{DDS} = K_{DDS,L1} = K_{DDS,E1}$.

Pro koeficienty K_a, K_b platí $K_a = a \cdot K_0 K_d K_{DDS}$ a $K_b = b \cdot K_0 K_d K_{DDS}$.

Implementace bitové synchronizace je shodná s implementací popisovanou v kapitole 2.5. Pro každý kanál bylo vytvořeno 20 čítačů, z nichž se každý inkrementuje v případě, že na jemu odpovídající pozici dojde ke změně polarit derozprostředěného signálu. Jakmile je dosaženo bitové synchronizace, aktivuje se kanálový registr s názvem `bit_locked`, který indikuje, že daný kanál je bitově synchronní s pozicí bitového přechodu definovanou kanálovým registrem s názvem `bit_end_position`.

Protokol, podle kterého popisovaná komponenta ovládá fázový akumulátor a s ním i zbytek přijímacího řetězce, je popsán v tabulce 3.3.

Tabulka 3.3 Položky sběrnice M_AXIS CTRL TDATA

Název položky	Počet bitů	Základní popis
sat_ID	5	Číslo družice.
chan_ID	4	Číslo kanálu.
phase_increment	21	Fázový inkrement.
CA_phase ₁	27	Fáze C/A kódu, případně jeho korekce. 1 bit pro znaménko, 10 bitů pro celočíselnou část, 16 bitů pro zlomkovou část.
prim_phase ₂	29	Fáze primárního kódu, případně jeho korekce. 1 bit pro znaménko, 12 bitů pro celočíselnou část, 16 bitů pro zlomkovou část.
sec_phase ₂	5	Fáze sekundárního kódu
enabled_flag	1	Logická 1 indikuje, že kanál je povolen.
new_flag	1	Logická 1 indikuje, že se jedná o nově zaváděnou družici.

₁ údaj platný pouze pro signál L1 C/A

₂ údaj platný pouze pro signál E1-OS

Jak popis tabulky 3.3 napovídá, řídicí informace jsou u rozhraní M_AXIS CTRL přenášeny po sběrnici TDATA. Jestliže fázový akumulátor přijme řídicí paket s položkou `enabled_flag` v logické 0, znamená to, že má bez ohledu na zbývající informace daný kanál vypnout. V opačném případě přichází na řadu vyhodnocení položky `new_flag`, která určuje, zda se jedná o zavádění nově nalezené družice, nebo o aktualizaci fázového inkrementu a fáze rozprostírací sekvence u již zavedené družice. V závislosti na položce `new_flag` určují položky týkající se rozprostíracích sekvencí buď počáteční fázi získanou z akvizice, nebo korekci z DLL smyčky. V případě korekce fáze rozprostírací sekvence mají význam pouze položky `CA_phase` a `prim_phase`, přičemž tyto položky mohou v tomto případě nabývat pouze hodnot v maximálním rozsahu ± 1 . Bitová šířka zlomkové části rozprostíracích sekvencí je dána bitovou šířkou zlomkové části použité děličky. Položka `phase_increment` může přenášet hodnoty fázového inkrementu v rozsahu odpovídajícímu frekvenčnímu posuvu v intervalu $\pm 10 \text{ kHz}$, a k tomu je pro formát dvojkového doplňku potřeba 21 bitů, jak je možné určit úpravou vztahu 3.5.

3.2 Maximální počet kanálů

I přes to, že pole nesoucí informaci o čísle kanálu mají šířku 4 bitů, nemusí skutečný maximální možný počet aktivních, paralelně běžících, kanálů dosahovat hodnoty 16. Důvodem může být nedostatečně vysoká taktovací frekvence a naopak příliš vysoká vzorkovací frekvence vstupního signálu. Zvolená kanálová kapacita, které by bylo dobré dosáhnout, vychází z pokusu, kdy se za příznivých podmínek podařilo na volném vyvýšeném prostranství nalézt pomocí softwarového rádia celkem 12 družic pro signál L1 C/A. Při zvolené kanálové kapacitě by tedy neměla nastat situace, kdy by nebylo možné zařadit do přijímacího řetězce nově nalezenou družici i v případě, kdy by některé kanály byly ještě zaneprázdněny družicemi, jejichž signály zeslábly natolik, že u nich došlo ke ztrátě synchronizace, ale ještě nebyly vyřazeny z přijímacího řetězce.

Souvislost mezi taktovacími frekvencemi, taktovacími periodami, počtem kanálů a vzorkovací frekvencí lze v závislosti na dostupných parametrech vyjádřit pomocí vztahů 3.6 až 3.8, které říkají, že součet doby zpracování jedné periody rozprostírací sekvence ve vysokofrekvenční části a doby zpracování jedné periody rozprostírací sekvence v nízkofrekvenční části nesmí dohromady pro všechny kanály přesáhnout dobu odpovídající periodě této rozprostírací sekvence.

$$n_{ch} \cdot (n_L T_L + n_H T_H) < T_{sekv}, \quad (3.6)$$

$$n_{ch} \cdot \left(n_L \cdot \frac{1}{f_L} + n_H \cdot \frac{1}{f_H} \right) < T_{sekv}, \quad (3.7)$$

kde význam jednotlivých proměnných je následující:

- n_{ch} je počet paralelních kanálů,

- n_L je počet hodinových cyklů taktovacího signálu clk L pro vykonání potřebných výpočtů v nízkofrekvenční části,
- n_H je počet hodinových cyklů taktovacího signálu clk H pro vykonání potřebných výpočtů ve vysokofrekvenční části,
- T_L je perioda taktovacího signálu clk L,
- T_H je perioda taktovacího signálu clk H,
- f_L je frekvence taktovacího signálu clk L,
- f_H je frekvence taktovacího signálu clk H,
- T_{sekv} je perioda rozprostírací sekvence.

Simulací hardwarové implementace byla zjištěna maximální hodnota hodinových cyklů $n_L = 158$, ale pro účely výpočtů byla tato hodnota zaokrouhlena na $n_L = 200$. Počet cyklů n_H je možné určit podle vztahu 3.8 pro případ, kdy vysokofrekvenční část má propustnost jednoho kanálového vzorku na jednu náběžnou hranu taktovacího signálu.

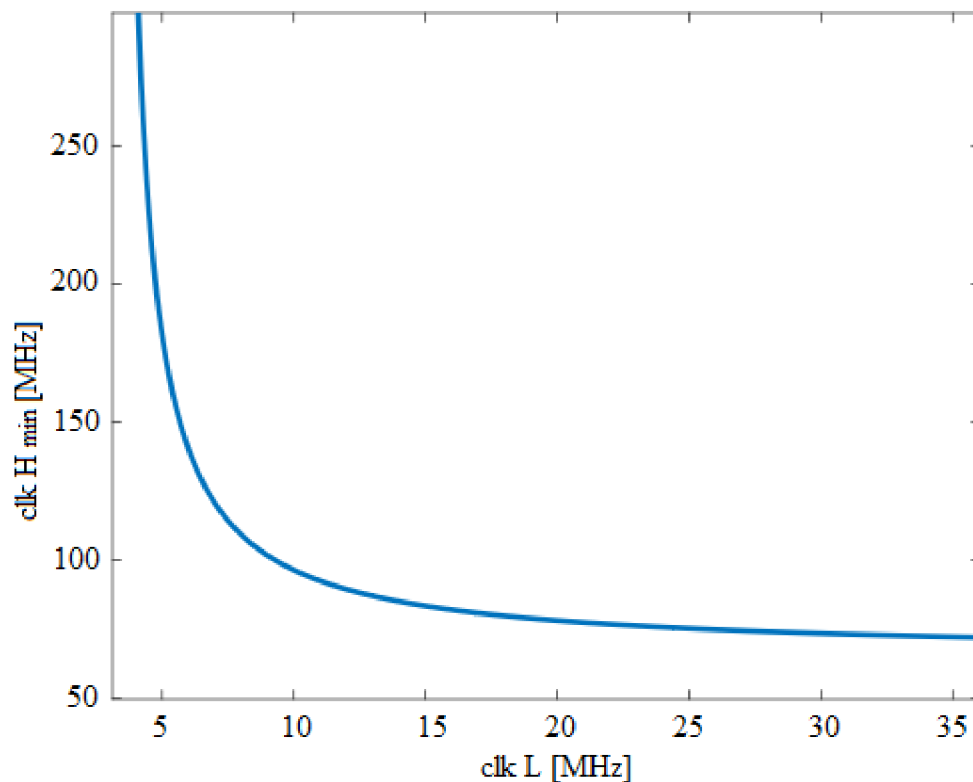
$$n_H = f_s T_{sekv}, \quad (3.8)$$

kde f_s představuje vzorkovací frekvenci signálu a význam zbývajících proměnných je shodný s proměnnými ve vztazích 3.6 a 3.7.

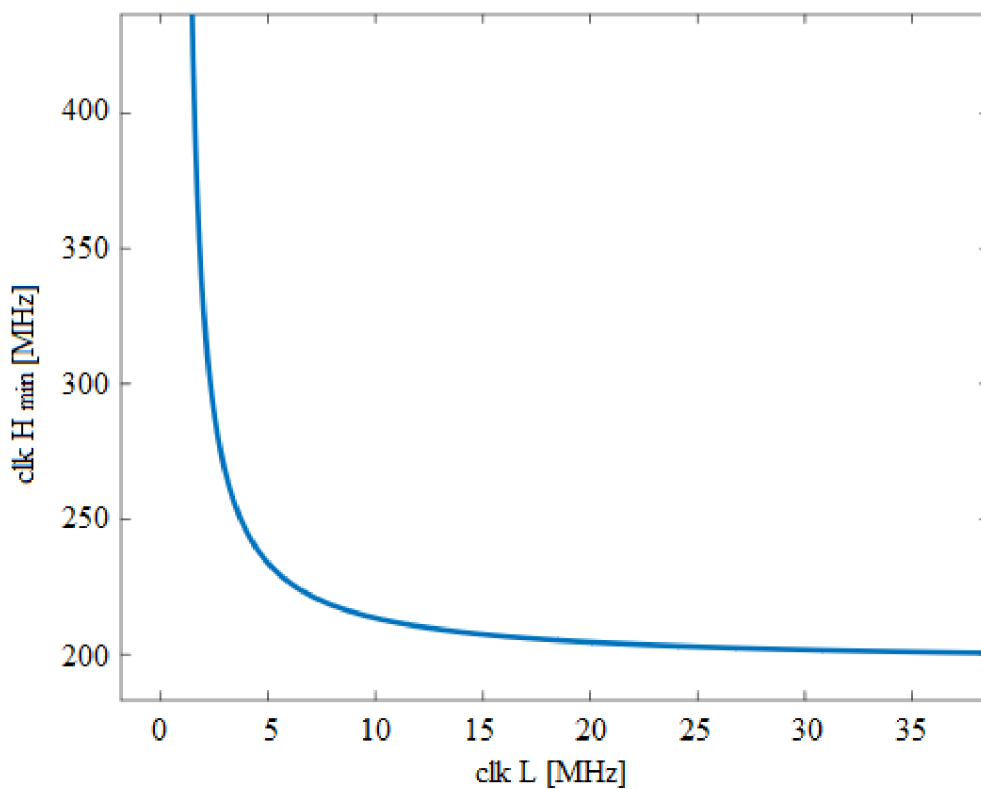
Na základě vztahů 3.6 až 3.8 byly pro oba navigační signály vytvořeny v MATLABu grafy závislosti minimální frekvence taktovacího signálu vysokofrekvenční části na zvolené frekvenci taktovacího signálu nízkofrekvenční části pro počet kanálů $n_{ch} = 16$. Pro signál L1 C/A byla na základě úvah v kapitole 3.1.7 zvolena vzorkovací frekvence $f_{s,L1} = 4,092 \text{ MHz}$ a pro signál E1-OS $f_{s,E1} = 12,276 \text{ MHz}$. Grafy jsou na obrázcích 3.5 a 3.6.

Z grafu na obrázku 3.5 lze vidět, že pro signál L1 C/A má zvyšování taktovací frekvence nízkofrekvenční části za účelem snížení minimální potřebné taktovací frekvence vysokofrekvenční části smysl přibližně do hodnoty $clk L = 25 \text{ MHz}$, pro kterou přibližně platí $clk H_{min} = 75 \text{ MHz}$. Další zvyšování nízkofrekvenční taktovací frekvence má již pouze minimální vliv na snížení minimální potřebné vysokofrekvenční taktovací frekvence.

Z grafu na obrázku 3.6 plyne, že i u signálu E1-OS je pro taktovací frekvenci nízkofrekvenční části optimální hodnota $clk L = 25 \text{ MHz}$, při které přibližně platí $clk H_{min} = 200 \text{ MHz}$.



Obr. 3.5 Závislost minimální hodnoty clk H na hodnotě clk L u signálu L1 C/A



Obr. 3.6 Závislost minimální hodnoty clk H na hodnotě clk L u signálu E1-OS

Zvolenou hodnotou taktovací frekvence nízkofrekvenční se tedy stala hodnota $clk L = 25 MHz$, které bylo bez problémů dosaženo. U obou signálů (L1 C/A a E1-OS) se v hardwarové implementaci podařilo pro taktovací frekvenci vysokofrekvenční části dosáhnout hodnoty přibližně $clk H = 90 MHz$, což je hodnota, která při zvolené frekvenci nízkofrekvenční části bezpečně zajistí možnost paralelního zpracování požadovaných 16 kanálů u signálu L1 C/A. U signálu E1-OS tato dosažená frekvence bohužel představuje přibližně polovinu minimální požadované hodnoty, a proto lze v době vzniku této práce mluvit u tohoto signálu o možnosti paralelního zpracování maximálně 7 až 8 kanálů.

Příčinou relativně nízké dosažené hodnoty frekvence $clk H$ je jednak přítomnost větších kombinačních částí v hardwarové implementaci, a jednak potřeba sčítat fáze rozprostíracích sekvencí s velmi dlouhou zlomkovou částí ve fázovém akumulátoru (viz. tabulka 3.2). Obě příčiny bude možné vyřešit při budoucích optimalizacích zmenšením kombinačních částí za cenu zvýšení celkové latence přijímacího řetězce, což ničemu nevádí, a zvolením vyhovujících vzorkovacích frekvencí popsanych v této kapitole a v kapitole 3.1.7.

3.3 Využití hardwarových prostředků

Využití hardwarových prostředků na použitém FPGA je pro přijímací moduly obou navigačních signálů shrnuto v tabulkách 3.4 a 3.5, ze kterých plyne, že při implementaci obou přijímacích modulů je v průměru celkem využito asi 50% dostupných hardwarových prostředků.

Tabulka 3.4 Využití hardwarových prostředků, L1 C/A

Hardwarová část	Počet použitých jednotek	Využití na použitém FPGA
Lookup tabulky (LUT)	17968	34%
Klopné obvody typu D (FF)	22906	22%
Bloky DSP48	39	18%
Distribuovaná paměť (LUTRAM)	743	4%
Blokovaná paměť (BRAM)	6	4%

Tabulka 3.5 Využití hardwarových prostředků, E1-OS

Hardwarová část	Počet použitých jednotek	Využití na použitém FPGA
Lookup tabulky (LUT)	16641	31%
Klopné obvody typu D (FF)	22205	21%
Bloky DSP48	35	16%
Distribuovaná paměť (LUTRAM)	786	5%
Bloková paměť (BRAM)	13	9%

Největší využití je u LUT tabulek, které dohromady činí 65%. Při budoucích optimalizacích bude možné toto číslo znatelně snížit přesunutím kanálových registrů jednotlivých funkčních bloků do BRAM pamětí, jejichž využití je naopak docela nízké. Důvodem k vytvoření kanálových registrů pomocí LUT a FF bylo jednak snazší odhalování problémů při vytváření hardwarové implementace, a jednak eliminace dalších potencionálních zdrojů chyb, které by počáteční vývoj dále komplikovaly. Zvýšení využití hardwarových prostředků lze naopak očekávat u klopných obvodů (FF) při již zmíněném budoucím zmenšení kombinačních částí a současném zvýšení latence.

4 Verifikace hardwarové implementace

Správnost popisované hardwarové implementace byla na závěr otestována srovnáním sledovacích schopností přijímače vytvořeného v MATLABu a přijímače implementovaného do FPGA.

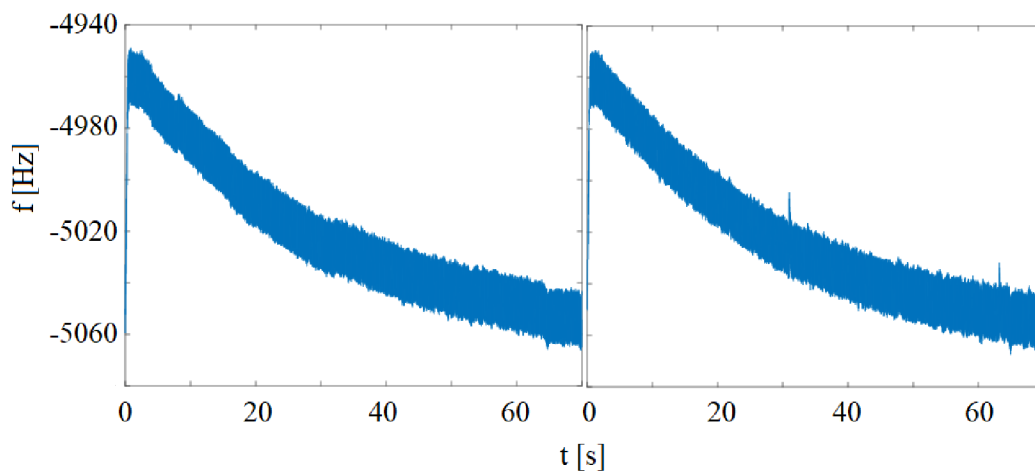
S každým přijatým symbolem a pro všechny aktivní kanály byly aktuální hodnoty fázových inkrementů ukládány do BRAM paměti, která byla pravidelně čtena procesorovou částí přes AXI GP rozhraní. Přečtené hodnoty byly procesorem následně ukládány do souboru na SD-kartu. Z hodnot uložených ve vytvořeném souboru byly pomocí MATLABu vytvořeny grafy odhadovaného frekvenčního posuvu v závislosti na čase.

Následující dvě podkapitoly obsahují srovnání frekvenčního i fázového sledování pro všechny nalezené družice včetně jedné s velmi slabým signálem, u které došlo ke ztrátě synchronizace. Graf vlevo je vždy výsledkem simulace v MATLABu a graf vpravo je vždy výsledkem hardwarové implementace v FPGA. Počáteční frekvenční posuv byl náhodně nastavován na hodnoty ± 90 Hz. Na horizontální ose je vyneseno čas v sekundách a na svislé ose je vyneseno odhadovaný frekvenční posuv v Hertzích.

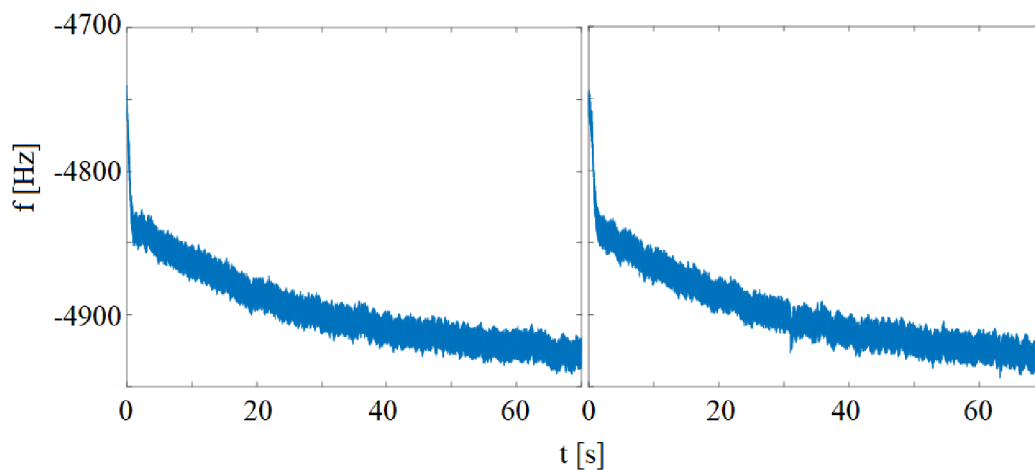
4.1 Signál L1 C/A

V rámci signálu L1 C/A bylo akvizicí nalezeno celkem 9 družic a u jedné z nich, družice číslo 20, došlo v obou případech po několika vteřinách ke ztrátě synchronizace. U zbývajících družic se grafy sledování prakticky shodují včetně šumové šířky pásma, což svědčí o tom, že parametry sledovacích smyček jsou pro oba případy stejné. Výjimkou je pouze družice číslo 7 s dosti slabým signálem, u které bitová synchronizace na FPGA trvala podstatně delší dobu, což lze vyvodit z poměrně dlouhé doby, kterou Costasova smyčka strávila ve stavu s dobou integrace 1 milisekundy, což je stav s větší šumovou šířkou pásma, než tomu je u stavu s dobou integrace 20 milisekund.

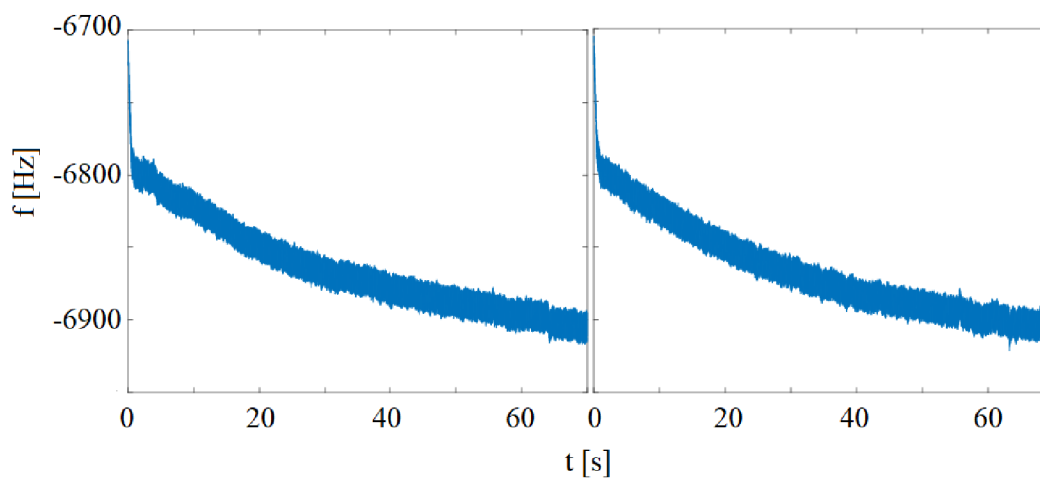
4.1.1 Frekvenční sledování



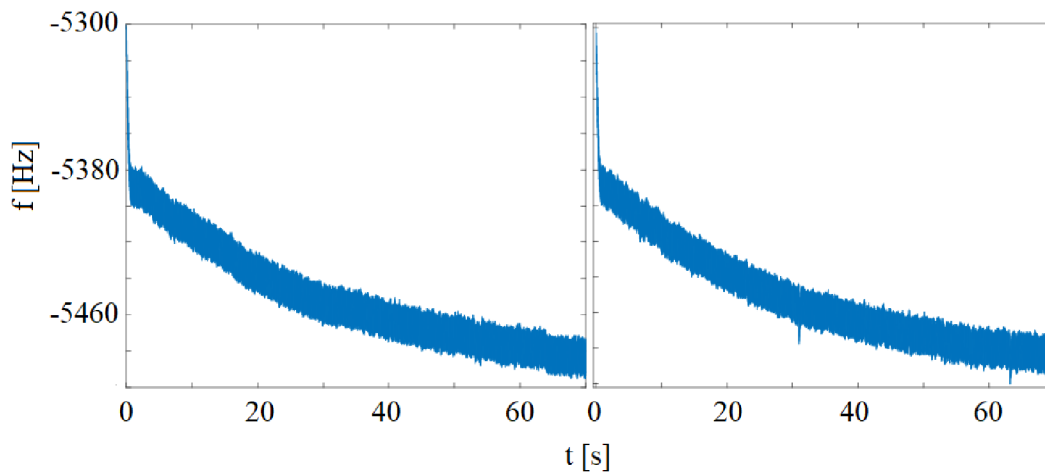
Obr. 4.1 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz



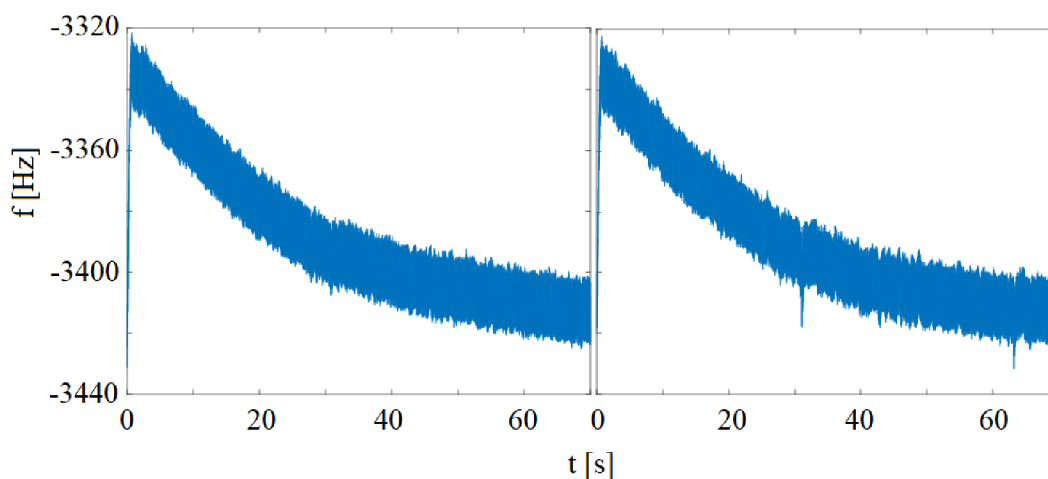
Obr. 4.2 Družice číslo 7, vlevo MATLAB, vpravo FPGA, +90 Hz



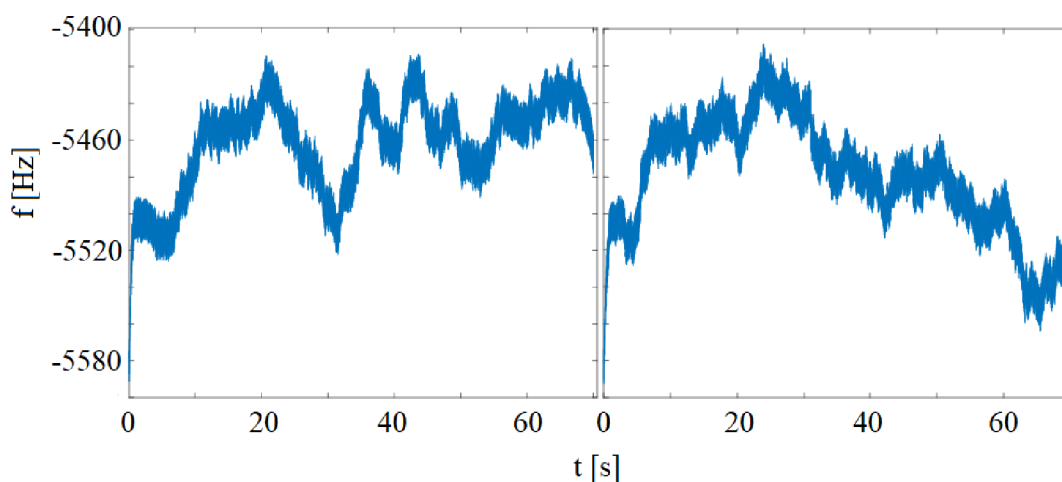
Obr. 4.3 Družice číslo 10, vlevo MATLAB, vpravo FPGA, +90 Hz



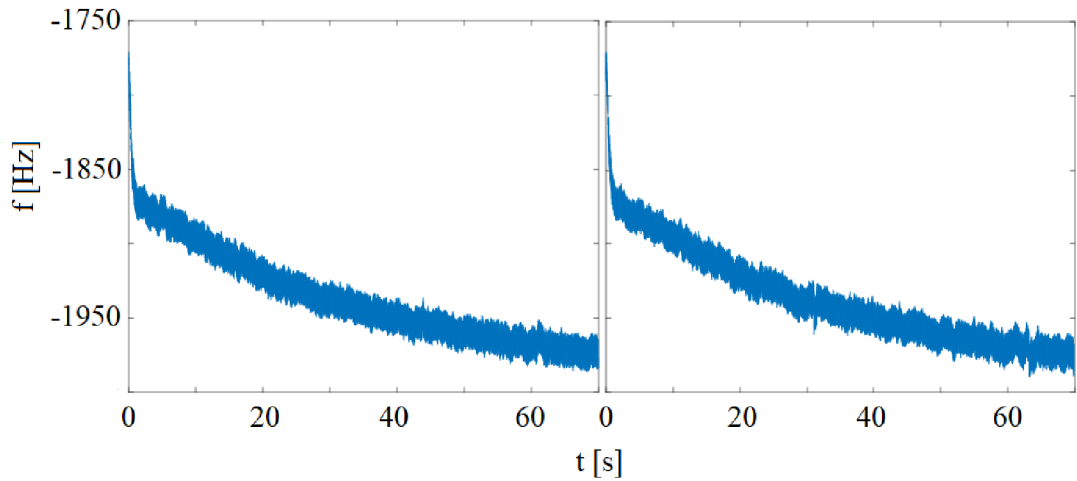
Obr. 4.4 Družice číslo 15, vlevo MATLAB, vpravo FPGA, +90 Hz



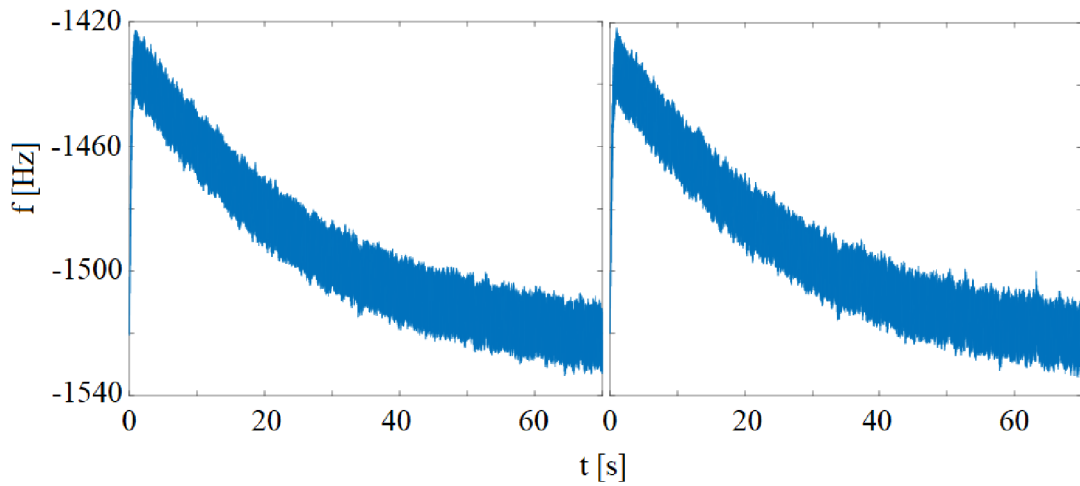
Obr. 4.5 Družice číslo 16, vlevo MATLAB, vpravo FPGA, -90 Hz



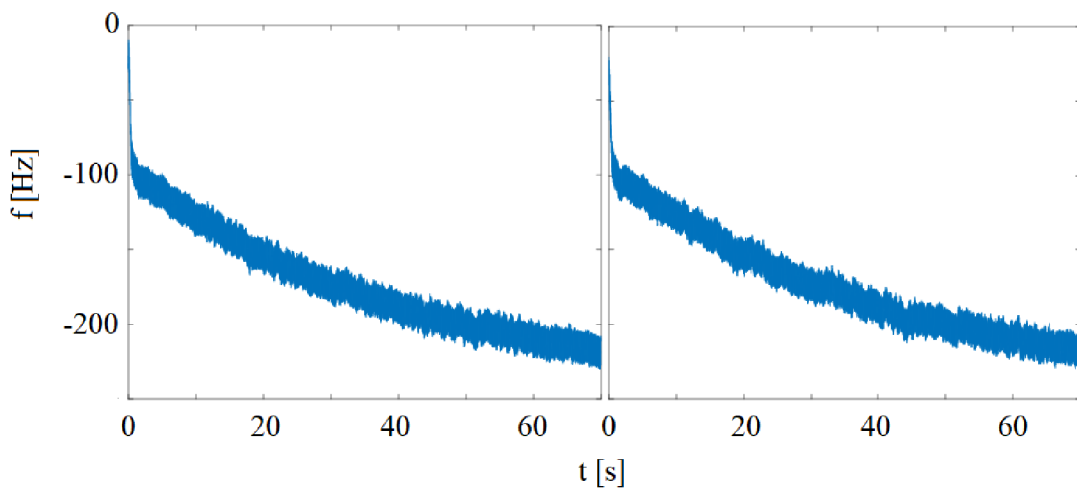
Obr. 4.6 Družice číslo 20, vlevo MATLAB, vpravo FPGA, -90Hz



Obr. 4.7 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz

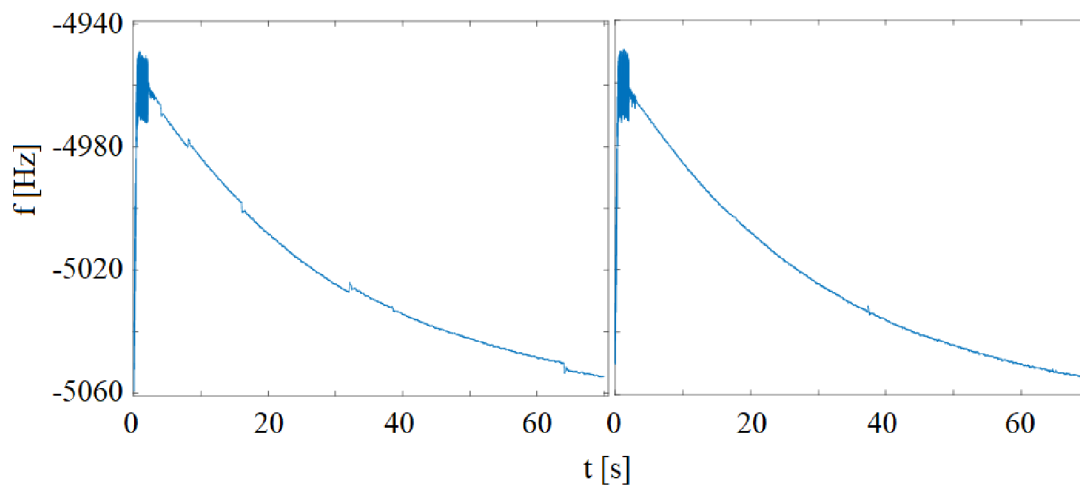


Obr. 4.8 Družice číslo 26, vlevo MATLAB, vpravo FPGA, -90 Hz

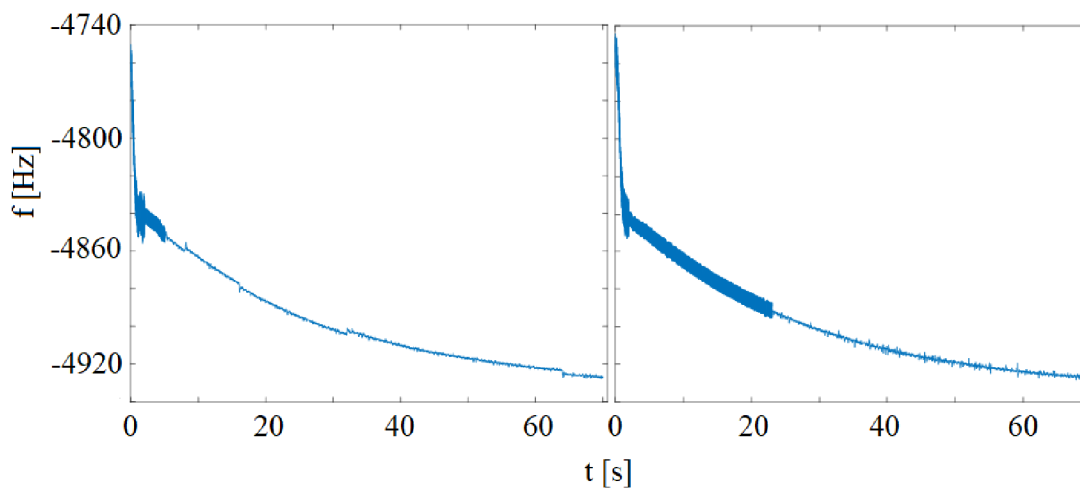


Obr. 4.9 Družice číslo 29, vlevo MATLAB, vpravo FPGA, +90 Hz

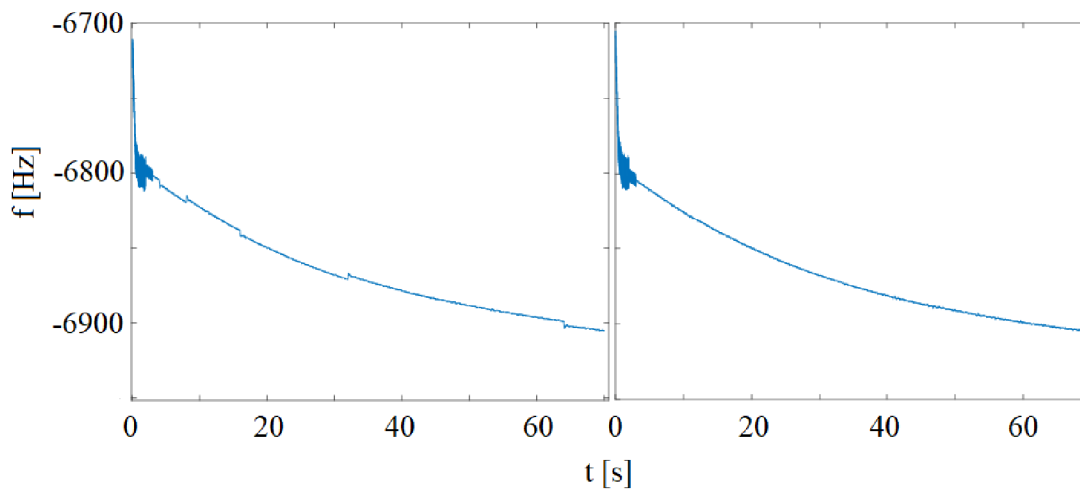
4.1.2 Fázové sledování



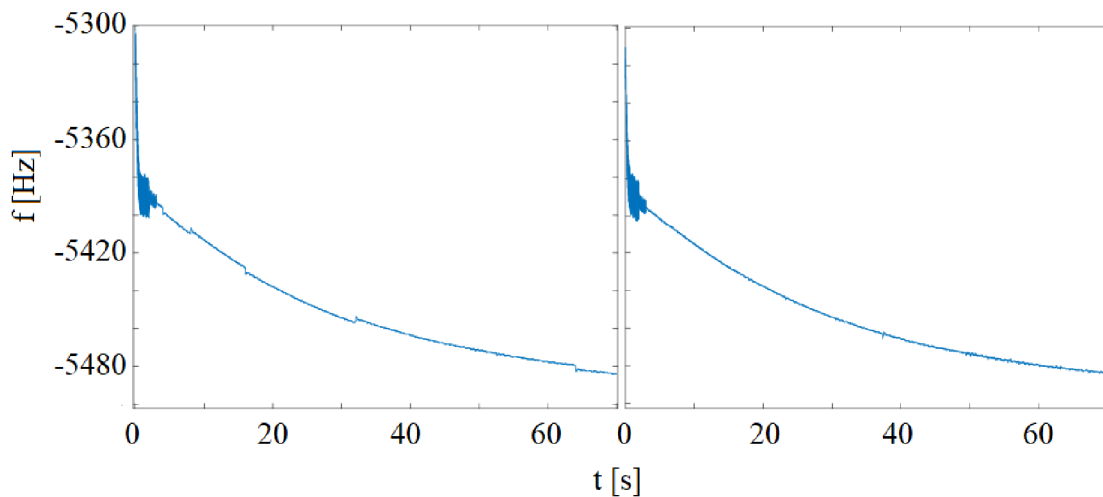
Obr. 4.10 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz



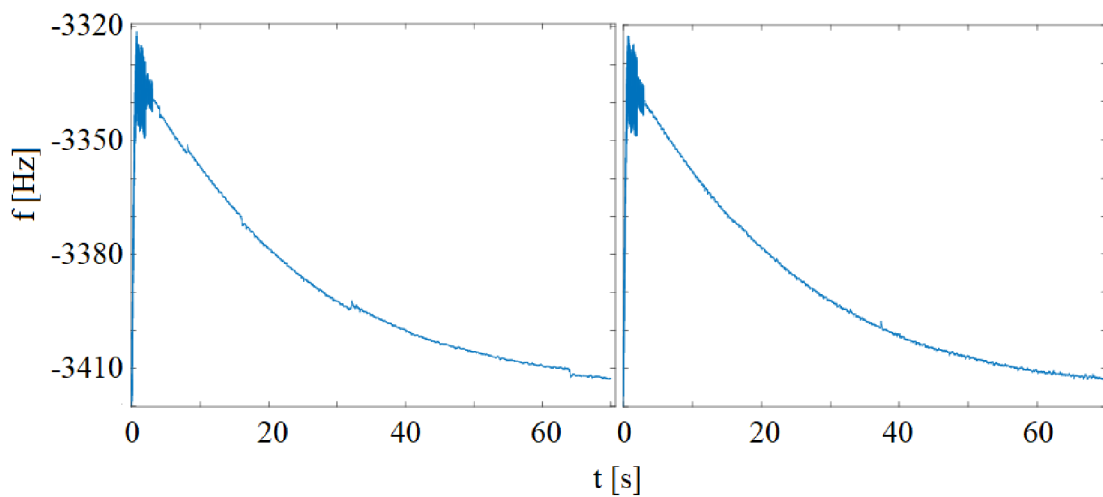
Obr. 4.11 Družice číslo 7, vlevo MATLAB, vpravo FPGA, +90 Hz



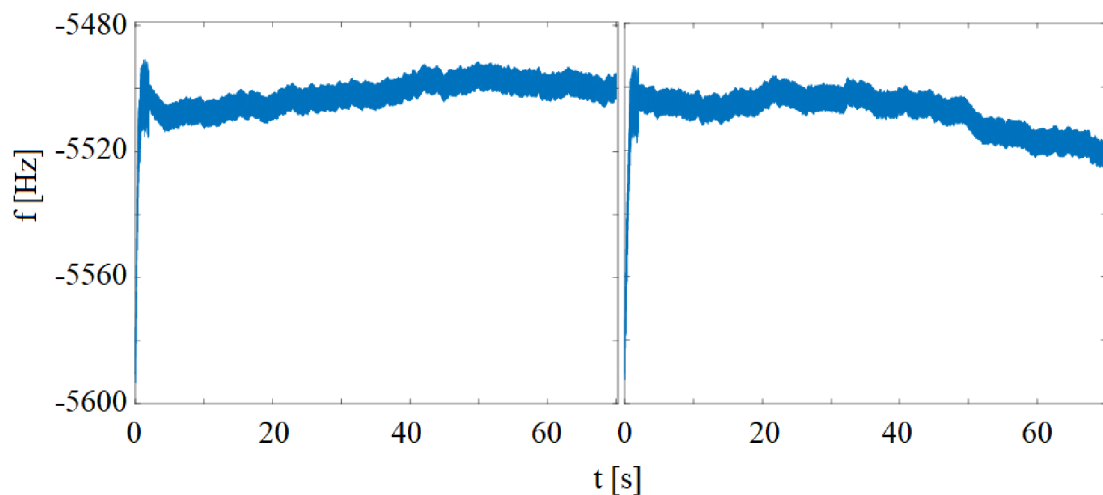
Obr. 4.12 Družice číslo 10, vlevo MATLAB, vpravo FPGA, +90 Hz



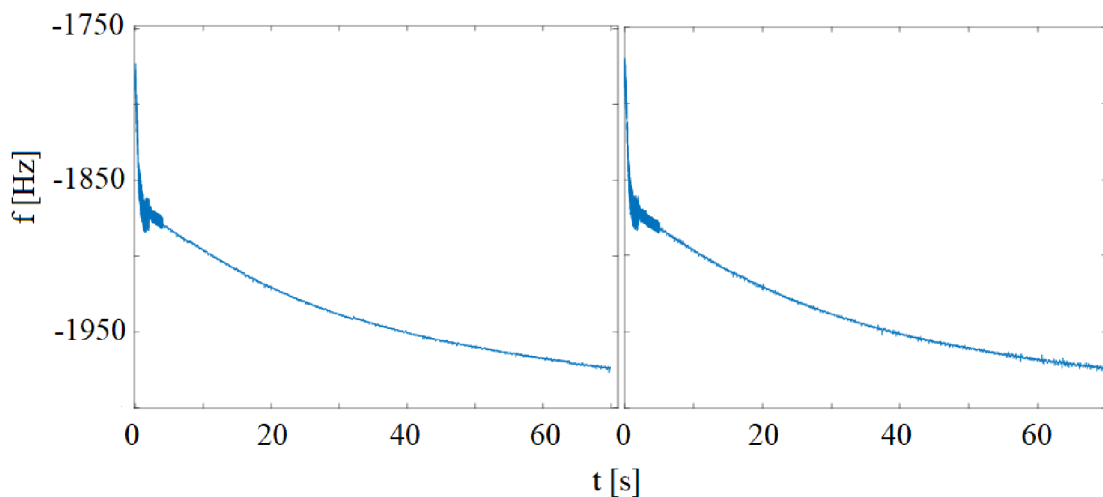
Obr. 4.13 Družice číslo 15, vlevo MATLAB, vpravo FPGA, +90 Hz



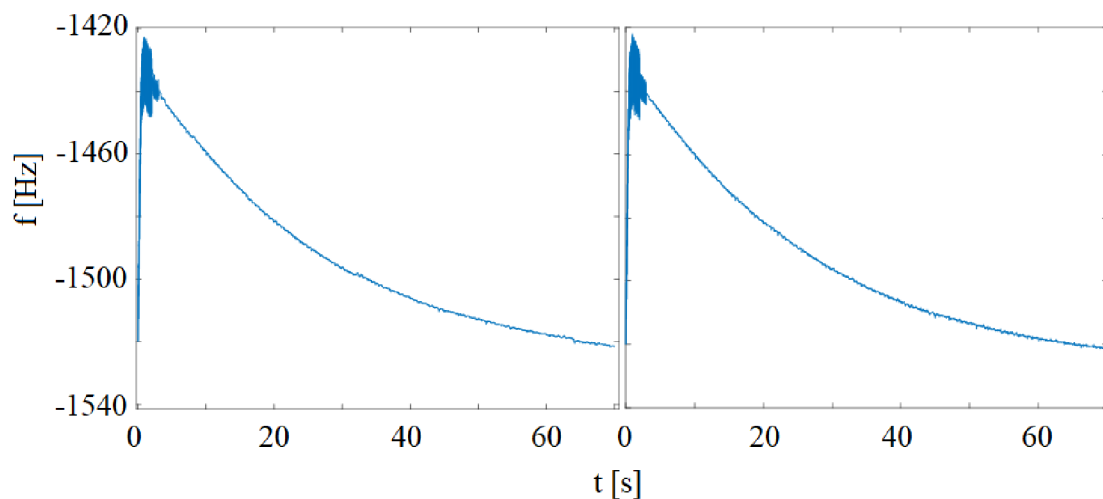
Obr. 4.14 Družice číslo 16, vlevo MATLAB, vpravo FPGA, -90 Hz



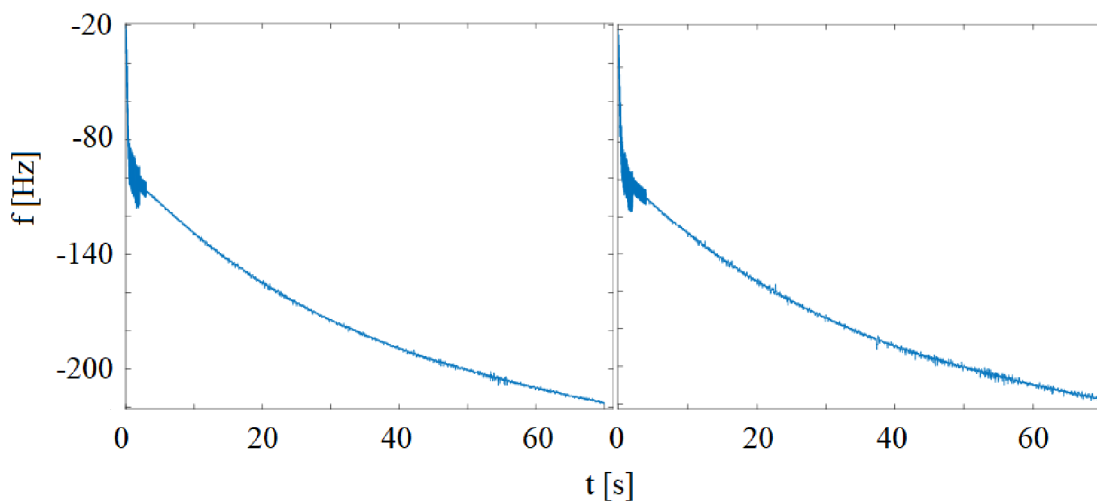
Obr. 4.15 Družice číslo 20, vlevo MATLAB, vpravo FPGA, -90 Hz



Obr. 4.16 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz



Obr. 4.17 Družice číslo 26, vlevo MATLAB, vpravo FPGA, -90 Hz

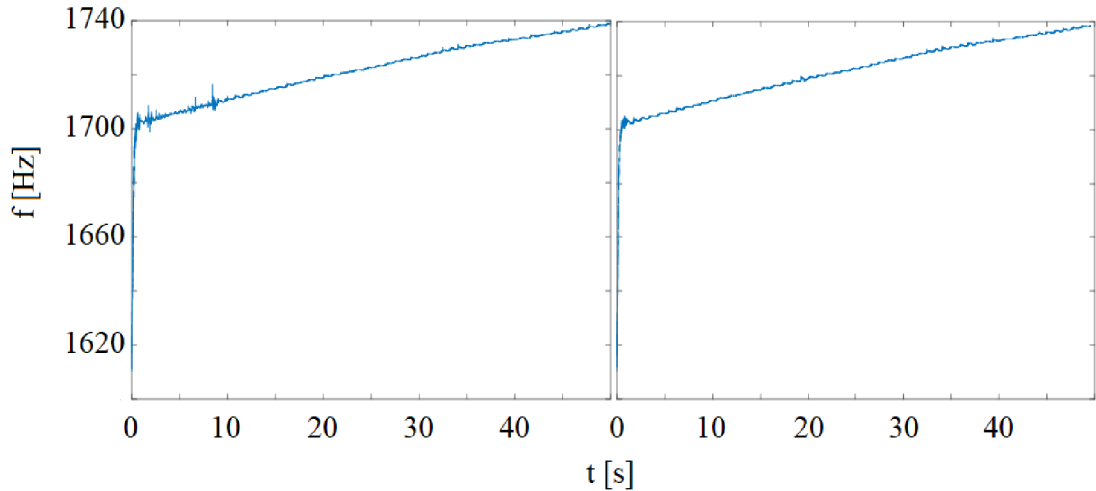


Obr. 4.18 Družice číslo 29, vlevo MATLAB, vpravo FPGA, +90 Hz

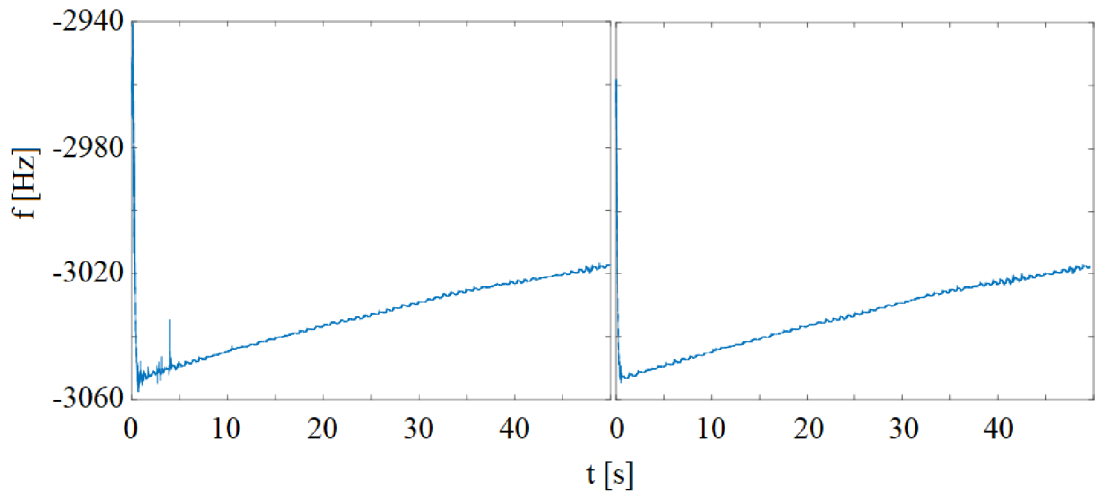
4.2 Signál E1-OS

V rámci signálu E1-OS bylo akvizicí nalezeno 6 družic, z nichž se všechny podařilo úspěšně sledovat jak při simulaci v MATLABu, tak po implementaci na FPGA. Stejně jako u signálu L1 C/A, i zde se průběhy obou grafů, včetně šumových šířek pásma, shodují.

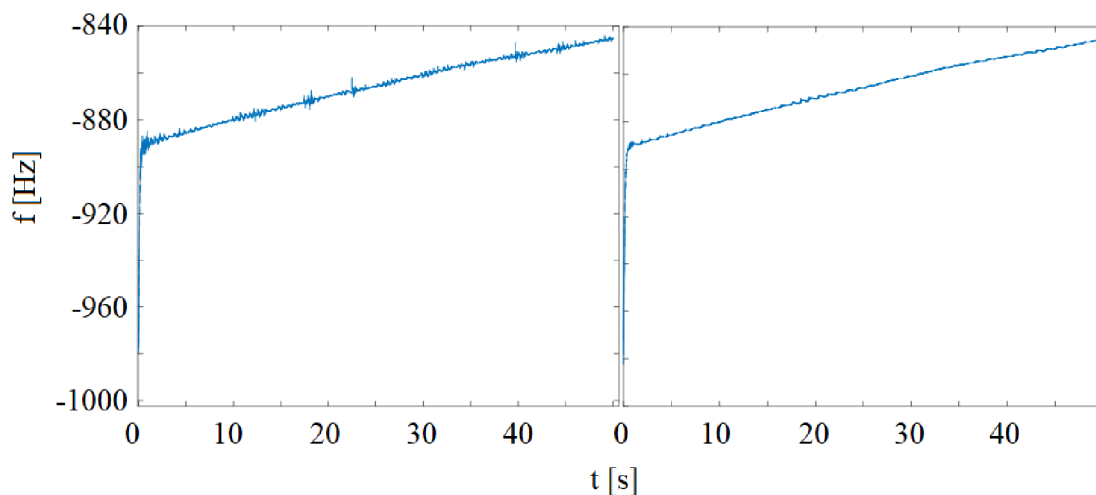
4.2.1 Frekvenční sledování



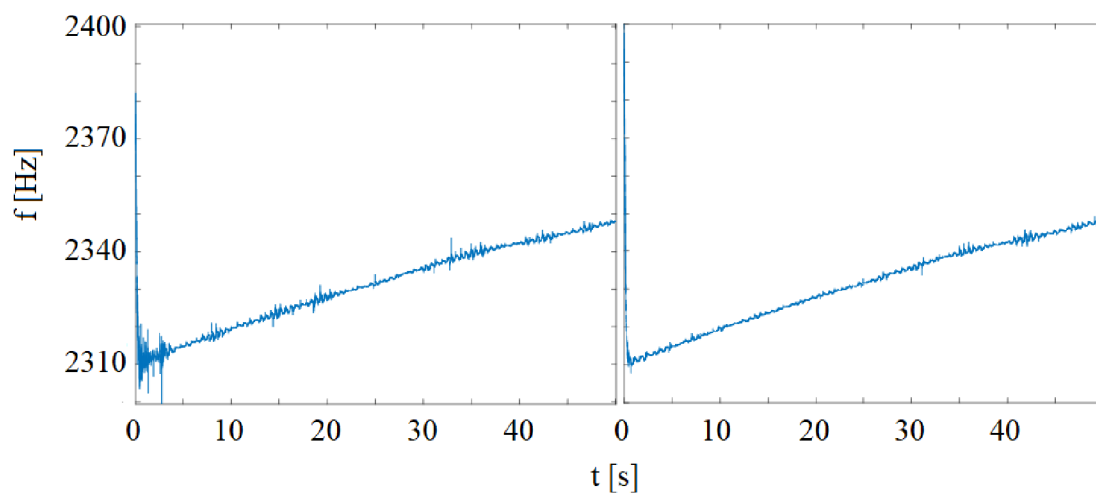
Obr. 4.19 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz



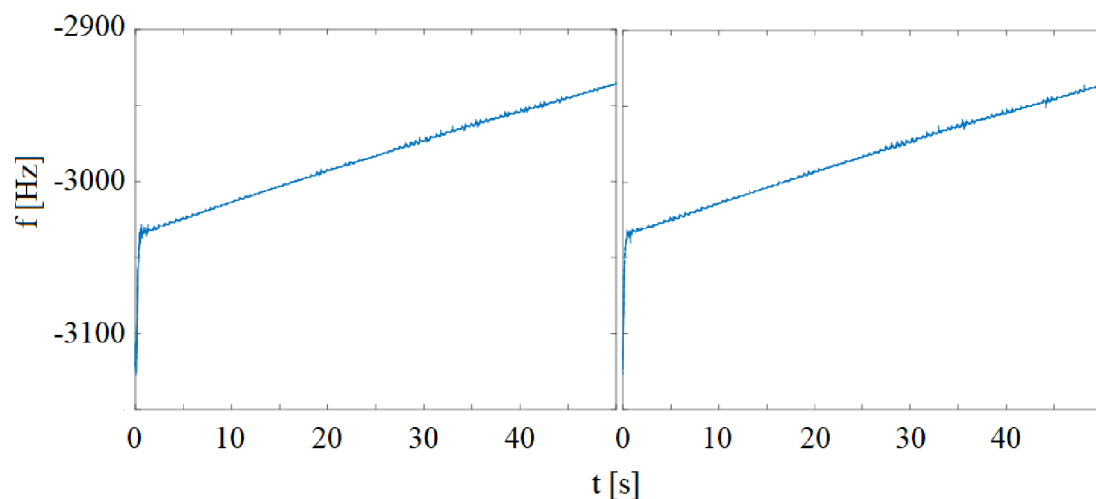
Obr. 4.20 Družice číslo 5, vlevo MATLAB, vpravo FPGA, +90 Hz



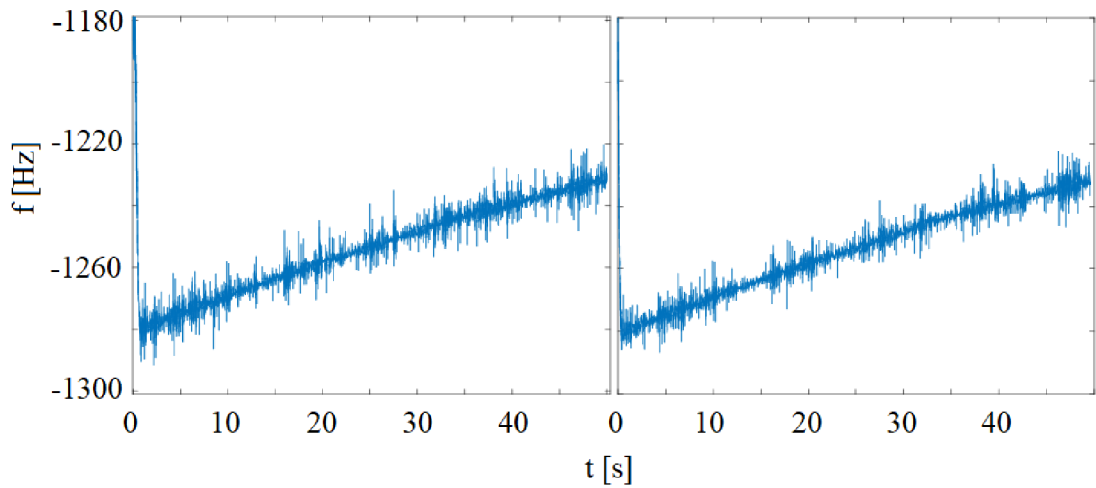
Obr. 4.21 Družice číslo 9, vlevo MATLAB, vpravo FPGA, -90 Hz



Obr. 4.22 Družice číslo 11, vlevo MATLAB, vpravo FPGA, +90 Hz

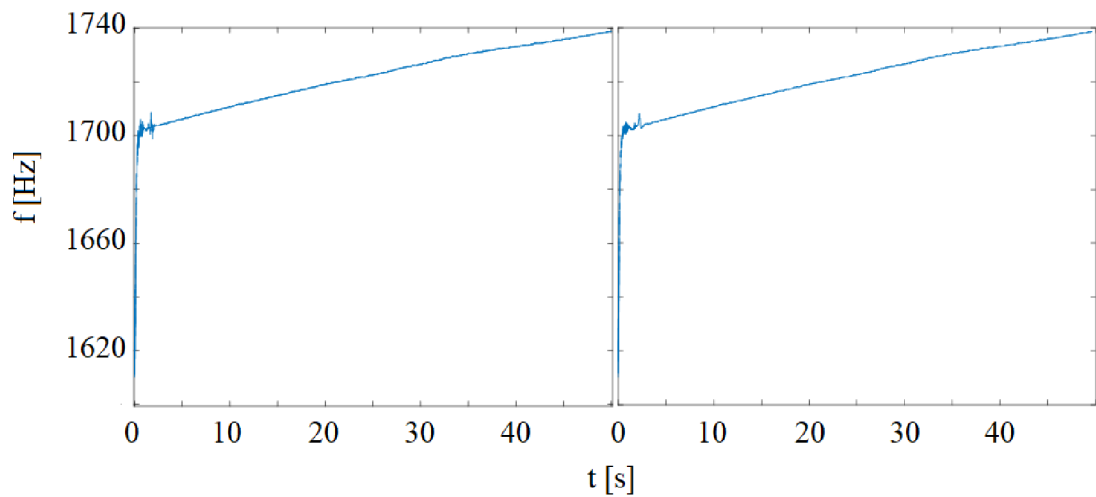


Obr. 4.23 Družice číslo 14, vlevo MATLAB, vpravo FPGA, -90 Hz

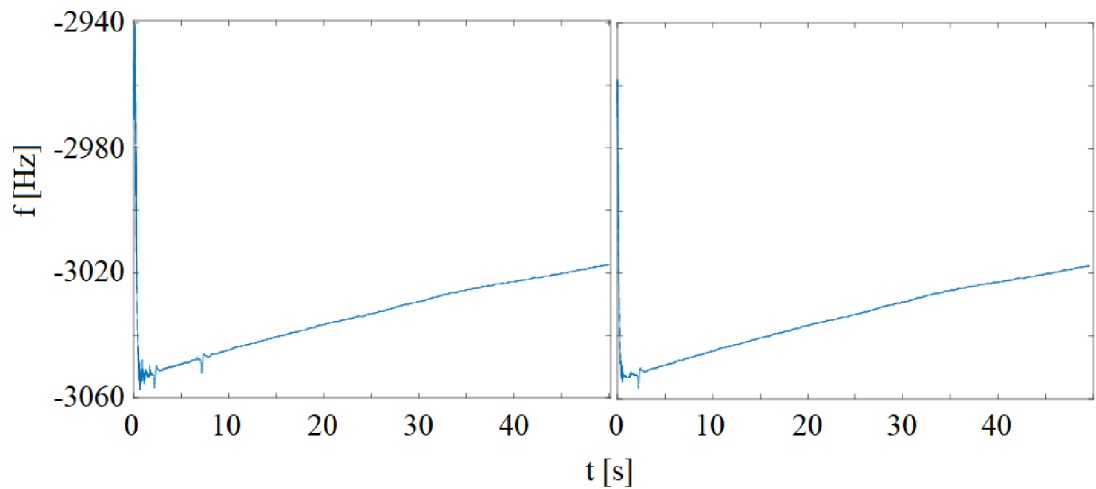


Obr. 4.24 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz

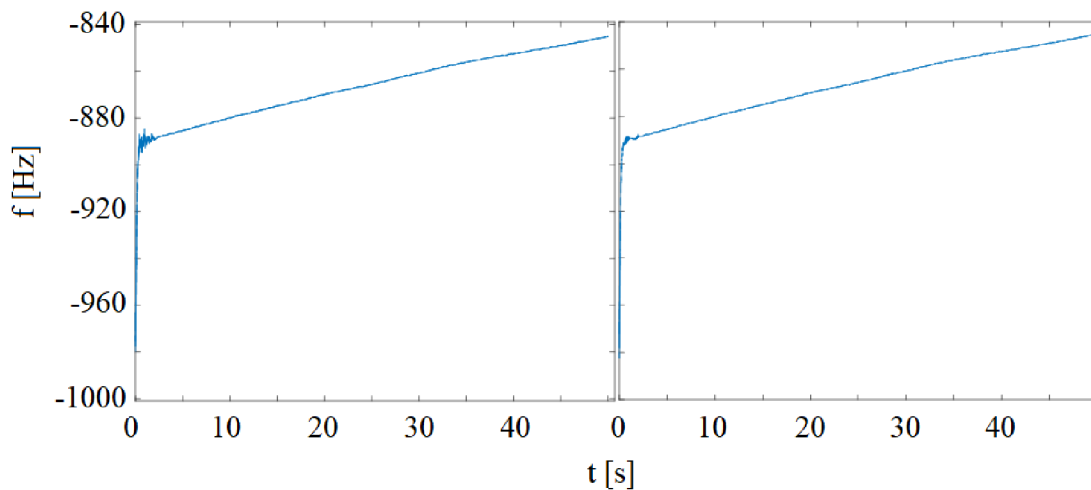
4.2.2 Fázové sledování



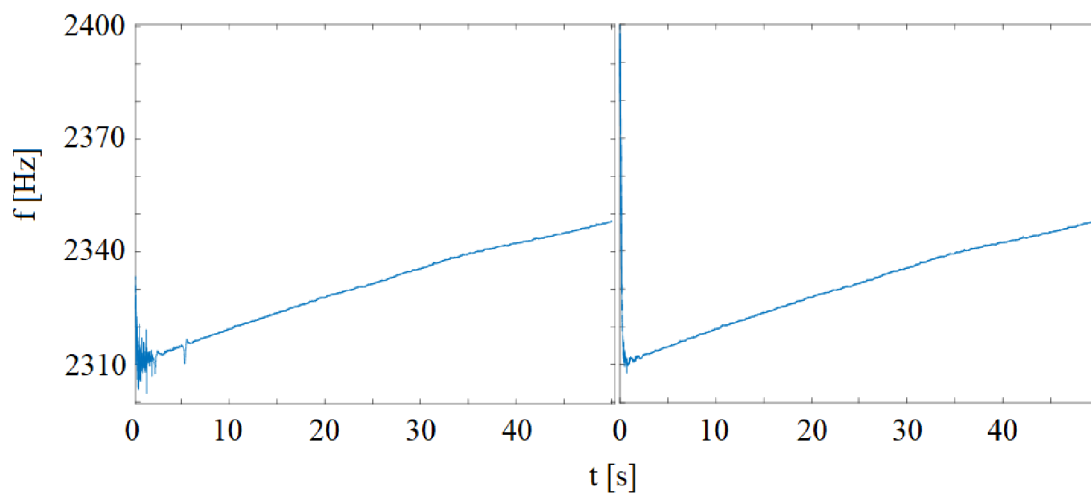
Obr. 4.25 Družice číslo 4, vlevo MATLAB, vpravo FPGA, -90 Hz



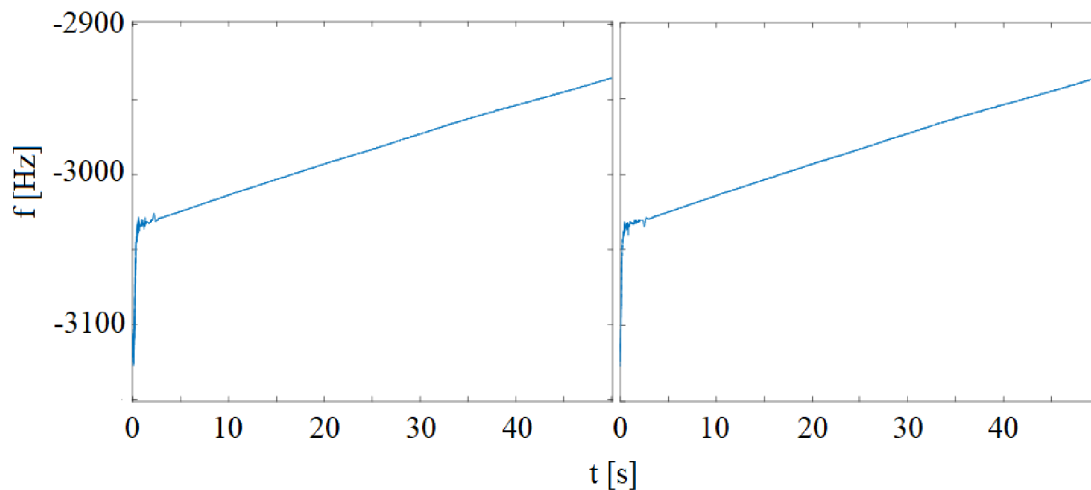
Obr. 4.26 Družice číslo 5, vlevo MATLAB, vpravo FPGA, +90 Hz



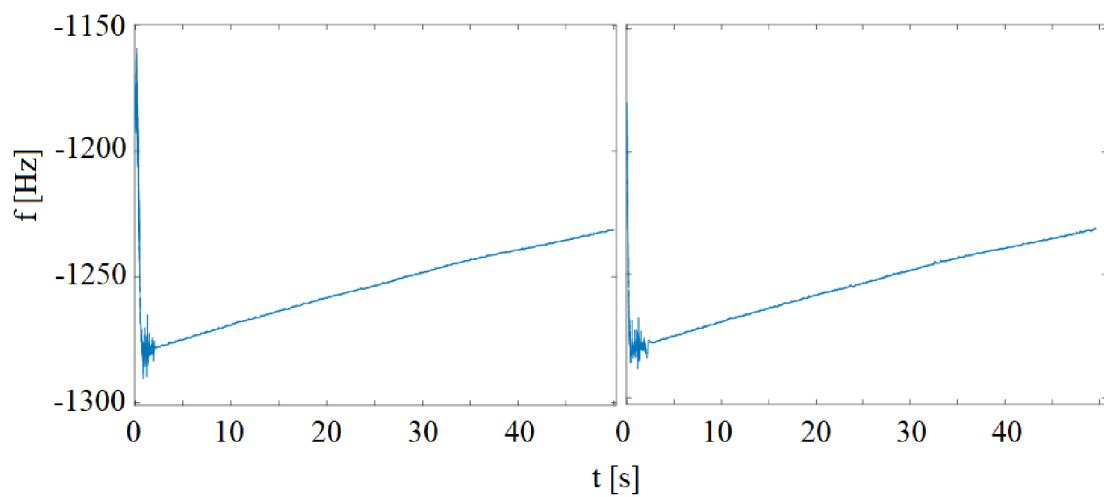
Obr. 4.27 Družice číslo 9, vlevo MATLAB, vpravo FPGA, -90 Hz



Obr. 4.28 Družice číslo 11, vlevo MATLAB, vpravo FPGA, +90 Hz



Obr. 4.29 Družice číslo 14, vlevo MATLAB, vpravo FPGA, -90 Hz



Obr. 4.30 Družice číslo 21, vlevo MATLAB, vpravo FPGA, +90 Hz

Závěr

V první kapitole, která představuje teoretickou část této práce, byly popsány satelitní navigační systémy s důrazem na zkoumané navigační signály L1 C/A a E1-OS systémů GPS a Galileo. V rámci těchto dvou systémů byly, kromě charakteristik výše uvedených navigačních signálů, okrajově zmíněny i informace týkající se ostatních navigačních signálů a konstelace družic systémů.

Ve druhé, prakticky zaměřené kapitole, byla popsána a v MATLABu simulována softwarová verze přijímače zmíněných navigačních signálů, přičemž největší pozornost byla věnována synchronizaci a sledování signálů. Méně podrobně byl zkoumán proces akvizice, v rámci které byl popsán algoritmus, který sice není zcela optimální, ale pro účely této práce plně postačoval. Důvodem, proč nebyla akvizice rozebrána detailněji, je skutečnost, že optimalizovaný algoritmus akvizice, který bude vhodnější pro implementaci do hardwaru pracujícím v reálném čase, je v době vzniku práce již vyvíjen vedoucím této práce, doktorem Filipem Záplatou. Dalšími částmi softwarového přijímače, které byly vytvořeny, jsou detektory synchronizace, bitový synchronizátor a kanálové dekodéry.

V rámci synchronizace a sledování byly implementovány v MATLABu dvě obecně známé metody fázové synchronizace, a sice Costasova smyčka a smyčka fázového závěsu, a čtyři metody frekvenční synchronizace, z nichž dvě metody, synchronizace pomocí FFT a s derivací fáze, byly vytvořeny autorem této práce a další dvě metody, ML frekvenční estimátor a kvadrikorelátor jsou výsledkem studia odborné literatury. Všechny popisované metody synchronizace byly úspěšně simulovány pro oba zkoumané navigační signály s výjimkou smyčky fázového závěsu a kvadrikorelátoru, které byly úspěšně vyzkoušeny pouze pro E1-OS signál systému Galileo, protože tyto dvě metody nejsou odolné vůči bitovým přechodům, které se ve druhém zkoumaném signálu, L1 C/A systému GPS, nachází.

Jako nejvýhodnější kombinace popisovaných synchronizačních metod se vzhledem k implementovatelnosti do hardwaru pracujícím v reálném čase a odolnosti vůči vysoké míře šumu jeví použití Costasovy smyčky a ML estimátoru pro signál L1 C/A a použití smyčky fázového závěsu společně s kvadrikorelátorem pro signál E1-OS.

Správná funkčnost v MATLABu vytvořeného softwarového přijímače byla ověřena implementací algoritmů kanálového dekódování, které potvrdily bezchybný příjem dat pro všechny nalezené družice s výjimkou jedné, u které došlo po několika vteřinách ke ztrátě signálu. Úspěšným kanálovým dekódováním se ověřila i správná funkčnost samotných algoritmů kanálového dekódování včetně Viterbiho dekodéru, který byl schopný zajistit bezchybný příjem dat i u družic, u kterých docházelo k symbolovým chybám.

Hardwarová implementace softwarového přijímače na vývojovou desku osazenou integrovaným obvodem s FPGA byla rozebrána ve třetí kapitole. Implementovány byly smyčky zajišťující synchronizaci a následné sledování frekvence i fáze nosného

kmitočtu a smyčky sledující fázi rozprostírací sekvence. Pro frekvenční a fázovou synchronizaci byly použity smyčky, které byly výše vyhodnoceny jako ty neoptimalnější, což znamená implementaci Costasovy smyčky a ML estimátoru pro příjem signálu L1 C/A a implementaci smyčky PLL a kvadrikorelátoru pro příjem signálu E1-OS.

Správná funkčnost hardwarově implementovaných přijímačů obou navigačních signálů byla otestována ve čtvrté kapitole porovnáním s výsledky simulací v MATLABu, přičemž byly porovnány časové průběhy odhadovaného frekvenčního posuvu všech nalezených družic pro frekvenční i fázové sledování. Porovnávané výsledky synchronizace a sledování nosného kmitočtu byly u všech družic prakticky shodné, na základě čehož lze říct, že popisovaná hardwarová implementace je správná.

I přes to, že přijímací moduly obou navigačních signálů pracují správně, stále zde je prostor pro řadu optimalizací, které mohou být předmětem navazujícího vývoje. Hlavním nedostatkem je relativně nízká maximální taktovací frekvence systému, které se podařilo dosáhnout, v důsledku čehož přijímací modul signálu E1-OS není schopen paralelně zpracovávat plný počet kanálů, kterých je v každém přijímacím modulu 16. Maximální taktovací frekvenci by mělo jít významně zvýšit rozdělením velkých kombinačních částí do několika menších celků vzájemně oddělených klopnými obvody. Dále bude možné snížit poměrně velké využití LUT tabulek na FPGA přesunutím kanálových registrů do BRAM paměti, které mají naopak v aktuálním stavu celkem nízké využití. Chyby a jiné problémy, které se přirozeně vyskytnou během zmíněných, případně i jiných, úprav přijímače, však již bude mnohem snazší odhalit a vyřešit díky přítomnosti funkčního, i když ne zcela optimálního, přijímače.

Na popisovanou hardwarovou implementaci bude také možné navázat vytvořením softwarové části mající na starosti akvizici, kanálové dekodování, získávání informací na základě přijatých dat a celkové řízení přijímače.

Vzhledem k výše uvedenému lze zadání diplomové práce považovat za splněné.

Literatura

- [1] PETROVSKI, Ivan G. *GPS, GLONASS, Galileo and BeiDou for mobile devices*. Cambridge: Cambridge University Press, 2014. ISBN 978-1-107-03584-3
- [2] BORRE, K. *A software-defined GPS and Galileo receiver: a single-frequency approach*. Boston, Mass.: Birkhäuser, c2007. Applied and numerical harmonic analysis. ISBN 978-0-8176-4390-4
- [3] Doppler effect. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-08]. Dostupné z: https://en.wikipedia.org/wiki/Doppler_effect
- [4] US DEPARTMENT OF DEFENSE. *Global Positioning System, Standard Positioning Service, Performance Standard* [pdf]. 2008 [cit. 2019-12-08]. Dostupné z: <https://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>
- [5] Space Segment. *GPS: The Global Positioning System* [online]. [cit. 2019-12-08]. Dostupné z: <https://www.gps.gov/systems/gps/space/>
- [6] US DEPARTMENT OF DEFENSE. *IS-GPS-200* [pdf]. 2019 [cit. 2019-12-08]. Dostupné z: <https://www.gps.gov/technical/icwg/IS-GPS-200K.pdf>
- [7] GPS Constellation Status. *US Coast Guard Navigation Center* [online]. [cit. 2019-12-08]. Dostupné z: <https://www.navcen.uscg.gov/?Do=constellationStatus>
- [8] US DEPARTMENT OF DEFENSE. *IS-GPS-705* [pdf]. 2019 [cit. 2019-12-08]. Dostupné z: <https://www.gps.gov/technical/icwg/IS-GPS-705F.pdf>
- [9] How many Galileo satellites are now in orbit? *GALILEO| European Global Navigation Satellite System* [online]. [cit. 2019-12-08]. Dostupné z: <https://galileognss.eu/how-many-galileo-satellites-are-now-in-orbit/>
- [10] FAQ | European Global Navigation Satellite Systems Agency. *European GNSS Agency* [online]. [cit. 2019-12-08]. Dostupné z: <https://www.gsa.europa.eu/european-gnss/galileo/faq#satellites>
- [11] EUROPEAN UNION. *Open Service, Service Definition Document* [pdf]. 2016 [cit. 2019-12-08]. Dostupné z: <https://galileognss.eu/wp-content/uploads/2017/07/Galileo-OS-SDD.pdf>
- [12] EUROPEAN UNION. *European GNSS (Galileo) Open Service, Signal In Space Interface Control Document* [pdf]. 2015 [cit. 2019-12-08]. Dostupné z: https://galileognss.eu/wp-content/uploads/2015/12/Galileo_OS_SIS_ICD_v1.2.pdf
- [13] Globální družicový polohový systém. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-08]. Dostupné z: https://cs.wikipedia.org/wiki/Glob%C3%A1ln%C3%AD_dru%C5%BEicov%C3%BD_polohov%C3%BD_syst%C3%A9m

- [14] Cross-correlation. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-08]. Dostupné z: <https://en.wikipedia.org/wiki/Cross-correlation>
- [15] JACOBSEN, Eric. *Digital PLL Design and Analysis Topics* [pdf]. 2010 [cit. 2019-12-08]. Dostupné z: https://www.compdsp.com/presentations/Jacobsen/abineau_dpll_analysis.pdf
- [16] BEST, Roland. *Costas loops*. New York, NY: Springer Berlin Heidelberg, 2018. ISBN 978-3-319-72007-4
- [17] GARDNER, Floyd Martin. *Phaselock techniques*. 3rd ed. Hoboken, NJ: John Wiley, 2005. ISBN 0-471-43063-3
- [18] ZÁPLATA, F. *Advanced algorithms for satellite communication signal processing. Dissertation*. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2015. 116 pages
- [19] MENGALI, Umberto a Aldo N. D'ANDREA. *Synchronization techniques for digital receivers*. New York: Plenum Press, c1997. ISBN 0-306-45725-3
- [20] Lock Detectors. In: *Navipedia* [online]. [cit. 2019-12-08]. Dostupné z: https://gssc.esa.int/navipedia/index.php/Lock_Detectors#Phase_Lock_Detectors
- [21] MIT. *Viterbi Decoding of Convolutional Codes* [pdf]. 2019 [cit. 2019-12-08]. Dostupné z: <http://web.mit.edu/6.02/www/f2010/handouts/lectures/L9.pdf>
- [22] US DEPARTMENT OF DEFENSE. *2017 Federal Radionavigation Plan* [pdf]. 2017 [cit. 2019-12-08]. Dostupné z: <https://www.navcen.uscg.gov/pdf/FederalRadioNavigationPlan2017.pdf>
- [23] XILINX. *DDS Compiler v6.0, LogiCORE IP Product Guide* [pdf]. 2017 [cit. 2020-05-25]. Dostupné z: https://www.xilinx.com/support/documentation/ip_documentation/dds_compiler/v6_0/pg141-dds-compiler.pdf

Seznam symbolů, veličin a zkratek

L1	GPS pásmo 1575,42 MHz
L2	GPS pásmo 1227,60 MHz
L5	GPS pásmo 1176,45 MHz
E1	Galileo pásmo 1575,42 MHz
E5	Galileo pásmo 1191,795 MHz
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
CDMA	Code Division Multiple Access
C/A	Coarse Acquisition
BPSK	Binary Phase Shift Keying
LFSR	Linear Feedback Shift Register
OS	Open Service
CS	Commercial Service
PRS	Public Regulated Service
SAR	Search and Rescue
CBOC	Composite Binary Offset Carrier
BOC	Binary Offset Carrier
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
DLL	Delay Locked Loop
I	Inphase
Q	Quadrature
E	Early
P	Prompt
L	Late
PLL	Phase Locked Loop
ML	Maximum Likelihood
TLM	Telemetry
CRC	Cyclic Redundancy Check
HOW	Hand Over Word
TOW	Time Of Week
ARM	Advanced RISC Machine
RISC	Reduced Instruction Set Computer
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Scale Integrated Circuit
IP	Intellectual Property
FIFO	First In First Out

EPL	Early, Prompt, Late
FAT	File Allocation Table
DRAM	Dynamic Random Access Memory
BRAM	Block Random Access Memory
ROM	Read Only Memory
DMA	Direct Memory Access