



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

**Fakulta Elektrotechnická
Katedra radioelektroniky**

Generátor GNSS signálu

GNSS signal generator

Bakalářská práce

Studijní program: Komunikace, multimédia a elektronika
Studijní obor: Komunikační technika

Vedoucí práce: Ing. Petr Kačmařík, Ph.D.

Radek Šindelář



Prohlášení

Prohlašuji, že předložená bakalářská práce Generátor GNSS signálu je mým původním autorským dílem, které jsem vypracoval samostatně. Veškerá literatura a další zdroje, z nichž jsem čerpal, jsou uvedeny v seznamu použité literatury.

Souhlasím s půjčováním, zveřejněním a dalším využitím práce, pokud s tím bude souhlasit katedra radioelektroniky FEL ČVUT v Praze.

V Praze dne

Podpis studenta



České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra elektromagnetického pole

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Radek Šindelář**

Studijní program: Komunikace, multimédia a elektronika
Obor: Komunikační technika

Název tématu: **Generátor GNSS signálu**

Pokyny pro vypracování:

Student se seznámí se strukturou signálů moderních družicových navigačních systémů (GPS, Glonass, Galileo). V rámci práce student vytvoří systém pro generování konkrétního GNSS signálu v místě příjmu (tj. včetně vlivu kanálu). Práce by měla být realizována jako toolbox pro prostředí MATLAB s grafickým uživatelským rozhraním.

Seznam odborné literatury:

[1] Kaplan, E.D.: Understanding GPS: Principles and Applications. Norwood, Artech House, Inc. 1996.

Vedoucí: Ing. Petr Kačmařík, Ph.D.

Platnost zadání: ZS 2012/2013

prof. Ing. Miloš Mazánek, CSc.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 1. 2012



Poděkování

Chtěl bych poděkovat Ing. Petru Kačmaříkovi, Ph.D. za odborné vedení, důležité připomínky a rady při zpracování této práce. Cením si jeho pomoci při sehnání potřebné literatury a času, jenž mi formou konzultace poskytl.

Děkuji Ing. Martině Nejepsové za poskytnutou podporu při řešení různých problémů v programovacím jazyku MATLAB.

Děkuji dále své rodině a přátelům za trpělivost a toleranci při tvorbě této práce.



Anotace

Cílem této bakalářské práce je sestavení toolboxu v jazyce MATLAB pro projekt Witch Navigator. Celý program obsahuje funkce pro generování jednotlivých navigačních pásem z určitých navigačních systémů jednotlivých družic. Zpracování těchto signálů je na bázi pásmového signálu či komplexní obálky. Zpracování je vykreslováno ve spektrální výkonové hustotě, autokorelační funkci a signálu samotném. V generovaných signálech je zohledněno místo příjmu Dopplerovským posuvem, zpožděním signálu a přidáním šumu.

Tento program by měl sloužit pro testování příjmu navigačních systémů.

Klíčové pojmy: GALILEO, GLONASS, GPS, GNSS, L1C/A, L2C, L5, E1, Dopplerovo zpoždění, šum, spektrální výkonová hustota, autokorelační funkce, toolbox, MATLAB

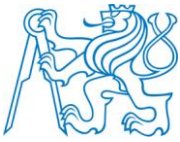


Annotation

The target of this bachelor work is to put together a toolbox in MATLAB for the project called Witch Navigator. The whole programme includes functions for generating of particular navigation bands from individual satellites of the navigation systems. The processing of these signals is based on band signal or complex envelope. The signal processing is displayed in the power spectrum density, the autocorrelation function and the signal itself. All is taken into account in terms of the doppler shift, the signal delay and the noise.

This programme should be used for testing of navigation systems.

Key words: GALILEO, GLONASS, GPS, GNSS, L1C/A, L2C, L5, E1, Doppler count, noise, power spectral density, auto-correlation function, toolbox, MATLAB



Obsah

SEZNAM POUŽITÝCH ZKRATEK	9
SEZNAM POUŽITÝCH SYMBOLŮ	10
ÚVOD	11
TEORETICKÁ ČÁST	12
1. NAVIGAČNÍ SYSTÉMY	13
1.1. Tranzit - NAVSAT	14
1.2. GPS – NAVSTAR	15
1.3. GLONASS	16
1.4. GALILEO	17
1.4.1. OS – Open Service	18
1.4.2. CS – Commercial Service	18
1.4.3. SOL – Safety Of Life service	18
1.4.4. PRS – Public Regulated Service	18
2. TEORETICKÝ ROZBOR	19
2.1. Teoretická východiska	19
2.1.1. Fourierova transformace	19
2.1.2. Analytický signál	19
2.1.3. Pásmový signál	19
2.1.4. Komplexní obálka	20
2.1.5. Autokorelační funkce	20
2.1.6. Spektrální výkonová hustota	20
2.1.7. Použité modulace	21
2.1.7.1. Modulace BPSK	21
2.1.7.2. Modulace QPSK	21
2.1.7.3. Modulace CBOC	22
2.2. Signál GPS	22
2.2.1. L1 C/A	23
2.2.1.1. C/A – kód	23
2.2.1.2. PRN – kód	24
2.2.2. L2	25
2.2.2.1. P(Y) – kód	25
2.2.2.2. Anti-spoofing (AS)	25
2.2.2.3. Selective Availability (SA)	26
2.2.3. L1 C	26
2.2.3.1. PRN – kód	26
2.2.4. L2 C	29
2.2.4.1. PRN – kód	29
2.2.5. L5	30
2.2.5.1. PRN – kód	30
2.3. Signál GLONASS	32



2.3.1.	L1	32
2.3.1.1.	PRN	32
2.3.2.	L2	32
2.3.2.1.	PRN	33
2.4.	Signál GALILEO	33
2.4.1.	E5	33
2.4.1.1.	E5a	33
2.4.1.2.	E5b.....	35
2.4.2.	E1	36
2.4.3.	E6	37
2.5.	Navigační zpráva	37
2.5.1.	NAV	37
2.5.2.	C/NAV	38
2.5.3.	F/NAV.....	38
2.5.4.	I/NAV.....	38
PRAKTICKÁ ČÁST		39
3.	GNSS GENERÁTOR.....	40
3.1.	Popis hlavního okna před výpočtem.....	43
3.1.1.	Parametry signálu.....	44
3.1.1.1.	Navigační systém	44
3.1.1.2.	Pásmo	44
3.1.1.3.	PRN	45
3.1.1.4.	Tint	45
3.1.1.5.	Fsa a Tsa	45
3.1.1.6.	BP a CE.....	46
3.1.1.7.	Save.....	46
3.1.2.	Navigační zpráva	46
3.1.2.1.	Ze souboru.....	46
3.1.2.2.	Jedničková.....	47
3.1.2.3.	Nulová	47
3.1.2.4.	Střídavá.....	47
3.1.2.5.	Ruční	47
3.1.3.	Místo příjmu.....	47
3.1.3.1.	C/N0.....	47
3.1.3.2.	Fd	48
3.1.3.3.	Tau.....	48
3.1.4.	Funkční tlačítka hlavního okna	48
3.1.4.1.	Start	48
3.1.4.2.	Konec.....	48
3.2.	Popis hlavního okna po výpočtu.....	49
3.2.1.	Graf.....	49
3.2.2.	Signal	49
3.2.3.	PSD	50



3.2.4. ACF.....	50
VÝSLEDKY PRÁCE.....	51
ZÁVĚR.....	53
POUŽITÉ ZDROJE.....	54
SEZNAM OBRÁZKŮ	56
SEZNAM TABULEK	57
SEZNAM PŘÍLOH	58



Seznam použitých zkratk

ACF	Auto-Correlation Function	Autokorelační funkce
AS	Anti-Spoofing	Šifrovaný P(Y)-kód
BPSK	Binary Phase Shift Keying	Modulace s binárním klíčováním fáze
C/A	Coarse/Acquisition	Civilní signál (volně přeloženo)
CBOC	Composite Binary Offset Carrier	Binární modulace se zpožděnou nosnou
CDMA	Code Division Multiple Access	Kódové dělení přenosových kanálů
CS	Commercial service	Komerční služby (volně přeloženo)
DSSS	Direct Sequence Spread Spectrum	Přímého rozprostřeného spektra
EHF	Extremely High Frequency	Extrémně vysoká frekvence
FDMA	Frequency Division Multiple Access	Frekvenční dělení přenosových kanálů
GCC	Galileo Control Centers	Kontrolní centrum Galilea
GLONASS	Globalnaja navigacionnaja sputnikovaja sistěma	Globální navigační sputníkův systém
GNSS	Global Navigation Satellite System	Globální družicový navigační systém
GPS	Global Positioning System	Globální zaměřovací systém
HF	High Frequency	Vysoká frekvence
ICAO	International Civil Aviation Organization	Mezinárodní organizace pro civilní letectví
IOC	Initial Operational Capability	Počáteční operační způsobilost
LF	Low Frequency	Nízká frekvence
MF	Medium Frequency	Střední frekvence
NATO	North Atlantic Treaty Organization	Severoatlantická aliance
NAVSOC	Naval Satellite Operations Center	Námořní operační satelitní centrum
OS	Open Service	Volné služby (volně přeloženo)
P(Y)	Precision code	Uautorizovaný kód (volně přeloženo)
PRN	Pseudo Random Noise	Pseudonáhodná posloupnost
PRS	Public Regulated Service	Veřejná regulovaná služba (volně přeloženo)
PSD	Power Spectral Density	Spektrální výkonová hustota
QPSK	Quadrature Phase Shift Keying	Kvadrurní fázová modulace
SA	Selective Availability	Selektivní dostupnost
SAR	Search And Rescue	Služba nouzové lokalizace
SHF	Super High Frequency	Super vysoká frekvence
SOL	Safety Of Life service	Služba bezpečí života (volně přeloženo)
UHF	Ultra High Frequency	Ultra vysoká frekvence
VHF	Very High Frequency	Velmi vysoká frekvence
VLF	Very Low Frequency	Velmi nízká frekvence
XOR	eXclusive OR	Exklusivní disjunkce



Seznam použitých symbolů

B	šířka pásma
c	rychlost světla
λ	vlnová rychlost
f	frekvence
f_0	referenční frekvence
f_{01}	referenční frekvence družic z GLONASS L1
f_{02}	referenční frekvence družic z GLONASS L2
f_T	přijatá frekvence
f_R	frekvence vnitřního oscilátoru
f_{k1}	nosné frekvence družic z GLONASS L1
f_{k2}	nosné frekvence družic z GLONASS L2
Δf_1	rozdílová frekvence družic z GLONASS L1
Δf_2	rozdílová frekvence družic z GLONASS L2
ω_c	nosný kmitočet
π	Ludolfovo číslo
ω_h	krajní kmitočet pásma
ω_l	krajní kmitočet pásma
α	řecké písmeno Alfa
A_c	amplituda nosné frekvence
$C(t)$	C/A-kód
$D(t)$	datová zpráva



ÚVOD

Tato práce se zabývá problematikou navigačních systémů GPS, GALILEO a GLONASS, jejich vysílací částí pásem a posléze charakteristikami signálu ve formě spektrální výkonové hustoty nebo autokorelační funkce. Vše je řešeno s ohledem na místo příjmu, čímž jsou zahrnuty rušivé parametry na vysílaný signál jako Dopplerův posun, zpoždění signálu a šum.

Cílem práce je seznámit se s moderními družicovými navigačními systémy a pomocí struktury signálů jednotlivých systémů vytvořit toolbox, který tyto signály bude generovat. Tento toolbox je realizován v prostředí MATLAB s grafickým uživatelským rozhraním. Výstupy tohoto programu jsou jednak charakteristiky signálů jako PSD a ACF, ale i vzorky samotného signálu. Práce je konstruována pro vývoj a testování GNSS signálu v projektu „Witch navigator“, který je realizován na katedře radioelektroniky FEL ČVUT.

Práce je členěna na dvě části – teoretickou a praktickou. Teoretická se zabývá popisem systémů GPS, GLONASS a GALILEO, složením jejich pásem a kódováním jednotlivých družic. Dále jsou v této kapitole vysvětleny pojmy PSD, ACF, Dopplerův posun, atd.

Praktická část obsahuje popis programu „GNSS_generator.m“ ve formě návodu pro obsluhu toolboxu. V návodu jsou popsány veškeré části toolboxu, jejich ošetření a podmínky pro vygenerování signálu. V praktické části jsou dále uvedeny výsledky generovaných signálů určité družice.



TEORETICKÁ ČÁST



1. Navigační systémy

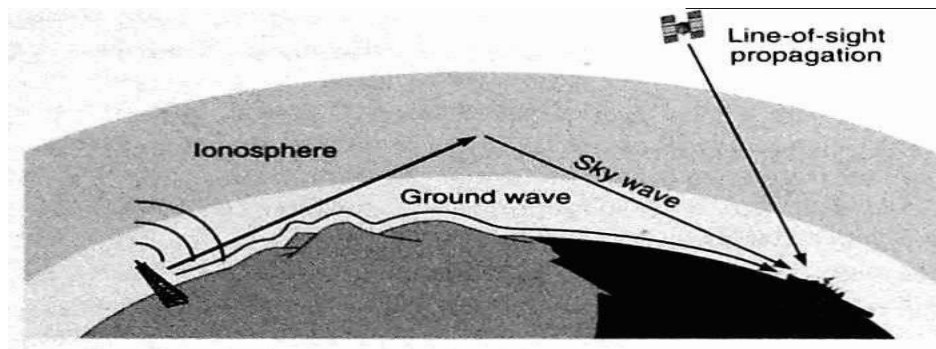
Díky nárůstu dopravního průmyslu a větším nárokům na orientaci po Zemi je kladen větší důraz na přesnější určování polohy. To má za následek modernizaci některých již spuštěných navigačních systémů GPS a GLONASS, a také vývoj systému GALILEO.

Podle [2] jsou základem navigačních systémů principy radiových vln, které odpovídají rozsahům elektromagnetických vln o frekvencích od 10 kHz do 30 GHz (*Tabulka 1*). Rychlost radiových vln je přibližně 3×10^8 m/s. To odpovídá vlnovým délkám od 10 km do 10 cm, vypočtených ze vzorce:

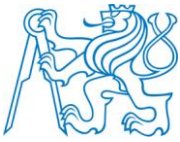
$$\lambda = \frac{c}{f} \quad (1.1)$$

Všechna pásma se šíří jako radiové vlny. Speciálně pásma SHF a EHF spadají pod mikrovlny. Navigační systémy používají pásma VHF a UHF. Rozlišujeme tři typy šíření radiových vln viz *Obr. 1*.

- 1) Přímé šíření (Vzduch – Země)
- 2) Odraz od ionosféry (Vzduch – Vzduch)
- 3) Pozemní šíření



Obr. 1 - Šíření radiových vln (převzato z [2])



Tabulka 1 - Frekvenčních pásem (převzato z [2])

Pásmo	Frekvence	Vlnová délka
Velmi malá frekvence (VLF)	< 30kHz	> 10 km
Malá frekvence (LF)	30 – 300 kHz	1 – 10 km
Střední frekvence (MF)	300 kHz – 3 MHz	100 m – 1 km
Vysoká frekvence (HF)	3 – 30 MHz	10 – 100 m
Velmi vysoká frekvence (VHF)	30 – 300 MHz	1 – 10 m
Ultra vysoká frekvence (UHF)	300 MHz – 3GHz	10 cm – 1 m
Super vysoká frekvence (SHF)	3 – 30 GHz	1 – 10 cm

Dodnes vzniklo mnoho navigačních systémů, jako např.: Loran, Omega, Decca, a jiné. Nejvíce však za zmínku stojí navigační systémy Tranzit, GPS, Glonass, Galileo.

1.1. Tranzit - NAVSAT

Systém byl spuštěn v roce 1964 pro potřeby vojenského námořnictva Spojených států. Pro civilní použití byl uvolněn o tři roky později a pokusně byl používán i v letectví. Své uplatnění našel nejprve v plném nasazení, později však jen jako podpůrný prostředek k inerciálním navigačním systémům¹.

Kosmický segment mělo tvořit šest družic ve výšce 1 000 km nad povrchem Země, přesněji tři družice aktivní a tři družice jako záložní na oběžné dráze. Oběh družic je tvořen na třech kruhových drahách, což neumožňuje nepřetržitý provoz po celé Zemi. Družice vysílaly na kmitočtech 150 a 400 MHz s výkonem 1W, s přibližnou dobou oběhu 107 minut. [2]

Řídicí a kontrolní segment řídil NAVSOC². Ten se skládal z několika pracovišť a měl za úkol především predikci orbity družic a korekci palubních hodin na 12 hodin dopředu.[20]

Systém Tranzit je založen na Dopplerově efektu³. V tomto systému jde o porovnávání přijaté frekvence s frekvencí vnitřního oscilátoru přijímače pomocí (1.2):

$$Doppler\ count = \int_{t_i-1}^{t_i} (f_T - f_R) dt, \quad (1.2)$$

¹ Systém pro orientaci v prostoru využívá různých senzorů, gyroskopů, akcelerometrů, atd.

² Naval Satellite Operations Center

³ Změna frekvence vlny při pohybu vyzařovacího tělesa oproti přijímacímu tělesu.

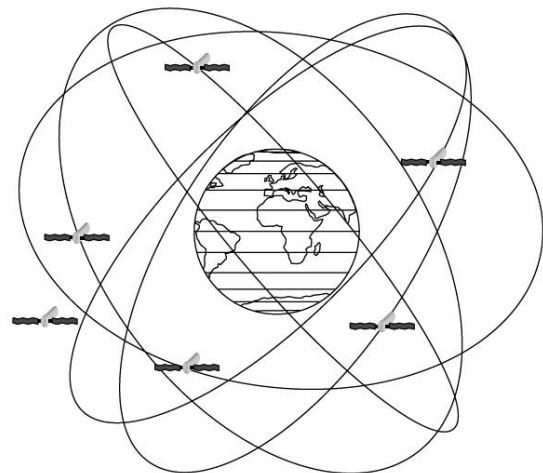


kde f_R je frekvence vnitřního oscilátoru přijímače, f_T je přijatá frekvence. Rozdíl těchto dvou frekvencí se provádí několikrát při průchodu satelitu po dobu asi 10-20 minut. Výsledné Dopplerovo zpoždění bývá až $\pm 9,8$ kHz podle přibližování či oddalování družice od přijímače. [2] [20]

1.2. GPS – NAVSTAR

Byl vyvinut Ministerstvem obrany Spojených států amerických za účelem použití pro armádu. Díky havárii raketoplánu Challenger bylo plné nasazení dosaženo 8. prosince 1993. Toto datum je označováno jako Počáteční operační způsobilost (IOC⁴). Přesnost příjmu se liší podle skupiny uživatelů. [2] [11]

Kosmický segment navigačního systému tvoří 31 aktivních družic. Tyto družice obíhají na šesti kruhových drahách (viz **Obr. 2**) ve výšce 20 200 km nad povrchem Země. Vzájemné posunutí drah družic je 60° a oběžná doba jednoho satelitu je 11 h 58 min. Vysílací frekvence systému GPS jsou 1 575,42 MHz, 1 227,6 MHz a 1 176,45 MHz. [5]



Obr. 2 – GPS (převzato z [18])

Řídicí a kosmický segment se skládá z velitelství, řídicího střediska, 3 povelových stanic a 18 monitorovacích stanic. Tento segment zasílá povely družicím a provádí údržbu atomových hodin. Celková práce segmentu je zveřejňována v navigační zprávě všech družic. [5]

GPS – NAVSTAR je systém založen na principu kódového určení polohy. Díky informacím o poloze družic je určení souřadnic uživatele dáno pomocí rovnice (1.3) pro čtyři družice ($n=1, 2, 3, 4$), kde X, Y, Z a T jsou neznámé parametry uživatele, c je rychlost světla a x, y, z, t jsou známé parametry čtyř družic.

⁴ Initial Operational Capability

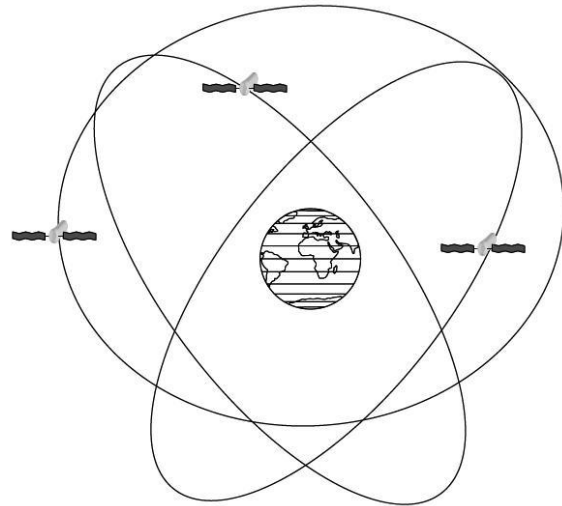


$$(X - x_n)^2 + (Y - y_n)^2 + (Z - z_n)^2 = [(T - t_n)c]^2, \quad (1.3)$$

1.3. GLONASS

Ruský navigační systém je z pohledu drah družic dosti podobný GPS systému. Vyvíjet se začal v polovině 70. let dvacátého století. Tento systém je pod správou vojenských kosmických sil ruského ministerstva obrany. Byl přihlášen Sovětským svazem jako civilní navigační systém, ale je známo, že také vysílá signály určeny pro vojenské použití. V roce 1988 zveřejnil neúplné záznamy parametrů systému v ICAO⁵. [11]

Kosmický segment je projektován pro 24 družic ve třech drahách (viz *Obr. 3*). V dnešní době je aktivních 24 družic, které jsou vůči sobě posunuty o 120° a obíhají kolem Země ve výšce 19 100 km. Systém používá dva signály L₁, L₂. Ve fázi testování je jeden z modernizovaných signálů na frekvenci L₃= 1202.025. Signály L₁ a L₂ mají frekvence 1 602,5 – 1 615,5 MHz, 1 246,4 – 1 257,3 MHz. Oproti systému GPS kde se používá kódové dělení CDMA⁶, se u typu Glonass využívá dělení kmitočtového FDMA⁷. U tohoto dělení je zapotřebí širší vysílací pásmo a je zde problém s interferencemi mezi kanály. Signál L₁ je určen pro civilní uživatele s přesností 75 m v horizontální poloze. Pro vojenské ruské účely byl dříve určen signál L₂, jehož přesnost byla utajována. Nyní však GLONASS vysílá na frekvenci L₂ civilní signál. Jako modernizace tohoto systému je připravován signál na frekvenci L₃ pro civilní použití. Toto pásmo bude používat dělení CDMA. [7]



Obr. 3 - GLONASS (převzato z [18])

⁵ International Civil Aviation Organization – Mezinárodní organizace pro civilní letectví

⁶ Družice vysílají na jednom kmitočtu speciální PRN kód (viz Teoretický rozbor).

⁷ Družice vysílají na různých kmitočtech identický kód.

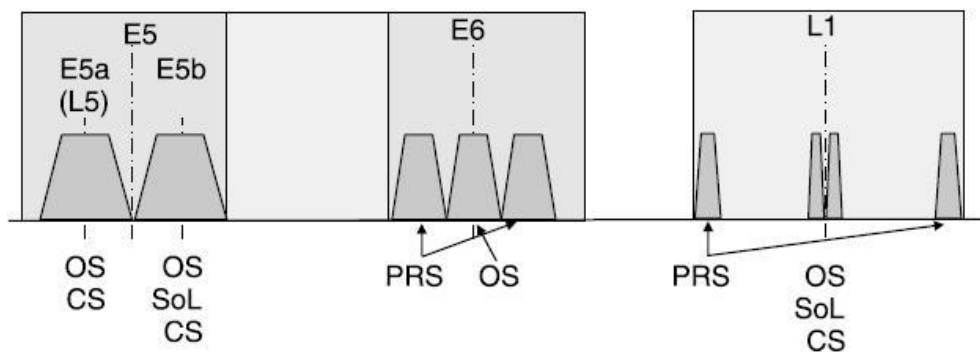


Řídicí a kontrolní segment je složen z řídicího střediska, 3 rozšířených stanic, 5 povelových stanic a 10 monitorovacích stanic. Funkce segmentu je stejná jako u systému GPS. Provádí údržbu atomových hodin, povely družicím, manévry družic a jejich monitoring je uložen v navigační zprávě. Veškerý pozemní segment se nachází v Rusku, což je značná nevýhoda. Tím je určování polohy a časové omezení dostupné převážně jen pro ruský stát. [7]

1.4. GALILEO

Jedná se o navigační systém, který je v současnosti vytvářen pod záštitou Evropské unie. Původně však měl být financován soukromými investory. Tento systém je tvořen jako Globální navigační satelitní systém (GNSS), který by měl zbavit uživatele závislosti na systému GPS.

Kosmický segment bude tvořen 30 družicemi ve třech oběžných drahách. Družice budou vůči sobě posunuty o 120° a budou obíhat ve výšce přibližně 23 000 km. Se sklonem 56° k rovníku Země bude na každé dráze aktivních 9 družic a jedna bude vždy záložní. Pásmo přenášených signálů jsou 1,191 GHz pro signál E5, 1,278 GHz pro signál E6 a 1,575 GHz pro signál L1. Systém GALILEO, stejně tak jako GPS, používá kódové dělení CDMA. GALILEO má pět referenčních služeb OS, CS, SOL, PRS, SAR obsažených ve všech pásmech. V pásmu E5a jsou služby OS, CS, v pásmu E5b jsou OS, SoL, CS, v pásmu E6 jsou OS, PRS a v pásmu E1 OS, CS, SOL a PRS (viz *Obr. 4*). [4] [16]



Obr. 4 - GALILEO pásma a jejich služby (převzato z [18])



1.4.1. OS – Open Service

Tato služba je pro poskytování informací o poloze, rychlosti a času, která nebude zpoplatněna uživateli. Nejvíce bude vhodná pro automobilovou a mobilní navigaci. OS bude vysíláno na třech frekvencích E5a, E5b a E1, čímž bude dosaženo velké přesnosti při určování polohy. [15]

1.4.2. CS – Commercial Service

CS bude sloužit pro placící uživatele, kteří budou využívat služeb jako je předpověď počasí, informace o dopravě a dopravních nehodách a aktualizace map v navigačních přijímačích. Servis bude vysílán na všech frekvencích E5a, E5b, E6 i E1. [15]

1.4.3. SOL – Safety Of Life service

SOL služba je určena pro námořní, letecké a železniční druhy dopravy, které vyžadují přesnější navigaci a vyšší úroveň bezpečnosti. SOL bude nekódovaný a bude mít možnost ověřit signály, které jsou vysílány. Podporuje kritické operace jako např. přesnost letecké dopravy s vertikálním vedením. Dále je možnost uplinku pro GCC⁸, kde bude prováděno monitorování regionální integrity dat. Tento servis bude vysílán na frekvencích E5b a E1. [15]

1.4.4. PRS – Public Regulated Service

Služba PRS je vyhrazena jen pro uživatele autorizované vládou, kteří potřebují větší přesnost a odolnost vůči rušení. Signály této služby budou zakódovány a zpřístupněny uživatelům s autorizačním klíčem. Také bude tato služba zajišťovat přenos alarmů z tísňových majáků na SAR⁹ organizaci. [15]

⁸ GALILEO Control Centers

⁹ Search and Rescue – služba nouzové lokalizace služby COSPAS/SARSAT



2. Teoretický rozbor

2.1. Teoretická východiska

Pro lepší orientaci v následujícím textu je zde uvedeno několik základních pojmů a vlastností signálů používaných v navigačních systémech.

2.1.1. Fourierova transformace

Transformace převádí signál z časové oblasti do oblasti frekvenční pomocí funkce komplexních exponenciál.

Ve spojitém čase:

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt \quad (1.4)$$

V diskretním čase

$$S(\Omega) = \sum_{k=-\infty}^{\infty} s[k] e^{-j\Omega k} \quad (1.5)$$

2.1.2. Analytický signál

Je signál, jehož záporné kmitočty jsou přes Fourierovu transformaci rovny nule. Spektrum analytického signálu je stejné jako u pásmového signálu jen jednostranné. Nejčastěji se používá část spektra v kladné části charakteristiky. [21]

2.1.3. Pásmový signál

Pásmový signál je reálný signál, respektive je to tvar vř modulovaného signálu, jehož kmitočty je popsán rovnicí (1.6). Spektrum pásmového signálu je soustředěno okolo nosného kmitočtu.

$$\omega_c \gg 2\pi B, \quad (1.6)$$

kde B je šířka pásma signálu a spočte se dle rovnice (1.7):

$$B = (\omega_h - \omega_l) \frac{1}{2\pi}, \quad (1.7)$$



kde ω_h a ω_l jsou krajní kmitočty pásma. Mimo tyto dva kmitočty je spektrum nulové. Spektrum pásmového signálu je oboustranné a symetrické. [21]

2.1.4. Komplexní obálka

Komplexní obálka je pásmový signál, který je vynásoben exponenciálou uvedenou v rovnici (1.8), kde tato exponenciála odpovídá frekvenčnímu posunutí, které je dáno vlastností Fourierovy transformace. [21]

$$s_o(t) = s_p(t)e^{-j2\pi f_c t}, \quad (1.8)$$

2.1.5. Autokorelační funkce

Funkce, která vyjadřuje závislost dvou hodnot téhož signálu vzájemně posunutých o τ . Vyjadřuje se pro signály [21]:

Spojité:

Energetické:

$$R(\tau) = \int_{-\infty}^{\infty} s(t+\tau)s^*(t)dt \quad (1.9)$$

Výkonové:

$$R(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T s(t+\tau)s^*(t)dt \quad (2.0)$$

Diskrétní:

Energetické:

$$R(m) = \sum_{k=-\infty}^{\infty} s(k+m)s^*(k) \quad (2.1)$$

Výkonové:

$$R(m) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{k=-N}^N s(k+m)s^*(k) \quad (2.2)$$

2.1.6. Spektrální výkonová hustota

Spektrální výkonová hustota vyjadřuje rozdělení výkonu signálu v závislosti na kmitočtu. Vyjadřuje se pro signály [21]:



Spojité:

$$C(\omega) = \lim_{T \rightarrow \infty} \frac{1}{2T} \left| \int_{-T}^T s(t) \exp(-j\omega t) dt \right|^2 \quad (2.3)$$

Diskrétní:

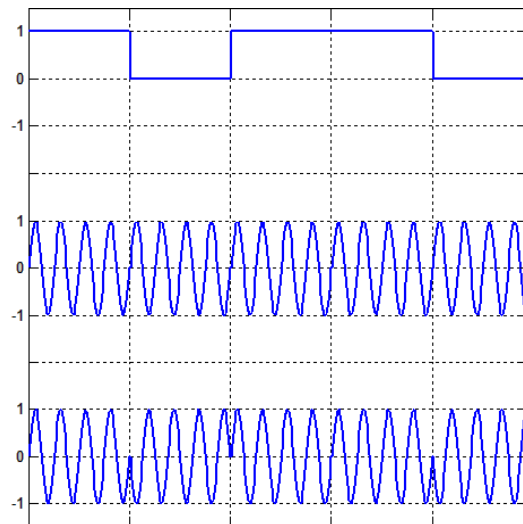
$$C(\Omega) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \left| \sum_{k=-N}^N s[k] \exp(-j\Omega k) \right|^2 \quad (2.4)$$

2.1.7. Použité modulace

Jako další vlastnosti jsou zde uvedeny některé modulace používané v navigační technice.

2.1.7.1. Modulace BPSK

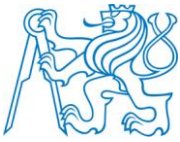
BPSK patří mezi digitální modulace. Je konstruována pomocí posunutí fáze harmonické nosné o 0° , nebo 180° . Nosná frekvence je namodulována na binární kód 0 a 1 (viz **Obr. 5**). Hodnota binárního kódu určuje fázi nosného signálu, buď je přenášena ve fázi, nebo o 180° otočená. Díky této modulaci dochází k namapování 0 a 1 na 1 a -1, nebo na -1 a 1.



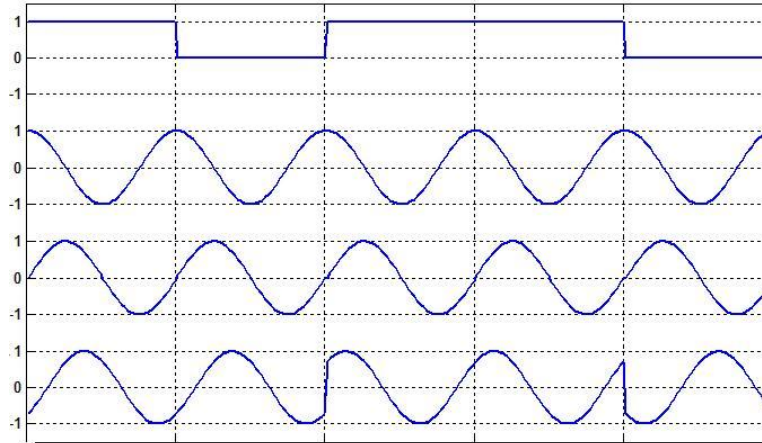
Obr. 5 - Modulace BPSK

2.1.7.2. Modulace QPSK

Quadrurní modulace je čtyřstavová, což zaručuje přenos dvou bitů najednou. Toho je docíleno díky rozdělení do dvou složek : kvadrurní (Q) a soufázové (I) (viz **Obr. 6**). Tyto složky



se dále násobí harmonickou nosnou pro Q složku ve fázi a pro I složku s fází otočenou o 90°. Nakonec jsou složky I a Q sečteny.



Obr. 6 - QPSK modulace

2.1.7.3. Modulace CBOC

CBOC modulace se skládá ze součtu BOC(1,1) a BOC(6,1), kde parametry BOC(m,n) jsou dány $m=f_{sc}/1,023$ a $n=f_c/1,023$. Celková modulace je řešena pomocí jedné periody signálu uvedeno v **Tabulka 2**, kde $\alpha = \sqrt{\frac{10}{11}}$ a $\beta = \sqrt{\frac{1}{11}}$. Tato posloupnost jedné periody se pronásobí jedním čipem určitého signálu.

Tabulka 2 - Jedna perioda BOC modulace (převzato z [3])

1–6 bitů první per.	6–12 bitů první per.	1–6 bitů druhé per.	6–12 bitů druhé per.
$\alpha+\beta$	$-\alpha+\beta$	$\alpha-\beta$	$-\alpha-\beta$
$\alpha-\beta$	$-\alpha-\beta$	$\alpha+\beta$	$-\alpha+\beta$
$\alpha+\beta$	$-\alpha+\beta$	$\alpha-\beta$	$-\alpha-\beta$
$\alpha-\beta$	$-\alpha-\beta$	$\alpha+\beta$	$-\alpha+\beta$
$\alpha+\beta$	$-\alpha+\beta$	$\alpha-\beta$	$-\alpha-\beta$
$\alpha-\beta$	$-\alpha-\beta$	$\alpha+\beta$	$-\alpha+\beta$

2.2. Signál GPS

Každá družice vysílá na pěti pásmech označených jako L_1 , L_1C , L_2 , L_2C a L_5 , kde L_1 a L_1C se vysílají na frekvenci 1 575,42 MHz, L_2 a L_2C na 1 227,6 MHz a L_5 na 1 176,45 MHz. Tyto



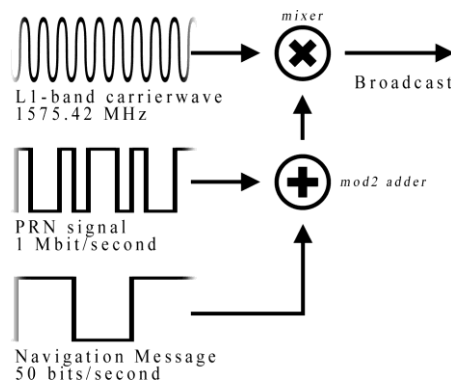
frekvence jsou odvozeny z násobků $f_0=10,23$ MHz, tj. pro pásmo $L_1=154f_0$, pro pásmo $L_2=120f_0$ a $L_5=115f_0$.

2.2.1. L1 C/A

Signál C/A na L1 dá popsat rovnicí (1.4), kde A_c je amplituda nosné frekvence. $C(t)$ je C/A¹⁰ – kód, který je určen pro civilní uživatele a je násoben $D(t)$, čili datovou zprávou.

$$s(t) = A_c C(t)D(t)\sin(2\pi L_1 t), \quad (1.4)$$

C/A – kód je násoben PRN¹¹ kódem, který zaručí vysílání na stejných frekvencích pro všechny družice. Celkový signál je modulován BPSK¹² modulací (viz Obr. 7). [2] [5] [6] [10]



Obr. 7 - Syntéza GPS L1 C/A

2.2.1.1. C/A – kód

Jedná se o kód dlouhý 1 ms s periodou 1 023 bitů (čipů), tedy jeho bitová rychlost je 1,023 Mb/s, což odpovídá:

$$\frac{f_0}{10} = \frac{10,23}{10} \quad (1.5)$$

Délka jednoho čipu je necelá 1 μ s. Pseudonáhodná posloupnost tohoto kódu je tvořena pomocí Goldova kódu s hodnotami 0 a 1. Modulace je zde použita DSSS(1)¹³. Generování tohoto kódu je tvořeno přes dva deseti-bitové posuvné registry. [2] [5] [6] [10]

¹⁰ Coarse Acquisition.

¹¹ Pseudo-random noise sequence – pseudonáhodná posloupnost.

¹² Binary Phase Shift Keying – Modulace s binárním fázovým klíčováním.

¹³ Direct Sequence Spread Spectrum



U prvního registru jsou vybrány aktuální hodnoty z pozic 3 a 10 a přes funkci modulo 2^{14} sečteny. Druhý registr má vybrány aktuální hodnoty z pozic 2, 3, 6, 8 a 9, které jsou taktéž sečteny přes modulo 2. Výsledky z obou registrů jsou posunuty na první pozici a přetečený bit je výsledný kód generátorů. Na počátku jsou oba registry naplněny všemi jedničkami. Celý postup je vyobrazen na *Obr. 8*, kde je selektor popsán v kapitole PRN – kód. Výsledek selektoru je poté sečten s výsledkem prvního posuvného registru, taktéž pomocí modulo 2. Tento součet dává výsledný C/A – kód. Pokud je výsledek správný, tak by mělo prvních deset bitů tohoto kódu souhlasit s hodnotou uvedenou v *Tabulka 3* v hexadecimální soustavě. [2] [5] [6] [10]

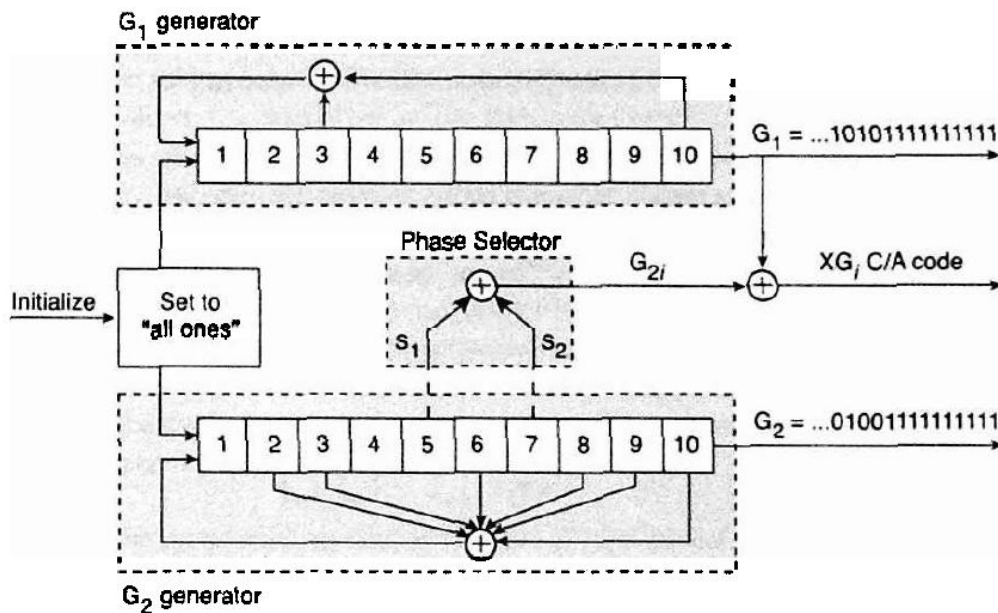
2.2.1.2. PRN – kód

Z druhého posuvného registru jsou vybrány určité aktuální hodnoty pozic, které jsou dány PRN-kódem určité družice (viz *Tabulka 3*). Aktuální hodnoty jsou v selektoru sečteny modulem 2.

Tabulka 3 - ID družic a výběr pozic selektoru (převzato z [2])

PRN	Selector	10 chipů (hex)	PRN	Selector	10 chipů (hex)
1	$2 \oplus 6$	1440	17	$1 \oplus 4$	1156
2	$3 \oplus 7$	1620	18	$2 \oplus 5$	1467
3	$4 \oplus 8$	1710	19	$3 \oplus 6$	1633
4	$5 \oplus 9$	1744	20	$4 \oplus 7$	1715
5	$1 \oplus 9$	1133	21	$5 \oplus 8$	1746
6	$2 \oplus 10$	1455	22	$6 \oplus 9$	1763
7	$1 \oplus 8$	1131	23	$1 \oplus 3$	1063
8	$2 \oplus 9$	1454	24	$4 \oplus 6$	1706
9	$3 \oplus 10$	1626	25	$5 \oplus 7$	1743
10	$2 \oplus 3$	1504	26	$6 \oplus 8$	1761
11	$3 \oplus 4$	1642	27	$7 \oplus 9$	1770
12	$5 \oplus 6$	1750	28	$8 \oplus 10$	1774
13	$6 \oplus 7$	1764	29	$1 \oplus 6$	1127
14	$7 \oplus 8$	1772	30	$2 \oplus 7$	1453
15	$8 \oplus 9$	1775	31	$3 \oplus 8$	1625
16	$9 \oplus 10$	1776	32	$4 \oplus 9$	1712

¹⁴ Realizována funkcí XOR $\Rightarrow 1 \oplus 1 = 0 = 0 \oplus 0, 1 \oplus 0 = 1 = 0 \oplus 1$



Obr. 8 - Generátor C/A a PRN kódu (převzato z [2])

2.2.2. L2

Signál je určen především pro armádní použití. Proto popis jeho struktury není v této práci možné zahrnout. L2 obsahuje P(Y) – kód, pro jehož zpřístupnění je zapotřebí autorizační klíč.

2.2.2.1. P(Y) – kód

Tento kód je pouze pro autorizované uživatele, kterým zajistí větší přesnost určování polohy. Kódování není známo, jelikož je využitelné pouze pro armádu Spojených států amerických a dále pro některé státy NATO. [2] [5] [6] [10]

2.2.2.2. Anti-spoofing (AS)

AS je šifrovaný P-kód, zvaný Y-kód. Kód byl vytvořen pro zhoršování kvality signálu se záměrem potlačit AS protivníka. Anti-Spoofing působí od roku 1994 a k získání přístupu Y-kódu je potřeba šifrovací klíč. Ten je jen pro oprávněné (autorizované) uživatele. Pro příjem je potřeba použít přijímač s možností přijímat dvě frekvence najednou. P(Y) – kód se vysílá na obou frekvencích L₁, L₂ a díky tomuto příjmu je možné přesnější určení polohy, než je tomu u kódu C/A. [2] [5] [6] [10] [11]



2.2.2.3. Selective Availability (SA)

SA je určen pro degradaci signálu za pomoci kontrolované chyby v měření. Tato chyba představovala asi pětinasobné zvýšení polohové chyby než chyby tkvící v systému. Docílení chyby je zapříčiněno záměrným rozladěním hodin satelitu. To má za následek ovlivnění C/A – kódu, P(Y) – kódu a měření nosné frekvence. Aktivaci a deaktivaci SA mohl provést pouze autorizovaný uživatel. Poprvé byl použit v roce 1990 v rámci politiky. Pravděpodobně díky rozvoji systému Galileo bylo SA v roce 2000 deaktivováno. [2] [5] [6] [10] [11]

2.2.3. L1 C

Tento signál je modernizací GPS navigačního systému spolu s L2C a L5. L1C má být vysílán na frekvenci $L1=1\ 575,42$ MHz. Jelikož podle zdroje [5] je spuštění tohoto signálu plánováno na rok 2014, tak jsou možnosti popisu struktury v této práci značně omezené. [5] [6] [14]

2.2.3.1. PRN – kód

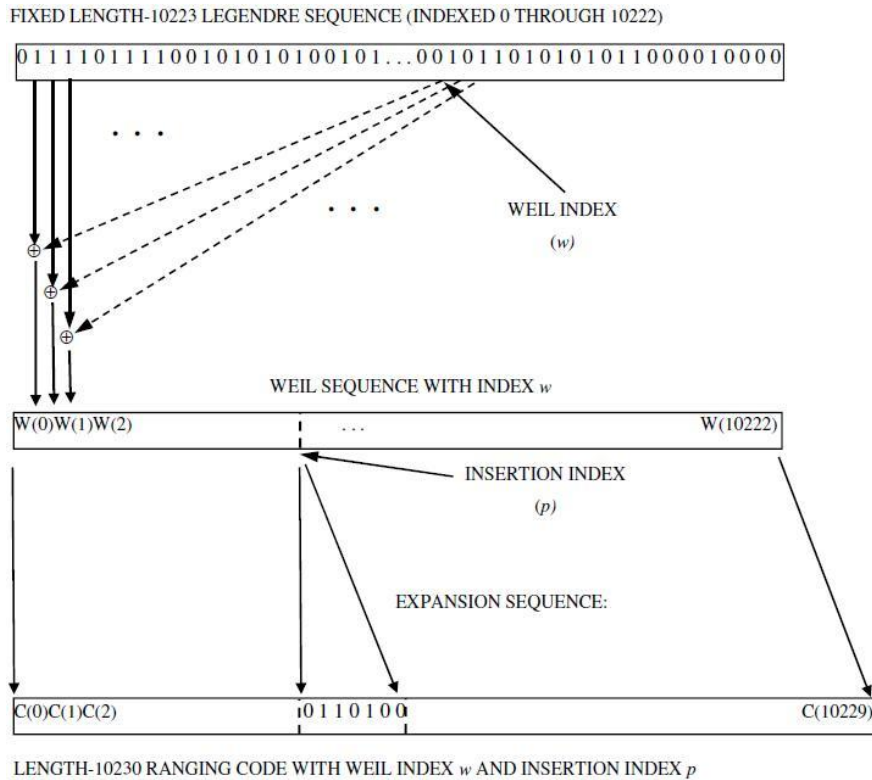
PRN kód se vytváří pomocí tří různých kódů, a to $L1C_P$, $L1C_D$ a $L1C_O$. $L1C_P$ a $L1C_D$ kódy se vytvářejí stejným způsobem. Konkrétní bity kódů na pozici označované jako weil index jsou násobeny pomocí funkce XOR s konkrétními bity déhož kódu. Tento kód je poté na určitém místě, označovaném jako insertion index (viz *Tabulka 4*), rozšířen o pevně danou sekvenci 0110100 zvanou jako expansion sequence (viz *Obr. 9*). Tím tyto kódy získávají konečnou délku 10 230 bitů trvající 10 ms s čipovou rychlostí 1,023Mbps. Hodnoty weil index, insertion index a inicializace 24 čipů, podle kterých je generována posloupnost 10 223 bitů jsou vybírány podle určitého satelitu uvedeného v *Tabulka 4*. [5] [6] [14]

Kód $L1C_O$ je dlouhý 1 800 bitů a trvá 18 ms se stejnou čipovou rychlostí jako u dvou předchozích kódů. Jeho generace je dána jedenácti bitovým posuvným registrem, kde je inicializována počáteční hodnota (viz *Tabulka 5*) podle určité družice. Posuvný registr je konstruován pomocí funkce XOR, jejíž výsledek je dán na první pozici registru a zůstatkové hodnoty se posunou směrem doprava. Vstupy do této funkce XOR jsou dány pomocí koeficientů (viz *Tabulka 5*), které jsou určeny taktéž podle určité družice. [5] [6] [14]



Tabulka 4 - Hodnoty pro PRN L1C_P a L1C_D (převzato z [14])

PRN	Weil in- dex L1C_P	Insertion index L1C_P	Initial chips (octal)	Weil in- dex L1C_D	Insertion index L1C_D	Initial chips (octal)
1	5111	412	05752067	5097	181	52231646
2	5109	161	70146401	5110	359	46703351
3	5108	1	32066222	5079	72	00145161
4	5106	303	72125121	4403	1110	11261273
5	5103	207	42323273	4121	1480	71364603
6	5101	4971	01650642	5043	5034	55012662
7	5100	4496	21303446	5042	4622	30373701
8	5098	5	35504263	5104	1	07706523
9	5095	4557	66434311	4940	4547	71741157
10	5094	485	52631623	5035	826	42347523
11	5093	253	04733076	4372	6284	12746122
12	5091	4676	50352603	5064	4195	34634113
13	5090	1	32026612	5084	368	47555063
14	5081	66	07476042	5048	1	01221116
15	5080	4485	22210746	4950	4796	37125437
16	5069	282	30706376	5019	523	32203664
17	5068	193	75764610	5076	151	62162634
18	5054	5211	73202225	3736	713	35012616
19	5044	729	47227426	4993	9850	00437232
20	5027	4848	16064126	5060	5734	32130365
21	5026	982	66415734	5061	34	51515733
22	5014	5955	27600270	5096	6142	16027175
23	5004	9805	66101627	4983	190	26267340
24	4980	670	17717055	4783	644	36272365
25	4915	464	47500232	4991	467	67707677
26	4909	29	52057615	4815	5384	07760374
27	4893	429	76153566	4443	801	73633310
28	4885	394	22444670	4769	594	30401257
29	4832	616	62330044	4879	4450	72606251
30	4824	9457	13674337	4894	9437	37370402
31	4591	4429	60635146	4985	4307	74255661
32	3706	4771	73527653	5056	5906	10171147



LENGTH-10230 RANGING CODE WITH WEIL INDEX w AND INSERTION INDEX p

Obr. 9 - Tvorba posloupnosti L1Cd a L1Cp (převzato z [14])

Tabulka 5 - Hodnoty pro PRN L1C₀ (převzato z [14])

PRN	Index reg.	Initial (octal)	PRN	Index reg.	Initial (octal)
1	5111	3266	17	7665	2761
2	5431	2040	18	6325	3750
3	5501	1527	19	4365	2701
4	5403	3307	20	4745	1206
5	6417	3756	21	7633	1544
6	6141	3026	22	6747	1774
7	6351	0526	23	4475	0546
8	6501	0420	24	4225	2213
9	6205	3415	25	7063	3707
10	6235	0337	26	4423	2051
11	7751	0265	27	6651	3650
12	6623	1230	28	4161	1777
13	6733	2204	29	7237	3203
14	7627	1440	30	4473	1762
15	5667	2412	31	5477	2100
16	5051	3516	32	6163	0571



2.2.4. L2 C

Signál je vysílán na frekvenci 1 227,6 MHz. L2C je další z modernizovaných signálů systému GPS, který je určen pro civilní uživatele. Použitá modulace v tomto systému je BPSK. [5] [6] [12]

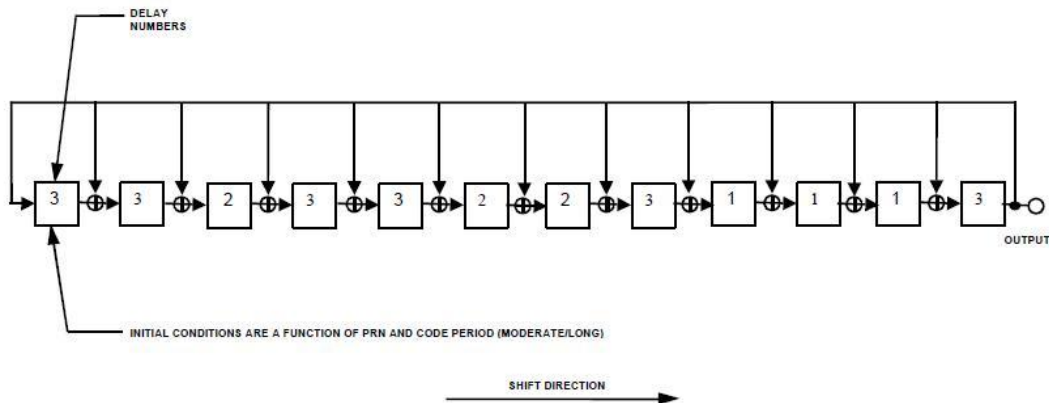
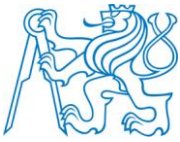
2.2.4.1. PRN – kód

Kód je tvořen dvěma kódy CL a CM, které jsou generovány s čipovou rychlostí 511,5 kchip/s. První kód CM je dlouhý 10 230 bitů trvající 20 ms. Tento kód je generován 24 bitovým posuvným registrem (viz *Obr. 10*), jehož počáteční inicializace je dána číslem určité družice určeném v *Tabulka 6*. Celá posloupnost 10 230 bitů je pomocí funkce XOR násobena jedním bitem navigační zprávy. [5] [6] [12]

Druhý kód CL je extrémně dlouhý. Trvá 1,5 s a obsahuje 767 250 čipů. Je tvořen stejně jako CM kód posuvným registrem stejného typu, kde je dána počáteční inicializace (viz *Tabulka 6*). Do délky tohoto kódu se 75x opakuje posloupnost CM kódu. Pomocí multiplexování se poté skládá čip po čipu CL a CM kód, čímž je získána čipová rychlost celkového L2C kódu 1,023 Mchip/s.

Tabulka 6 - Hodnoty pro PRN L2C (převzato z [12])

PRN	Initial L2CM	Initial L2CL	PRN	Initial L2CM	Initial L2CL
1	742417664	742417664	17	742417664	742417664
2	756014035	756014035	18	756014035	756014035
3	002747144	002747144	19	002747144	002747144
4	066265724	066265724	20	266527765	266527765
5	601403471	601403471	21	006760703	006760703
6	703232733	703232733	22	501474556	501474556
7	124510070	124510070	23	743747443	743747443
8	617316361	617316361	24	615534726	615534726
9	047541621	047541621	25	763621420	763621420
10	733031046	733031046	26	720727474	720727474
11	713512145	713512145	27	700521043	700521043
12	024437606	024437606	28	222567263	222567263
13	021264003	021264003	29	132765304	132765304
14	230655351	230655351	30	746332245	746332245
15	001314400	001314400	31	102300466	102300466
16	222021506	222021506	32	255231716	255231716



Obr. 10 - Registr pro tvorbu CM a CL kódu (převzato z [12])

2.2.5. L5

Poslední z modernizovaných signálů GPS systému je pásmo L5, které je vytvořeno také pro civilní použití a je vysíláno na frekvenci 1 176,45 MHz. V tomto pásmu je oproti L2C a L1 C/A použita modulace QPSK. [5] [6] [13]

2.2.5.1. PRN – kód

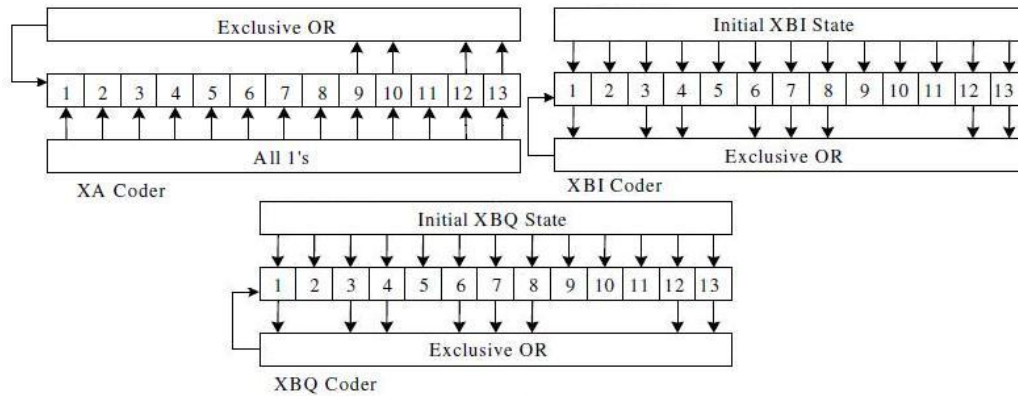
Celý kód je tvořen pomocí tří různých kódů XA, XBI a XBQ. Všechny tyto tři kódy jsou generovány pomocí 13 bitového posuvného registru (viz **Obr. 11**). U kódu XA jsou na počátku registru inicializovány samé jedničky. Délka tohoto kódu je 8 190 čipů s rychlostí 10,23 Mbps. Tato posloupnost se opakuje do délky 10 230 bitů, která trvá 1 ms. [5] [6] [13]

Další dva kódy XBI a XBQ jsou dlouhé 8 191 čipů o stejné rychlosti 10,23 Mbps. Oba kódy se stejně jako u XA kódu opakují na délku 10 230 bitů trvajících 1ms. Jejich počáteční inicializace je dána číslem vysílané družice viz **Tabulka 7**. Každý z obou kódů je přes funkci XOR pronásoben s kódem XA. Výsledné kódy XQ a XI jsou následně pronásobeny stejnou funkcí s navigační zprávou. U XI je navigační zpráva pronásobena Neuman-Hofmanovým synchronizačním kódem 0000110101, který trvá 10 ms. Navigační zpráva u kódu XQ je pronásobena Neuman-Hofmanovým synchronizačním kódem 00000100110101001110, trvajících 20 ms. [5] [6] [13]



Tabulka 7 - Hodnoty pro PRN L5 (převzato z [13])

PRN	Index XB – I5	Index XB – Q5	Initial XB – I5	Initial XB – Q5
1	266	1701	0101011100100	1001011001100
2	365	323	1100000110101	0100011110110
3	804	5292	0100000001000	1111000100011
4	1138	2020	1011000100110	0011101101010
5	1509	5429	1110111010111	0011110110010
6	1559	7136	0110011111010	0101010101001
7	1756	1041	1010010011111	1111110000001
8	2084	5947	1011110100100	0110101101000
9	2170	4315	1111100101011	1011101000011
10	2303	148	0111111011110	0010010000110
11	2527	535	0000100111010	0001000000101
12	2687	1939	1110011111001	0101011000101
13	2930	5206	0001110011100	0100110100101
14	3471	5910	0100000100111	1010000111111
15	3940	3595	0110101011010	1011110001111
16	4132	5135	0001111001001	1101001011111
17	4332	6082	0100110001111	1110011001000
18	4924	6990	1111000011110	1011011100100
19	5343	3546	1100100011111	0011001011011
20	5443	1523	0110101101101	1100001110001
21	5641	4548	0010000001000	0110110010000
22	5816	4484	1110111101111	0010110001110
23	5898	1893	1000011111110	1000101111101
24	5918	3961	1100010110100	0110111110011
25	5955	7106	1101001101101	0100010011011
26	6243	5299	1010110010110	0101010111100
27	6345	4660	0101011011110	1000011111010
28	6477	276	0111101010110	1111101000010
29	6518	4389	0101111100001	0101000100100
30	6875	3783	1000010110111	1000001111001
31	7168	1591	0001010011110	0101111100101
32	7187	1601	0000010111001	1001000101010



Obr. 11 - Registry pro kódy XA, XBI a XBQ (převzato z [13])

2.3. Signál GLONASS

Struktura tohoto navigačního systému je dosti podobná struktuře GPS systému. Hlavním rozdílem je FDMA, oproti CDMA u GPS. Tím má každá družice tři odlišné frekvence na nosných $f_{01} = 1\,602$ MHz, $f_{02} = 1\,246$ MHz a $f_{03} = 1\,201$ MHz s $\Delta f_1 = 562,5$ kHz a $\Delta f_2 = 437,5$ kHz. Výchozí frekvence je dána rovnicemi (1.4) a (1.5), kde K udává číslo družice, které je v rozmezí od -7 do 13. U připravovaného signálu L3 je použito naopak dělení CDMA s nosnou frekvencí 1 202, 025 MHz. [18] [15]

$$f_{K1} = f_{01} + K\Delta f_1 \quad (1.4)$$

$$f_{K2} = f_{02} + K\Delta f_2, \quad (1.5)$$

2.3.1. L1

2.3.1.1. PRN

PRN kód je tvořen pomocí devíti bitového posuvného registru. Na páté a deváté pozici tohoto registru je určena následující první pozice pomocí funkce XOR. Celá délka tohoto kódu je dlouhá 511 bitů a trvá 1 ms, při čipové rychlosti 0,511 MHz. Dvacetinásobné opakování PRN kódu se pronásobí s jedním bitem navigační zprávy trvajícím 20 ms. [9] [15]

2.3.2. L2

Tento signál je určen pro civilní uživatele. L2 má stejnou strukturu signálu jako L1, ale je vysílán bez navigační zprávy.



2.3.2.1. PRN

Je tvořen pomocí 25 bitového posuvného registru. Délka kódu je 33 554 432 bitů trvajících 1 sekundu. Čipová rychlost těchto bitů je 5,11 Mchips/s. [9]

2.4. Signál GALILEO

Signál je složen z tří základních pásem E5, E1 a E6. Tato pásma jsou vysílána ve formě pilotních a datových signálů. S ohledem na zadání této práce budou popsány pouze datové části pásem, protože pilotní části signálů jsou určeny pro autorizované uživatele.

2.4.1. E5

Signál je přenášen na nosné frekvenci 1 191,75 MHz a je rozdělen na pásma E5a a E5b. Frekvence těchto pásem je 1 176,45 MHz pro E5a a 1 207,14 MHz pro E5b. Obě pásma jsou modulována AltBOC modulací. [15] [18]

2.4.1.1. E5a

Kód E5a je rozděleno na dvě složky, soufázovou E5a-I a kvadraturní E5a-Q. Soufázová a kvadraturní složka je tvořena pomocí primárního a sekundárního kódu. Primární kód soufázové a kvadraturní složky má 10 230 bitů trvajících 1 μ s. Tento kód je tvořen pomocí dvou 14-ti bitových posuvných registrů. Každý z těchto registrů má určenou hodnotu indexů (viz *Tabulka 10*), kde jejich hodnoty jsou pronásobeny přes funkci XOR. Výsledek tohoto pronásobení je vložen na první index bitového registru a počáteční hodnoty se posunou o jednu pozici doprava. Hodnoty posuvného registru jsou inicializovány pomocí čísla satelitu uvedeného v *Tabulka 8* a *Tabulka 9*. Kódy z obou posuvných registrů jsou nakonec mezi sebou pronásobeny také funkcí XOR. [3] [15]

Sekundární kód je u soufázové složky dlouhý 20 bitů s periodou 20 ms a u kvadraturní je dlouhý 100 bitů trvajících 100 ms. Tentokrát je číselná hodnota tohoto kódu pevně daná. Čipová rychlost sekundárního kódu je stejně jako u kódu primárního 10,23MHz. [3] [15]



Tabulka 8 - Hodnoty pro registr 2 E5a-I (převzato z [3])

PRN	Index registr 2	Initial registr 2	PRN	Index registr 2	Initial registr 2
1	30305	3CEA9D	17	24252	5D55CE
2	14234	9D8CF1	18	11631	B19B7C
3	27213	45D1C8	19	24776	5805FC
4	20577	7A0133	20	00630	F99EA1
5	23312	64D423	21	11560	B23CE5
6	33463	23300D	22	17272	8515E8
7	15614	91CEF2	23	27445	436822
8	12537	AA82DC	24	31702	30F77B
9	01527	F2A17D	25	13012	A7D629
10	30236	3D84AE	26	14401	9BFAC7
11	27344	446D38	27	34727	18A25B
12	07272	C514F2	28	22627	69A39F
13	36377	0C0184	29	30623	39B27D
14	17046	8767E0	30	27256	454598
15	06434	CB8EFF	31	01520	F2BC62
16	15405	93EBCD	32	14211	9DDBC6

Tabulka 9 - Hodnoty pro registr 2 E5a-Q (převzato z [3])

PRN	Index registr 2	Initial registr 2	PRN	Index registr 2	Initial registr 2
1	25652	515537	17	02265	ED28B3
2	05142	D67539	18	06430	CB9F5B
3	24723	58B2E5	19	25046	576592
4	31751	305914	20	12735	A88811
5	27366	442710	21	04264	DD3649
6	24660	593CF8	22	11230	B59F42
7	33655	214AD7	23	00037	FF81F6
8	27450	435EA6	24	06137	CE8128
9	07626	C1A7D5	25	04312	DCD55C
10	01705	FOE94A	26	20606	79E450
11	12717	A8C239	27	11162	B63460
12	32122	2EB63B	28	22252	6D562B
13	16075	8F0A46	29	30533	3A9010
14	16644	896DD4	30	24614	59CD72
15	37556	0245F1	31	07767	C0211A
16	02477	EB0160	32	32705	28EB96



2.4.1.2. E5b

U pásma E5b je stejně jako u E5a obsaženo dvou složek, soufázové a kvadraturní. Tyto složky jsou indikovány stejnými posuvnými registry, jen s jinými hodnotami indexů pro pronásobení dle *Tabulka 10* a inicializací druhého registru dle *Tabulka 11* a *Tabulka 12*. U sekundárního kódu je situace obdobná s tím, že u soufázové složky je délka kódu 4 bity s periodou 4ms a u kvadraturní je délka 100 bytů trvajících 100 ms. Hodnoty sekundárního kódu jsou také pevně dány jako u pásma E5a. [3] [15]

Tabulka 10 - Indexy primárních kódů E5 (převzato z [3])

Pásma	Index registr 1	Index registr 2
E5a – I	40503	50661
E5a – Q	40503	50661
E5b – I	64021	51445
E5b – Q	64021	43143

Tabulka 11 - Hodnoty pro registr 2 E5b-I (převzato z [3])

PRN	Index registr 2	Initial registr 2	PRN	Index registr 2	Initial registr 2
1	07220	C5BEA1	17	32231	2D9BC6
2	26047	4F6248	18	10353	BC5146
3	00252	FD5488	19	00755	F848B0
4	17166	86277B	20	26077	4F01E8
5	14161	9E39D5	21	11644	B16C9B
6	02540	EA7EDE	22	11537	B2827D
7	01537	F28321	23	35115	16C809
8	26023	4FB0C9	24	20452	7B570F
9	01725	F0AB64	25	34645	1969C0
10	20637	79833B	26	25664	512FA9
11	02364	EC2D91	27	21403	73F36
12	27731	409B11	28	32253	2D5317
13	30640	397E16	29	02337	EC8390
14	34174	1E0FCD	30	30777	380374
15	06464	CB2F5A	31	27122	46B4DE
16	07676	C1079A	32	22377	6C01D9



Tabulka 12 - Hodnoty pro registr 2 E5b-Q (převzato z [3])

PRN	Index registr 2	Initial registr 2	PRN	Index registr 2	Initial registr 2
1	03331	E49AF0	17	02456	EB443C
2	06143	CE701F	18	30013	3FD0B1
3	25322	54B709	19	00322	FCB7CF
4	23371	641AB1	20	10761	B83815
5	00413	FBD0AE	21	26767	48224A
6	36235	0D8BC9	22	36004	0FEE25
7	17750	805FA5	23	30713	38D33B
8	04745	D86BA0	24	07662	C135B9
9	13005	A7E921	25	21610	71DE13
10	37140	067E55	26	20134	7E8CFB
11	30155	3E4B58	27	11262	B536C3
12	20237	7D82FB	28	10706	B8E68C
13	03461	E33BC2	29	34143	1E7272
14	31662	31372C	30	11051	B75B69
15	27146	46676F	31	25460	533F65
16	05547	D2613E	32	17665	812B41

2.4.2. E1

Signál E1 je rozdělen na E1C a E1B kódy, které jsou modulovány CBOC modulací. Kód E1B je tvořen dvěma 13-ti bitovými posuvnými registry. První posuvný registr má inicializovanou počáteční hodnotu samých jedniček. Indexy posuvného registru jsou dány hodnotou 23261 v osmičkové soustavě. Této hodnotě odpovídá hodnota indexů 4, 5, 7, 9, 10 a 13, které jsou mezi sebou pronásobeny funkcí XOR a dány na první index registru. Tím je počáteční hodnota posunuta o jeden index doprava. Takto je tvořen i druhý posuvný registr, jehož indexová hodnota je 30741(o), což odpovídá hodnotám indexu 5, 6, 7, 8, 12 a 13. Jediný rozdíl oproti prvnímu registru je v počáteční inicializaci, která je dána PRN družicí viz **Tabulka 13**. Celkový kód je vytvořen pronásobením funkcí XOR obou posuvných registrů a je dlouhý 4 092 bitů s čipovou rychlostí 1 μ s. [1] [3] [15]

Kód E1C je tvořen podobným způsobem jako E1B. První a druhý posuvný registr má hodnoty indexů dány čísly 20033(o) a 23261(o). Těmto hodnotám odpovídají indexy posuvných registrů 1, 3, 4, 13 a 4, 5, 7, 9, 10, 13. U prvního registru je počáteční inicializace dána opět samými jedničkami a u druhého registru pomocí PRN družice z **Tabulka 13**. Pouze u druhého



registru je jeho výsledný kód pronásoben posloupností 11100000001010110110010000, která trvá 25 ms. Celkový kód je opět tvořen pronásobením obou registrů přes funkci XOR. [1] [3] [15]

Tabulka 13 - Inicializace registrů 2 E1C a E1B (převzato z [1])

PRN	E1C (hex)	E1B (hex)
1	15603	14603
2	11774	04277

2.4.3. E6

Tento kód se skládá také ze dvou kódů primárního a sekundárního. Zatímco sekundární kód je přesně dán, primární kód se nedá zjistit. To je zapříčiněno tím, že je signál určen pro autorizované uživatele. [3] [15]

2.5. Navigační zpráva

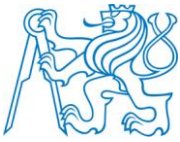
2.5.1. NAV

Celá navigační zpráva se skládá z 25 stránek (rámců). Každý z rámců má pět podrámců trvajících 6 s. Jeden tento podrámec obsahuje deset 30-ti bitových slov. Navigační zpráva trvá 12,5 min a má 37 500 bitů. Po uplynutí této doby se zpráva opakuje. U každého slova jsou paritní (kontrolní) bity, jimž náleží pozice 6-ti posledních bitů ve slově. Těchto šest bitů je zakódováno rozšířeným Hammingovým kódem¹⁵. Při zakódování se k 24 bitům přidávají dva poslední bity z předchozího kódového slova, protože při výpočtu šesti paritních bitů je potřebné mít 26 zdrojových bitů. [2] [5] [12]

Každý podrámec má dvě první slova označována jako TLM (telemetry word) a HOW (handover word). Prvních 8 bitů slova TLM určuje začátek podrámece. Následujících 14 bitů je pro diagnostickou zprávu řídicího segmentu. Zbývající dva bity 23 a 24 jsou rezervované. U druhého slova HOW je prvních 17 bitů vyhrazeno pro tzv. z-count, který určuje pořadí následujícího podrámece od půlnoci ze soboty na neděli. Následující dva bity 18 a 19 indikují nenulovou hodnotu, z nichž 18. bit sníženou přesností údajů družice a 19. bit režim AS¹⁶. V bitech 20, 21 a 22 je

¹⁵ Je možná až třináásobná chyba v kódovém slově, nebo lze opravit jednoduchou chybu.

¹⁶ Viz předchozí kapitola Anti-spoofing (AS)

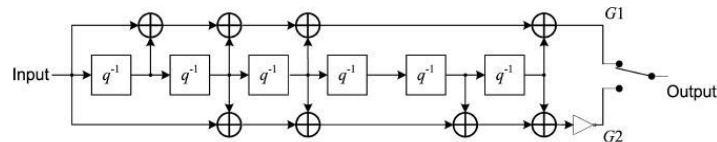


alokováno místo pro dekadické vyjádření čísla podrámece. Poslední dva bity 23 a 24 jsou pro zachování parity. [2] [5] [12]

První podrámece nese informaci o stavu družice, korekci atomových hodin a číslo týdne. Druhý a třetí podrámece je určen pro nesení informace tzv. efemeridy družice. Díky této informaci je možné určit polohu dané družice. Čtvrtý a pátý podrámece obsahuje almanach, který nese informaci o přibližné poloze družic a jejich zdravotním stavu. [2] [5] [12]

2.5.2. C/NAV

Stránky navigační zprávy NAV jsou kódovány pomocí konvolučního dekodéru uvedeného na *Obr. 12*. [12]



Obr. 12 - 1/2 konvoluční kodér (převzato z [3])

2.5.3. F/NAV

Každý rámeček navigační zprávy F/NAV obsahuje 12 podrámečků trvajících 50 s. Každý podrámeček je složen z pěti stránek trvajících 10 s. Stránka začíná 12-ti synchronizačními symboly 101101110000, které nejsou kódovány F/NAV. Každá stránka obsahuje dvě F/NAV slova. Prvních šest bitů tohoto slova obsahuje číslo stránky. Dalších 208 bitů nese navigační data zakončená 24-mi bity CRC, které jsou alokovány pro odhalení možných chyb a šesti bity posloupnosti nul, které označují konec F/NAV stránky. [3]

2.5.4. I/NAV

Rámce navigační zprávy I/NAV obsahují 24 podrámečků, které trvají 30 s. Každý z podrámečků může obsahovat jednu ze dvou možných stránek (nominální a upozorňující). Struktura obou stránek je tvořena synchronizačními bity 0101100000, dvěma podstránkami, jejichž obsah se liší dle vysílaného systému a jsou zakončeny 6-ti nulovými bity pro označení konce I/NAV stránky. [3]

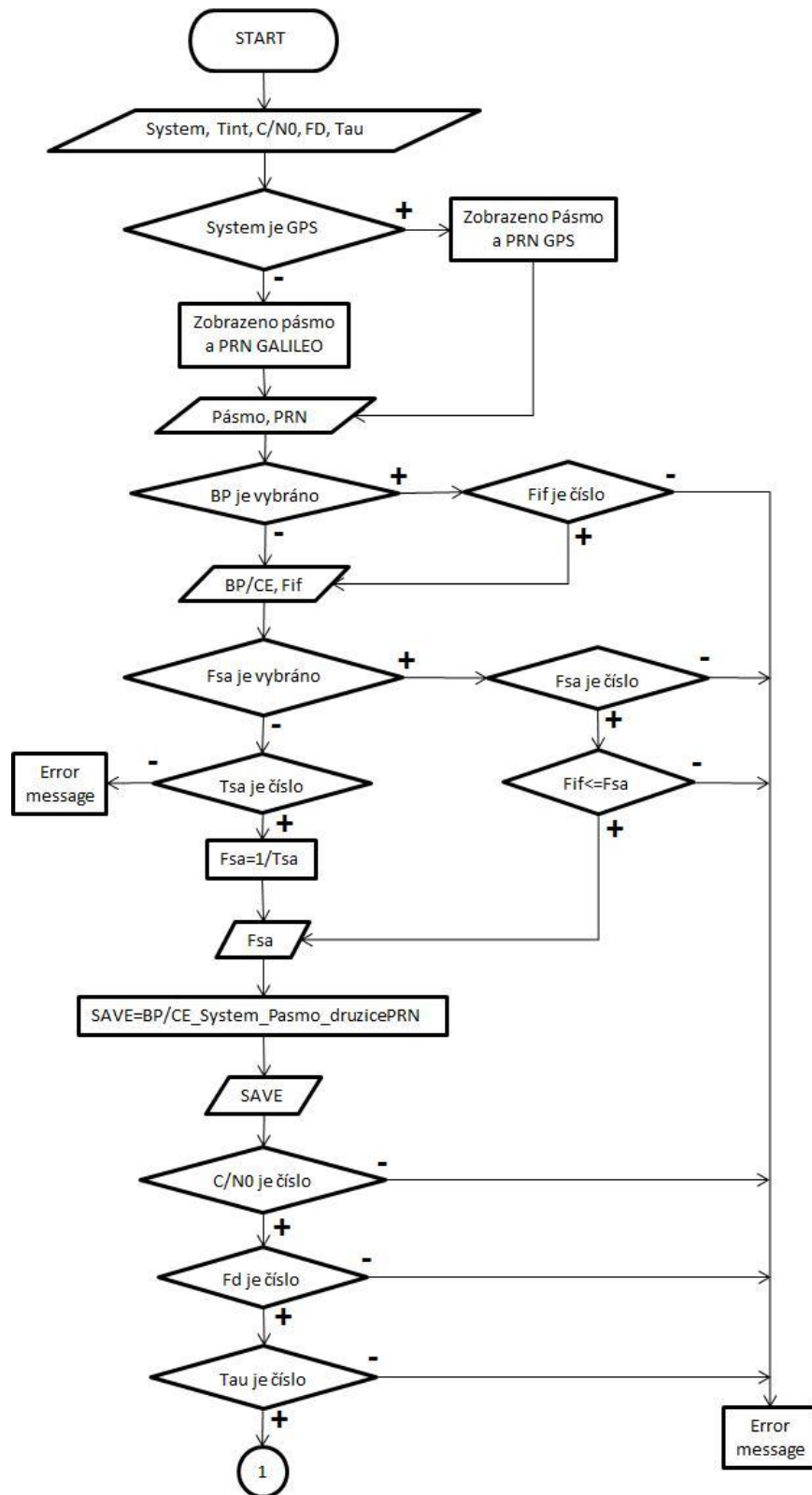


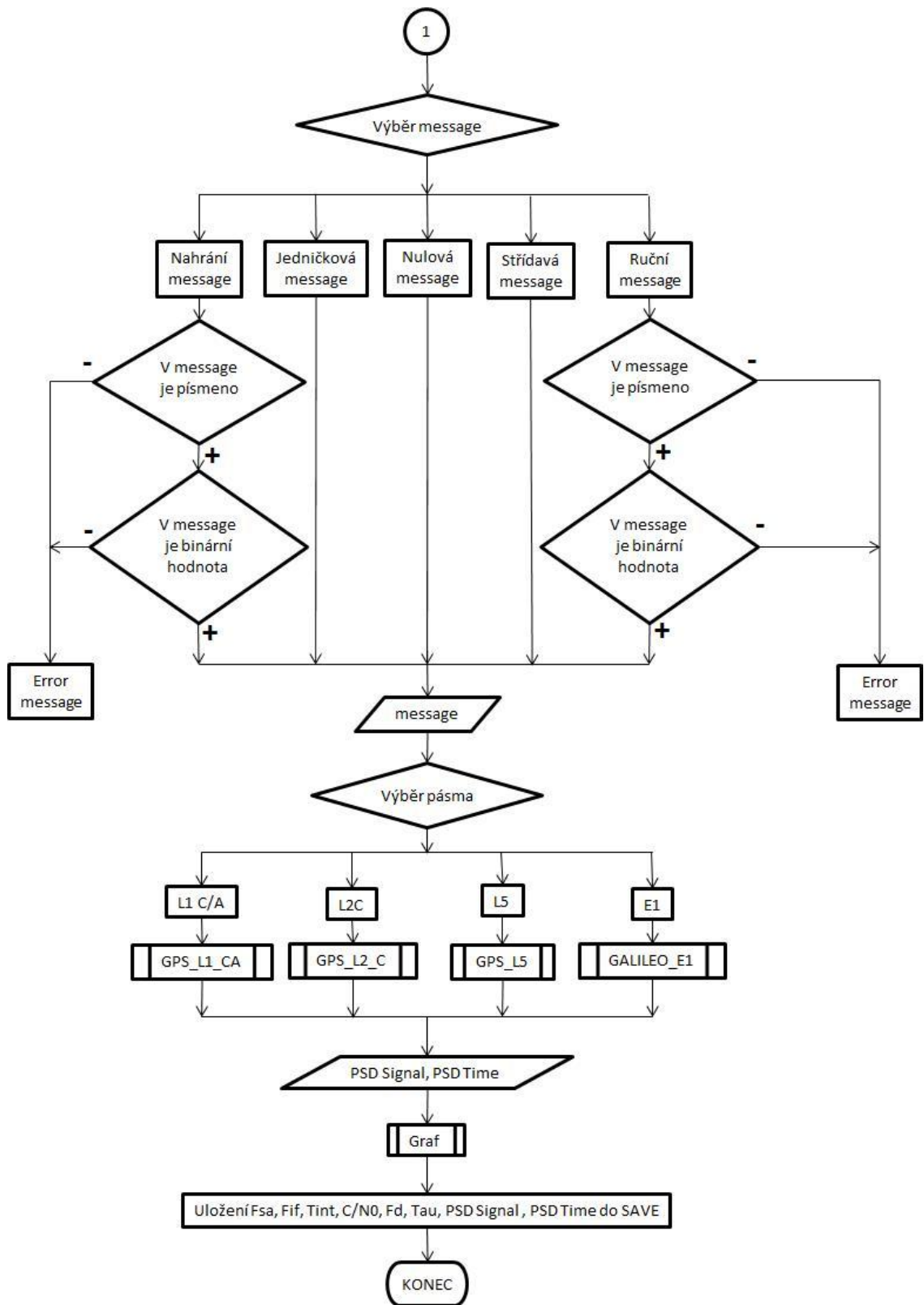
PRAKTICKÁ ČÁST



3. GNSS generátor

Struktura tohoto toolboxu je rozdělena na několik částí (m-filů). Je zde metoda `GNSS_generátor`, která má za úkol vykreslení hlavního okna. Dále tato metoda podle vstupních parametrů rozhoduje jaké další funkce na zpracování signálu bude volat. Výpočet každého pásma v navigačním systému je složen ze dvou funkcí. První volaná funkce je pro samotný výpočet signálu. Je zde obsažena modulace, posun čipů, Dopplerův posun, atd. Na začátku těchto programů se volá druhá funkce, která má za úkol generaci různých kódů pro další práci při tvorbě signálů. Ke konci první funkce je volána další metoda pro přidání šumu a posléze metoda pro výpočet spektrální výkonové hustoty. Pro lepší orientaci a představu o programu je zde uveden vývojový diagram `GNSS_generator.m`. Zbylé vývojové diagramy se zdrojovými kódy jsou uvedeny v přílohách.







3.1. Popis hlavního okna před výpočtem

Po spuštění programu je zobrazeno okno, které je rozděleno do tří hlavních částí:

- 1) Parametry signálu
- 2) Navigační zpráva
- 3) Místo příjmu

The screenshot shows the 'GNSS_generator' application window. It is divided into three main sections:

- Parametry signálu (Signal Parameters):** Includes dropdowns for 'Systém: GPS', 'Pásmo: L1 C/A', and 'PRN: 1'. Radio buttons for 'Fsa:' (selected), 'Tsa:', 'BP', and 'CE'. 'Fsa:' has a value of '8.3 MHz'. 'BP' has a value of '2 MHz'. 'Tint:' is '0.02 s'. A 'Save:' field contains 'BP_GPS_L1CA_druzice1'. A red '1)' is next to the 'BP' radio button.
- Navigační zpráva (Navigation Message):** Radio buttons for 'Ze souboru:', 'Jedničková:' (selected), 'Nulová:', 'Střídavá:', and 'Ruční:'. The 'Jedničková:' field contains '111111111111111111...'. A red '2)' is next to the 'Nulová:' radio button.
- Místo příjmu (Reception Location):** Input fields for 'C/No: 1000 dB-Hz', 'Fd: 0 MHz', and 'Tau: 0 chip'. A red '3)' is next to the 'Fd:' field.

At the bottom, there are two buttons: 'Výpočet' (Calculate) and 'Konec' (End).



3.1.1. Parametry signálu

V této části jsou zadávány základní požadavky na tvořený signál jako je například navigační systém, pásmo určitého systému, PRN družice, délka signálu, frekvenční požadavky a mimo jiné i název souboru pro ukládání.

3.1.1.1. Navigační systém

Systém: GPS
GPS
GALILEO

Program je vytvořen prozatím pro celkový systém GPS a GALILEO. Při vybrání určitého systému se mění ostatní parametry signálu podle aktuálního pásma jako např. Vzorkovací frekvence nebo také počty družic.

3.1.1.2. Pásmo

Pásmo: L1 C/A
L1 C/A
L2 C
L5

Pásmo: E1
E1

Výběr pásma je závislý na navigačním systémem. Tento výběr ovlivňuje parametry signálu F_{sa} a F_{if} v případě BP¹⁷. Dále je podle výběru pásma při výpočtu volen určitý m-file pro vygenerování signálu.

¹⁷ Band Pass – pásmový signál



3.1.1.3. PRN

Pomocí hodnoty PRN je při výpočtu vybírána inicializace kódů, které jsou následně použity v posuvných registrech.

3.1.1.4. Tint

Tint je délka požadovaného signálu, který dále určuje počet bitů navigační zprávy s ohledem na její číповou rychlost v různých pásmech.

3.1.1.5. Fsa a Tsa



Fsa je vzorkovací frekvence ta musí být dvakrát větší než Fif s šířkou pásma u BP. Tato podmínka je při nesplnění zachycena chybovým hlášením. U Fsa je možnost změny předpony u hertzů, což mění hodnotu v řádech 10^0 , 10^3 , 10^6 a 10^9 hertzů.

3.1.1.6. BP a CE

The screenshot shows a control panel with two rows of radio buttons. The first row has a selected radio button for 'BP' and an unselected one for 'CE'. To the right of these is a text input field containing the number '2' and a dropdown menu currently set to 'MHz'. The second row has an unselected radio button for 'BP' and a selected radio button for 'CE'.

U BP je možnost změny Fif předpony hertzů stejně jako u Fsa.

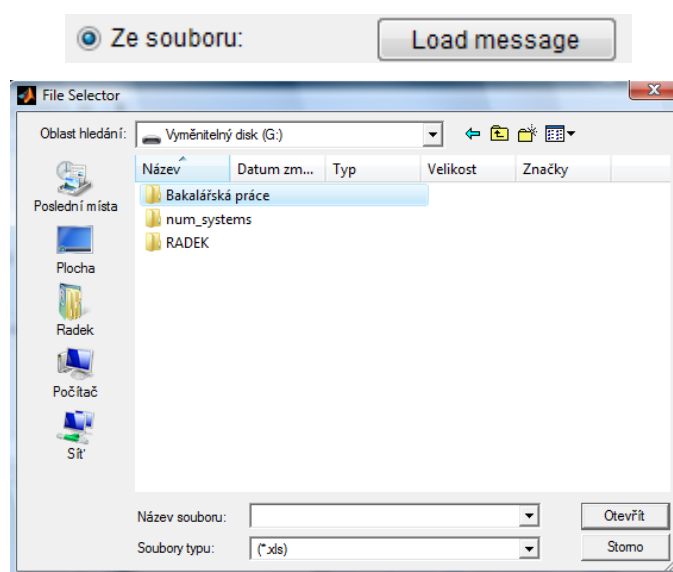
3.1.1.7. Save

The screenshot shows two 'Save:' labels. The first is followed by a text input field containing the file name 'BP_GPS_L1CA_druzice1'. The second is followed by a text input field containing the file name 'CE_GPS_L1CA_druzice1'.

Název souboru se mění automaticky podle zadání navigačního systému, pásma, PRN a BP, nebo CE. Do tohoto souboru jsou ukládány informace o signálu jako Tint, Fif, Fsa nebo Tsa, C/N0, Fd, Tau a po výpočtu také signál samotný. Tento název souboru pro uložení se dá změnit dle vlastních potřeb uživatele.

3.1.2. Navigační zpráva

3.1.2.1. Ze souboru





Nahrání navigační zprávy je provedeno pro formáty typu *.xls a *.xlsx. V těchto formátech souborů musí být v každé buňce binární hodnota. Pokud by byla zjištěna jiná než binární hodnota, je toto nahrávání zprávy ošetřeno chybovým oknem s popisem pozice, na které se nachází. Zohledněno je také chybovým hlášením zadání písmene a délka navigační zprávy.

3.1.2.2. Jedničková

Jedničková: 1111111111111111...

Jedničková navigační zpráva je tvořena pomocí vektoru jedniček, které jsou zohledněny na délku signálu.

3.1.2.3. Nulová

Nulová: 0000000000000000...

Nulová navigační zpráva je tvořena pomocí vektoru nul, které jsou zohledněny na délku signálu.

3.1.2.4. Střídavá

Střídavá: 1010101010101010...

Střídavá navigační zpráva je tvořena pomocí vektoru, kde se periodicky střídají hodnoty 1 a 0, které jsou zohledněny na délku signálu.

3.1.2.5. Ruční

Ruční:

Ruční zadávání navigační zprávy je jako v předchozích případech zohledněno délkou signálu, což ošetřuje chybové okno, které hlásí počet potřebných bitů do plné délky. Stejně jako u nahrání ze souboru jsou zde chybová okna na nebinární hodnotu a na zadání písmene.

3.1.3. Místo příjmu

3.1.3.1. C/N0

C/N0: dB-Hz



C/N0 je odstup signálu k šumu. Implicitně je zde nastavená hodnota 1000 dB-Hz, kde se nad tuto hodnotu šum neprojeví. To je zohledněno informačním oknem. Dále je zde chybové okno proti zadání textu. Funkce výpočtu bílého šumu by měla simulovat okolní rušivé signály na vysílaná pásma.

3.1.3.2. Fd

Fd: MHz ▾
Hz
kHz
MHz
GHz

Dopplerovo zpoždění je dáno pohybem družic a uživatele při příjmu signálu. To je zde reprezentováno Dopplerovou frekvencí F_d , která stejně jako F_{if} nebo F_{sa} může být v 10^0 , 10^3 , 10^6 a 10^9 hertzů. U zadávání hodnoty se objeví chybové okno, pokud je zadán nečíselný formát.

3.1.3.3. Tau

Tau: chip

K zpoždění čipů dochází šířením signálu konečnou rychlostí c na velkou vzdálenost. Toto zpoždění je zde hodnota τ , která se přičítá či odečítá při tvorbě signálů. Opět je zde ošetřeno zadávání textu.

3.1.4. Funkční tlačítka hlavního okna

3.1.4.1. Start

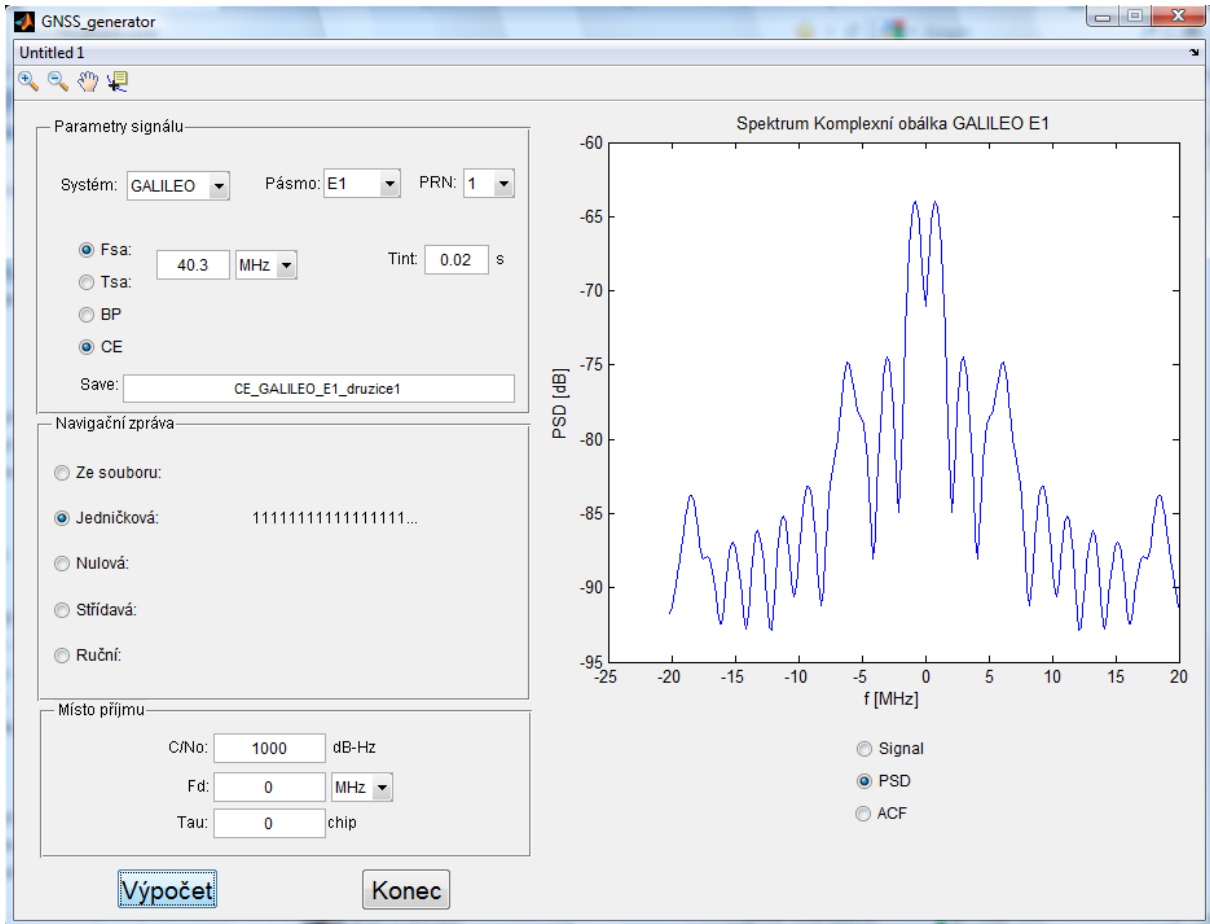
Po stisku tohoto tlačítka se provede vygenerování signálu a výpočet jeho charakteristik, které jsou voleny z parametrů signálu. Nakonec je celý signál vykreslen.

3.1.4.2. Konec

Toto tlačítko ukončí celý program.



3.2. Popis hlavního okna po výpočtu



3.2.1. Graf

Vykreslování v grafu je prvotně ve spektrální výkonové hustotě (PSD). X-ová osa je v takové frekvenci ve které jsou vstupní parametry. Y-ová osa je v decibelových souřadnicích. Dále je možnost překreslení na samotný signál či autokorelaci (ACF). Nad grafem je popisek, který označuje vykreslený signál.

3.2.2. Signal

Při kliknutí na toto tlačítko je z pomocného souboru načten původní signál a ten je posléze vykreslen.



3.2.3. PSD

Při kliknutí na PSD je volána funkce pro výpočet spektrální výkonové hustoty, kde se opět ze souboru nahraje původní signál a touto funkcí zpracuje. Po zpracování do frekvenční oblasti je signál vykreslen.

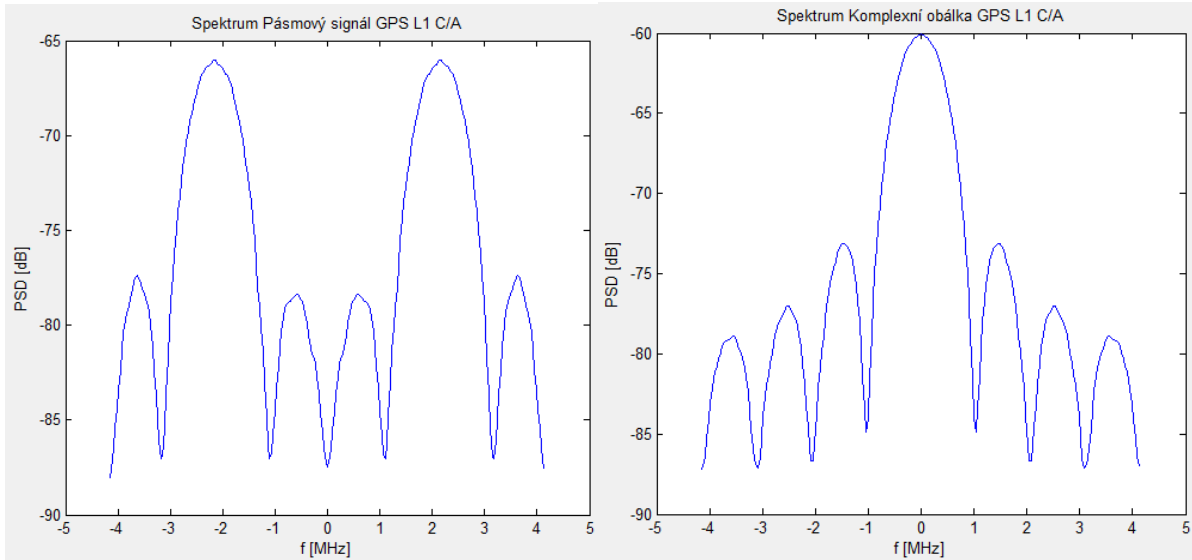
3.2.4. ACF

Stejným způsobem jako u PSD je zpracování signálu pomocí ACF. Zavolá se funkce na zpracování autokorelační funkce signálu, který je nahrán z pomocného souboru, a ten je poté vykreslen.

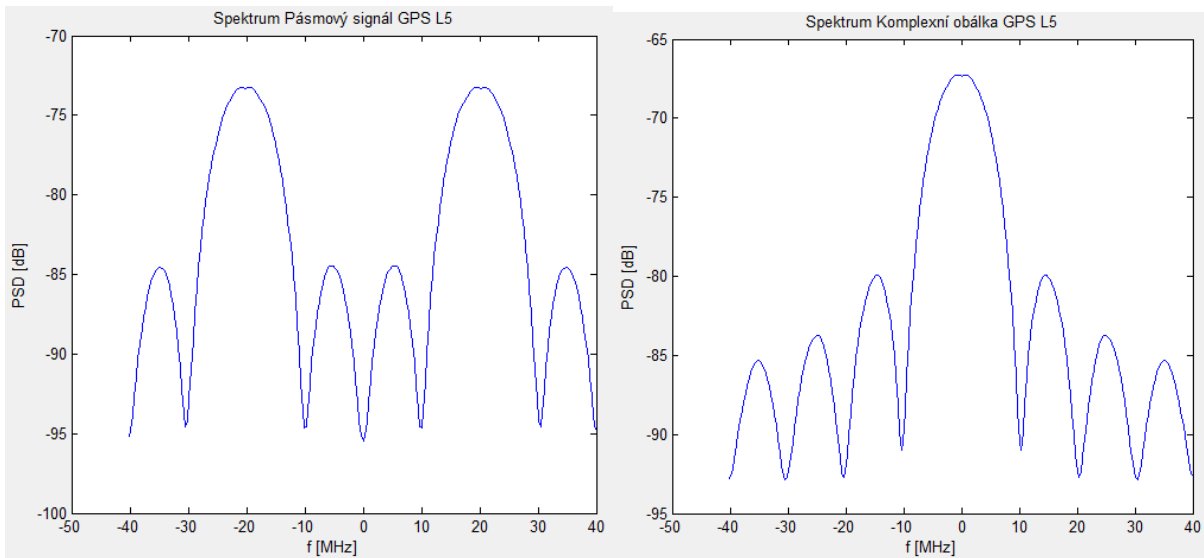


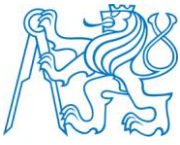
VÝSLEDKY PRÁCE

Vygenerované obrázky spekter pásmového signálu a komplexní obálky GPS L1C/A

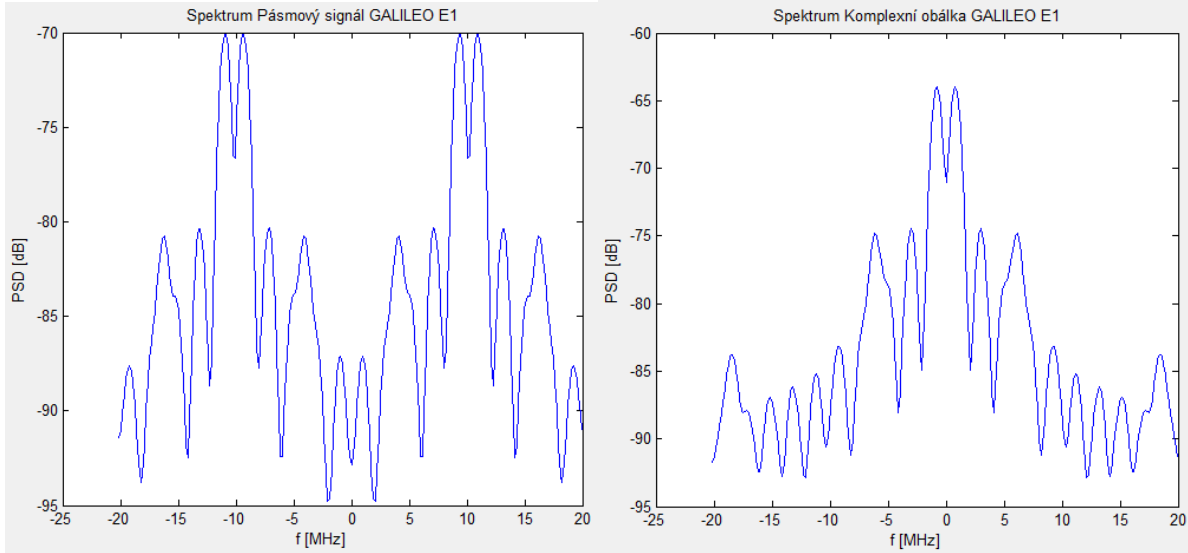


Vygenerované obrázky spekter pásmového signálu a komplexní obálky GPS L5

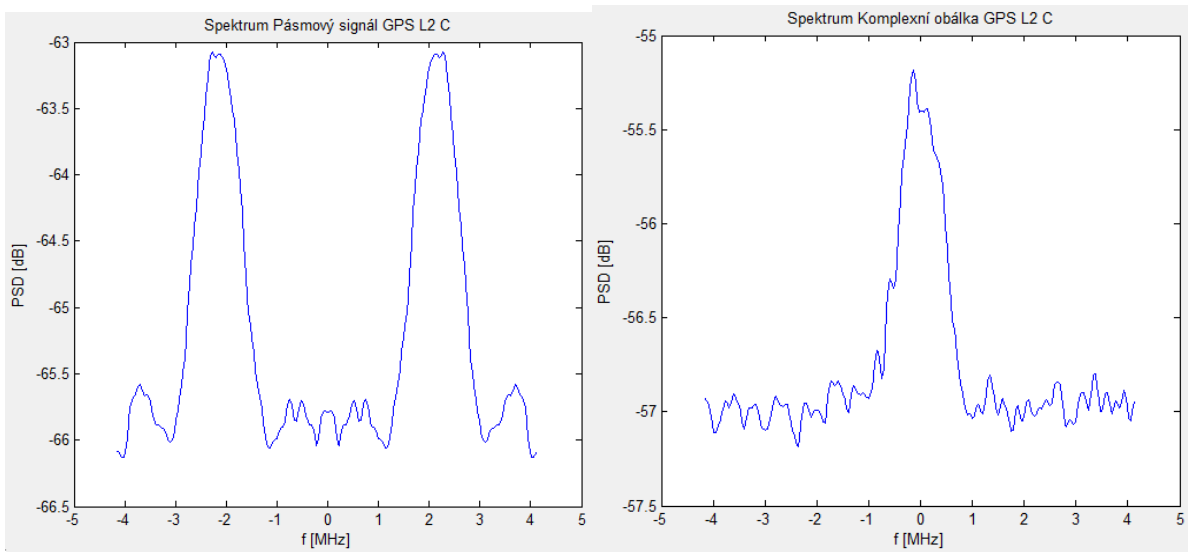




Vygenerované obrázky spekter pásmového signálu a komplexní obálky GALILEO E1



Vygenerované obrázky spekter pásmového signálu a komplexní obálky GPS L2C s šumem





ZÁVĚR

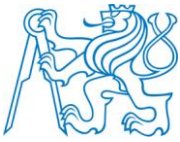
Zpracovaná problematika je daleko rozsáhlejší, než by umožňoval rozsah této bakalářské práce. Cílem bylo obeznámení s navigačními systémy a jejich strukturou. Na základě těchto informací posléze vypracovat toolbox v prostředí MATLAB.

V teoretické části práce jsou uvedeny struktury systémů a pásem. Na začátku této části jsou vysvětleny základní pojmy potřebné k vypracování toolboxu a Modulace používané v navigačních systémech.

Praktickou část jsem věnoval popisu vytvořeného toolboxu s podrobným rozebráním jednotlivých částí. Na konec praktického bloku jsou uvedeny některé vygenerované signály.

Vytvořený program generuje různé typy pásem navigačních systémů. Pomocí vstupních parametrů je vykreslen signál ve spektrální výkonové hustotě. Další možnosti vykreslení jsou v autokorelační funkci nebo signálu samotném. Mezi vstupní parametry patří zejména Dopplerův kmitočet, odstup signál ku šumu a posunutí čipů. S těmito třemi parametry je vytvořen signál s ohledem na místo příjmu.

Vykreslené signály se shodují s obrázky a informacemi v uvedených a použitých zdrojích. Následné rozvinutí programu by mohlo být pomocí dalších navigačních systémů, nebo také některých nepříznivých vlivů na signál jako je například vícecestné šíření. To by mohlo prohloubit mé znalosti nejen v signálech, ale také například v programovacím jazyku MATLAB.



Použité zdroje

- [1] Dissertacao [online]. Dspace.ist, 4. 2011 [cit. 08-05-2012].
<<https://dspace.ist.utl.pt/bitstream/2295/1066816/1/dissertacao.pdf>>
- [2] Enge, P. – Mistry, P. Global positioning system: Signals, Measurements and Performance. Massachusetts: Ganga-Jamuna Press, 2001. ISBN 0-9709544-0-9
- [3] Galileo OS SIS ICD [online]. Ec.europa, 11. 2010 [cit. 08-05-2012].
<http://ec.europa.eu/enterprise/policies/satnav/galileo/files/galileo-os-sis-icd-issue1-revision1_en.pdf>
- [4] Galileo Signal Plan [online]. Navipedia, 23. 2. 2012 [cit. 08-05-2012].
<http://navipedia.net/index.php/GALILEO_Signal_Plan>
- [5] Global Positioning System [online]. Wikipedie – otevřená encyklopedie, 16. 4. 2012 [cit. 08-05-2012]. <http://cs.wikipedia.org/wiki/Global_Positioning_System>
- [6] Globální družicový polohový systém [online]. Wikipedie – otevřená encyklopedie, 2. 4. 2012 [cit. 08-05-2012].
<http://cs.wikipedia.org/wiki/Glob%C3%A1ln%C3%AD_dru%C5%BEicov%C3%A9_polohov%C3%A9_syst%C3%A9my>
- [7] Glonass [online]. Wikipedie – otevřená encyklopedie, 6. 3. 2012 [cit. 08-05-2012].
<<http://cs.wikipedia.org/wiki/Glonass>>
- [8] GLONASS [online]. Glonass/ianc.rsa, 20. 3. 2012 [cit. 11-05-2012].
<<http://www.glonass-ianc.rsa.ru/en/index.php>>
- [9] GLONASS ICD [online]. Facility.unavco, 2008 [cit. 08-05-2012].
<[http://facility.unavco.org/data/docs/ICD_GLONASS_5.1_\(2008\)_en.pdf](http://facility.unavco.org/data/docs/ICD_GLONASS_5.1_(2008)_en.pdf)>
- [10] GPS-broadcast-signal [online]. Wikipedia – The Free Encyklopedia, 4. 12. 2011 [cit. 60-04-2012]. <<http://en.wikipedia.org/wiki/File:GPS-broadcast-signal.png>>
- [11] Hrdina, Z. – Pánek, P. – Vejražka, F. Radiové určování polohy (Družicový systém GPS). Praha: ČVUT, 1995. ISBN
- [12] IS-GPS-200F [online]. GPS.GOV, 21. 11 2011 [cit. 08-05-2012].
<<http://www.gps.gov/technical/icwg/IS-GPS-200F.pdf>>



- [13] IS-GPS-705B [online]. GPS.GOV, 21. 11. 2001 [cit. 08-05-2012].
<<http://www.gps.gov/technical/icwg/IS-GPS-705B.pdf>>
- [14] IS-GPS-800B [online]. GPS.GOV, 21. 11. 2011 [cit. 08-05-2012].
<<http://www.gps.gov/technical/icwg/IS-GPS-800B.pdf>>
- [15] Kaplan, E. D. – Hegarty, Ch. J. Understanding GPS: Principles and Applications. Norwood: Artech house, INC., 2006. ISBN 1-58053-894-0
- [16] Navigační systém Galileo [online]. Wikipedie – otevřená encyklopedie, 25. 11. 2011 [cit. 08-05-2012].
<http://cs.wikipedia.org/wiki/Naviga%C4%8Dn%C3%AD_syst%C3%A9m_Galileo>
- [17] Pšenčíková, J. Algoritmizace. Computer Media s.r.o., 2009. ISBN 978-80-7402-034-6
- [18] Samana, N. Global Positioning“ Technologies and Performance. John Wiley & Sons, INC., 2008. ISBN 978-0-471-79376-2
- [19] Siddiui [online]. Dspace.cc.tut, 4. 11. 2009 [cit. 08-05-2012].
<<http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/6688/siddiqui.pdf?sequence=3>>
- [20] Tranzit [online]. Wikipedie – otevřená encyklopedie, 16. 4. 2012 [cit. 08-05-2012].
<<http://cs.wikipedia.org/wiki/Transit>>
- [21] Učebnice teorie radiové komunikace [online]. Urel.feec.vutbr, 4. 1. 2011 [cit. 22-4-2012].
<<http://www.urel.feec.vutbr.cz/MTRK/?%DAvod>>
- [22] Zaplatílek, K. – Doňar, B. MATLAB: Tvorba uživatelských aplikací. Praha: BEN – technická literatura, 2004. ISBN 80-7300-133-0
- [23] Zaplatílek, K. – Doňar, B. MATLAB: Začínáme se signály. Praha: BEN – technická literatura, 2006. ISBN 80-7300-200-0



SEZNAM OBRÁZKŮ

<i>Obr. 1</i> - Šíření radiových vln (převzato z [2])	13
<i>Obr. 2</i> – GPS (převzato z [18]).....	15
<i>Obr. 3</i> - GLONASS (převzato z [18]).....	16
<i>Obr. 4</i> - GALILEO pásma a jejich služby (převzato z [18])	17
<i>Obr. 5</i> - Modulace BPSK	21
<i>Obr. 6</i> - QPSK modulace.....	22
<i>Obr. 7</i> - Syntéza GPS L1 C/A	23
<i>Obr. 8</i> - Generátor C/A a PRN kódu (převzato z [2])	25
<i>Obr. 9</i> - Tvorba posloupnosti L1Cd a L1Cp (převzato z [14]).....	28
<i>Obr. 10</i> - Registr pro tvorbu CM a CL kódu (převzato z [12]).....	30
<i>Obr. 11</i> - Registry pro kódy XA, XBI a XBQ (převzato z [13]).....	32
<i>Obr. 12</i> - 1/2 konvoluční kodér (převzato z [3]).....	38



SEZNAM TABULEK

<i>Tabulka 1</i> - Frekvenčních pásem (převzato z [2]).....	14
<i>Tabulka 2</i> - Jedna perioda BOC modulace (převzato z [3])	22
<i>Tabulka 3</i> - ID družic a výběr pozic selektoru (převzato z [2])	24
<i>Tabulka 4</i> - Hodnoty pro PRN L1C _P a L1C _D (převzato z [13]).....	27
<i>Tabulka 5</i> - Hodnoty pro PRN L1C ₀ (převzato z [13]).....	28
<i>Tabulka 6</i> - Hodnoty pro PRN L2C (převzato z [11])	29
<i>Tabulka 7</i> - Hodnoty pro PRN L5 (převzato z [12])	31
<i>Tabulka 8</i> - Hodnoty pro registr 2 E5a-I (převzato z [3])	34
<i>Tabulka 9</i> - Hodnoty pro registr 2 E5a-Q (převzato z [3]).....	34
<i>Tabulka 10</i> - Indexy primárních kódů E5 (převzato z [3])	35
<i>Tabulka 11</i> - Hodnoty pro registr 2 E5b-I (převzato z [3]).....	35
<i>Tabulka 12</i> - Hodnoty pro registr 2 E5b-Q (převzato z [3])	36
<i>Tabulka 13</i> - Inicializace registrů 2 E1C a E1B (převzato z [1]).....	37

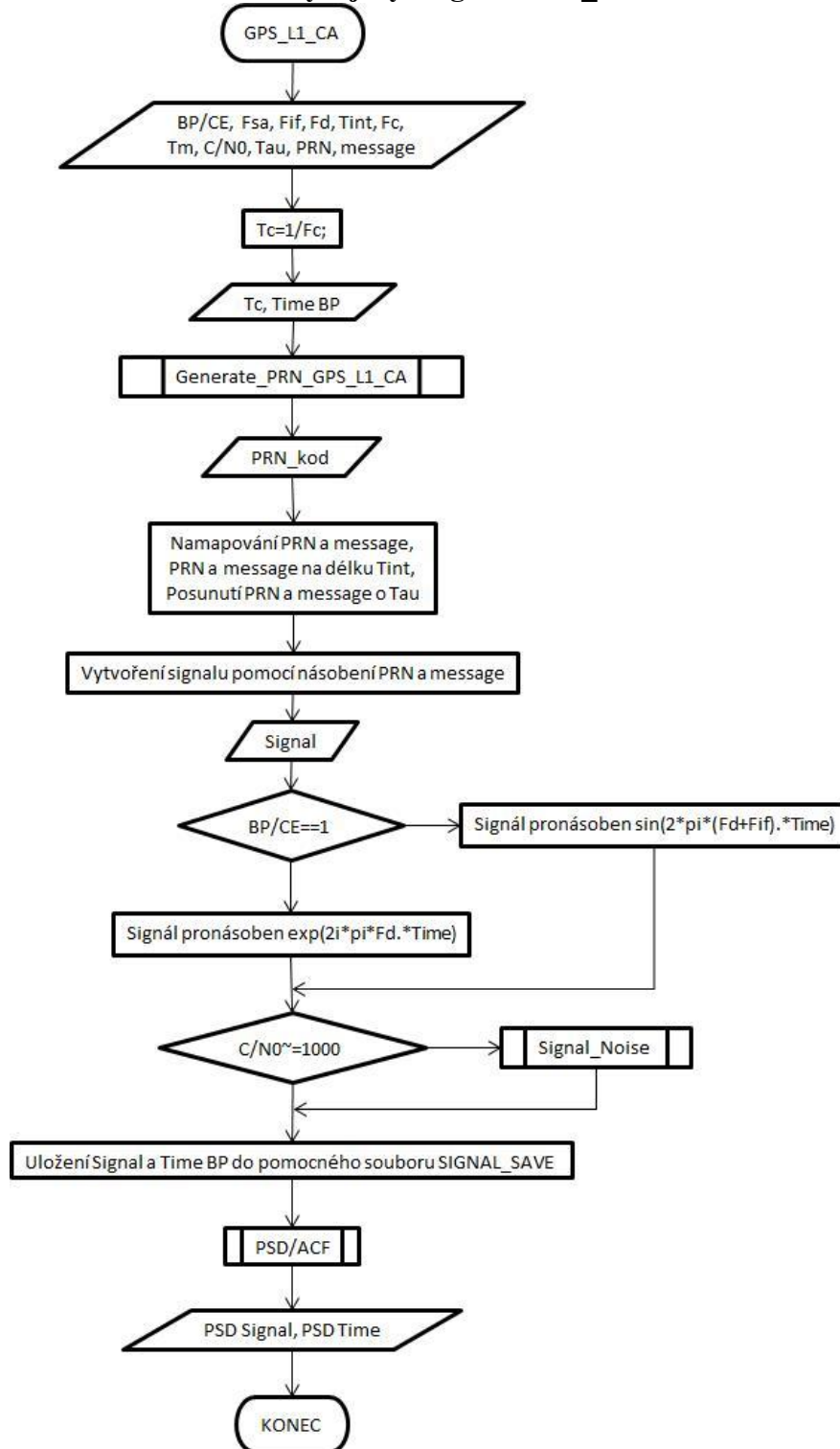


SEZNAM PŘÍLOH

Příloha A - Vývojový diagram GPS_L1_CA.m	I
Příloha B - Vývojový diagram GPS_L2_C.m.....	II
Příloha C - Vývojový diagram GPS_L5.m.....	III
Příloha D - Vývojový diagram GALILEO_E1.m	IV
Příloha E - Vývojový diagram Generate_GPS_L1_CA.m	V
Příloha F - Vývojový diagram Generate_GPS_L2_C.m	VI
Příloha G - Vývojový diagram Generate_GPS_L5.m.....	VII
Příloha H - Vývojový diagram Generate_GALILEO_E1.m	VIII
Příloha I - Vývojový diagram Niose_Signal.m	IX
Příloha J - Vývojový diagram PSD_ACF.m	X
Příloha K - Zdrojový kód GNSS_generator.m.....	XI
Příloha L - Zdrojový kód GPS_L1_CA.m	XXXI
Příloha M - Zdrojový kód GPS_L2_C.m	XXXII
Příloha N - Zdrojový kód GPS_L5.m	XXXIII
Příloha O - Zdrojový kód Generate_PRN_GPS_L2_C.m.....	XXXVI
Příloha P - Zdrojový kód Generate_PRN_GPS_L5.m	XXXVIII
Příloha Q - Zdrojový kód Generate_PRN_GALILEO_E1.m	XL
Příloha R - Zdrojový kód Noise_Signal.m.....	XLI
Příloha S - Zdrojový kód PSD_ACF.m	XLII

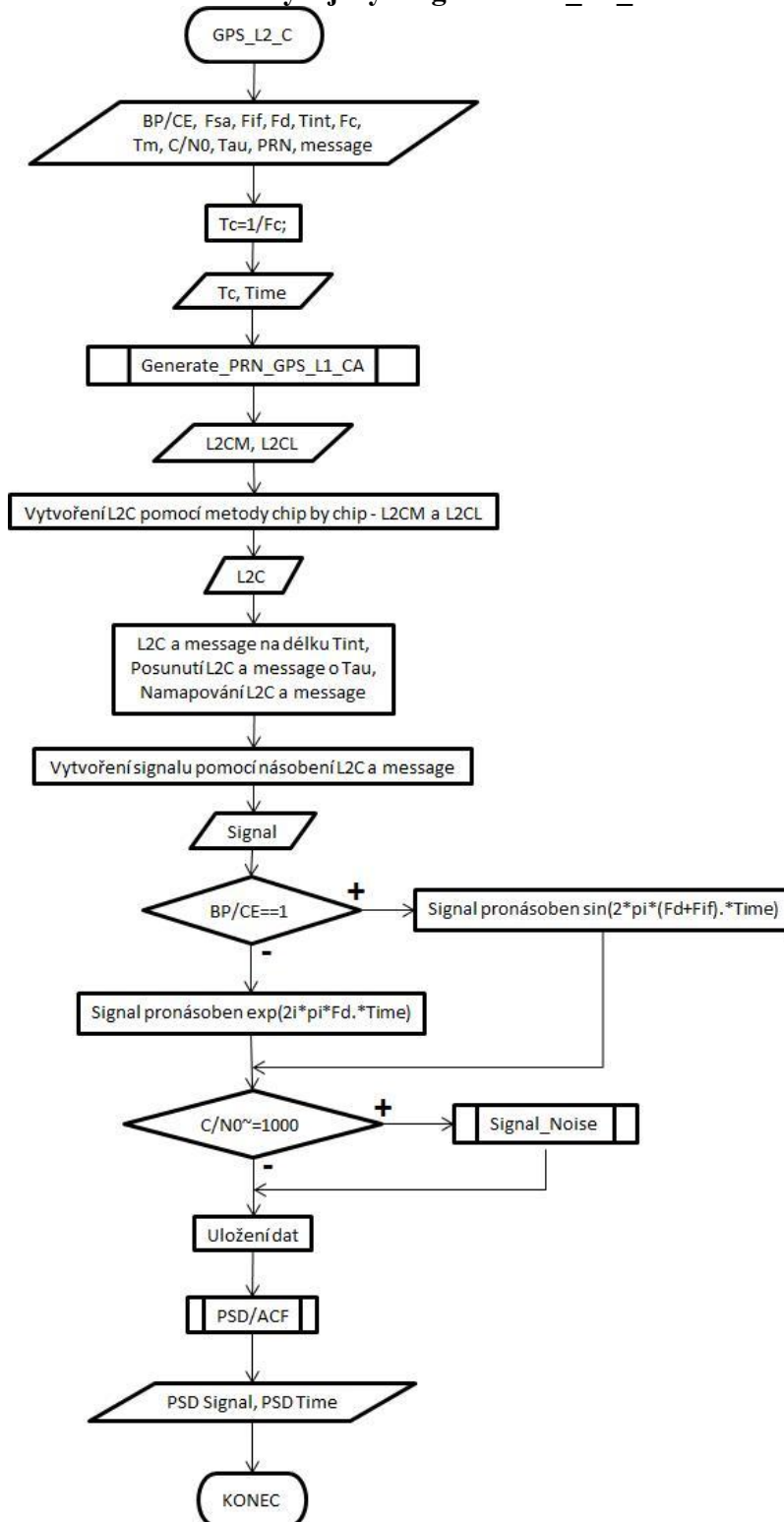


Příloha A - Vývojový diagram GPS_L1_CA.m



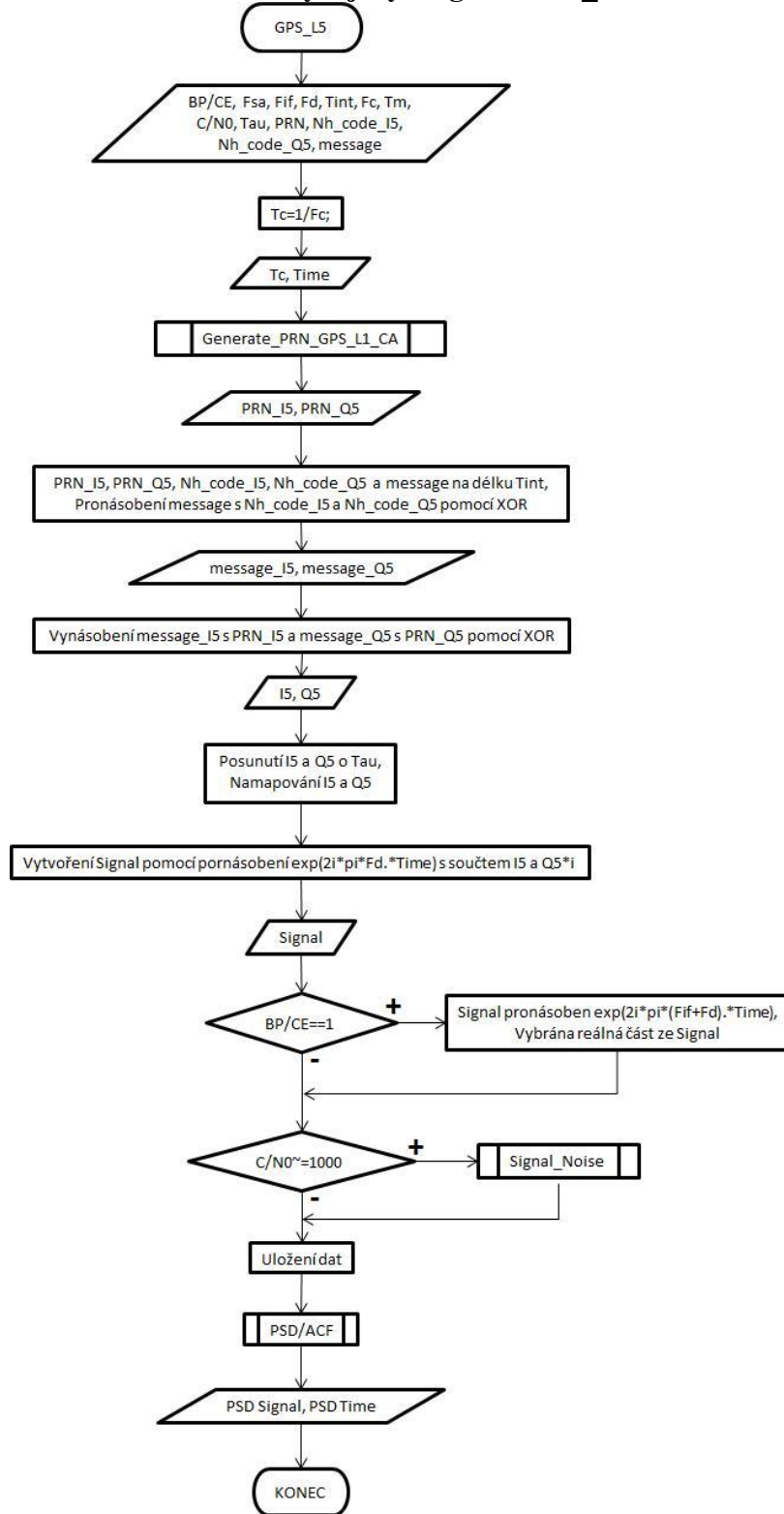


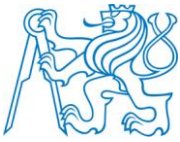
Příloha B - Vývojový diagram GPS_L2_C.m



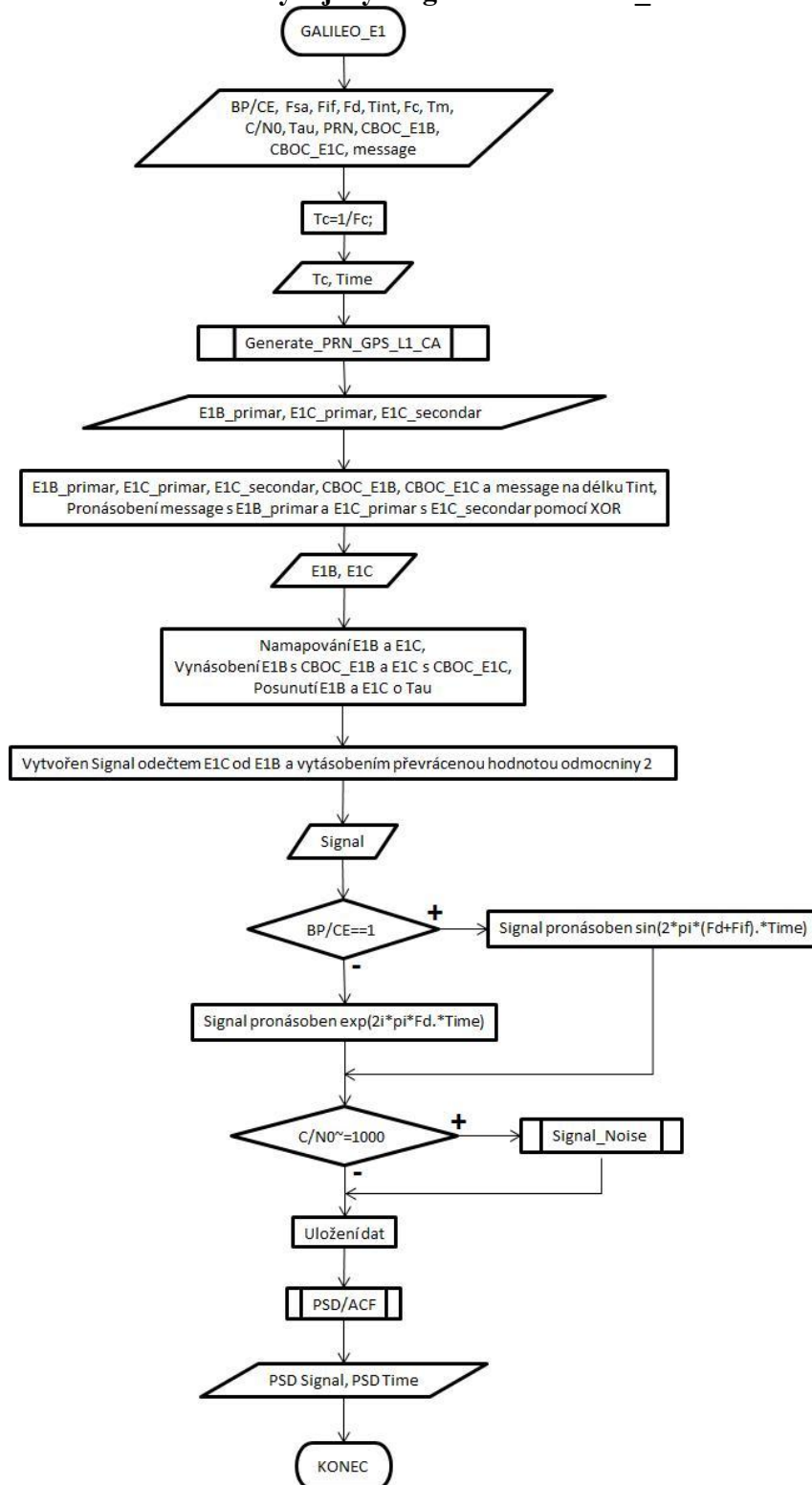


Příloha C - Vývojový diagram GPS_L5.m



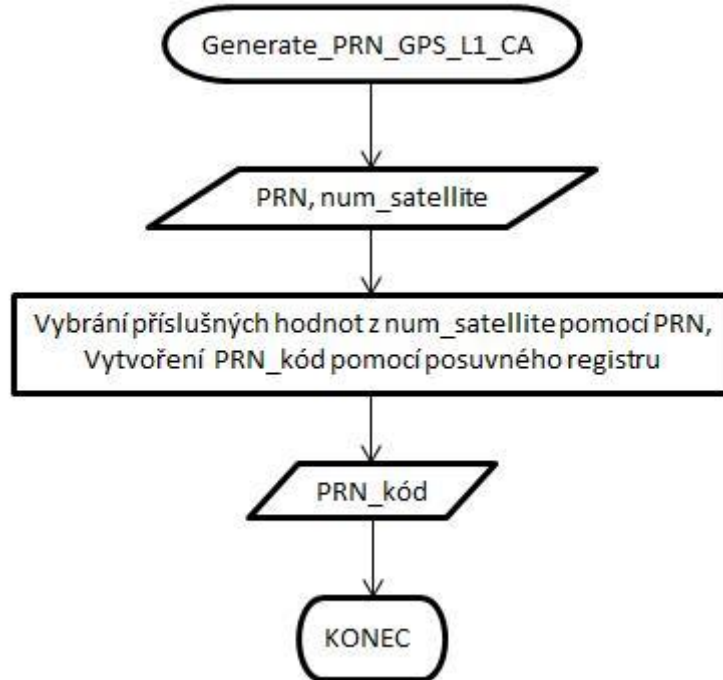


Příloha D - Vývojový diagram GALILEO_E1.m



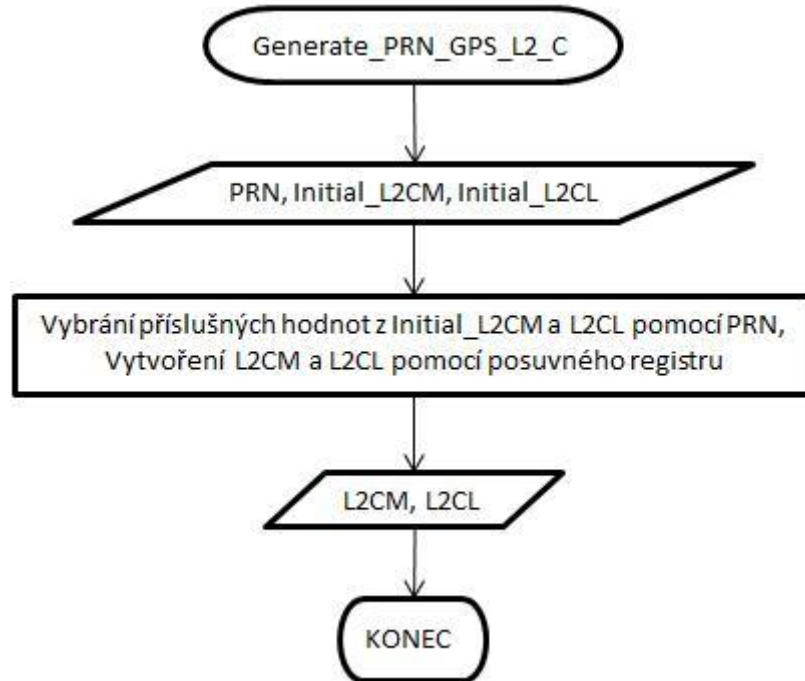


Příloha E - Vývojový diagram Generate_GPS_L1_CA.m



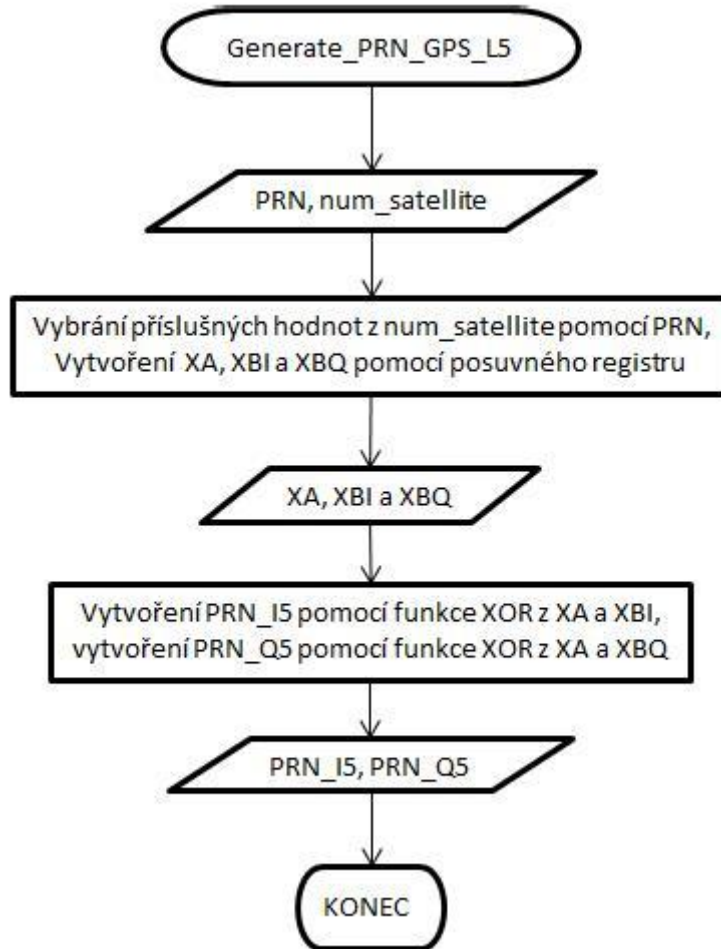


Příloha F - Vývojový diagram Generate_GPS_L2_C.m



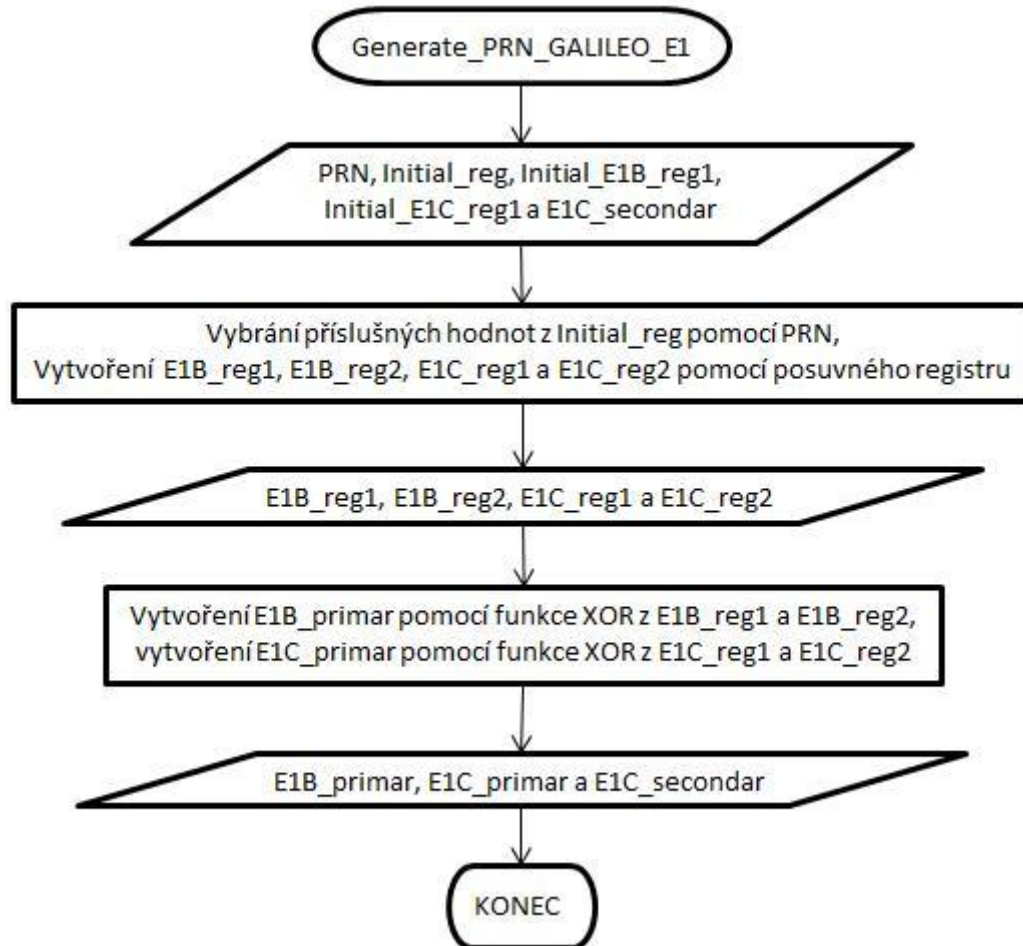


Příloha G - Vývojový diagram Generate_GPS_L5.m



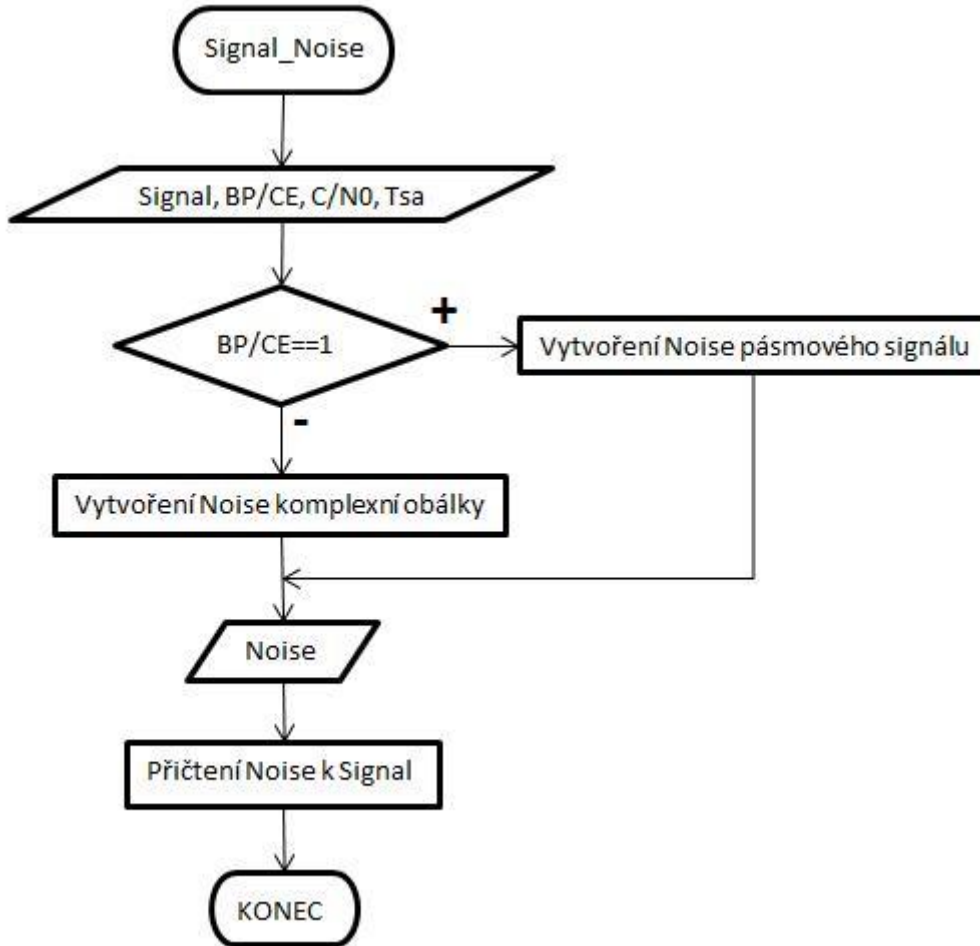


Příloha H - Vývojový diagram Generate_GALILEO_E1.m



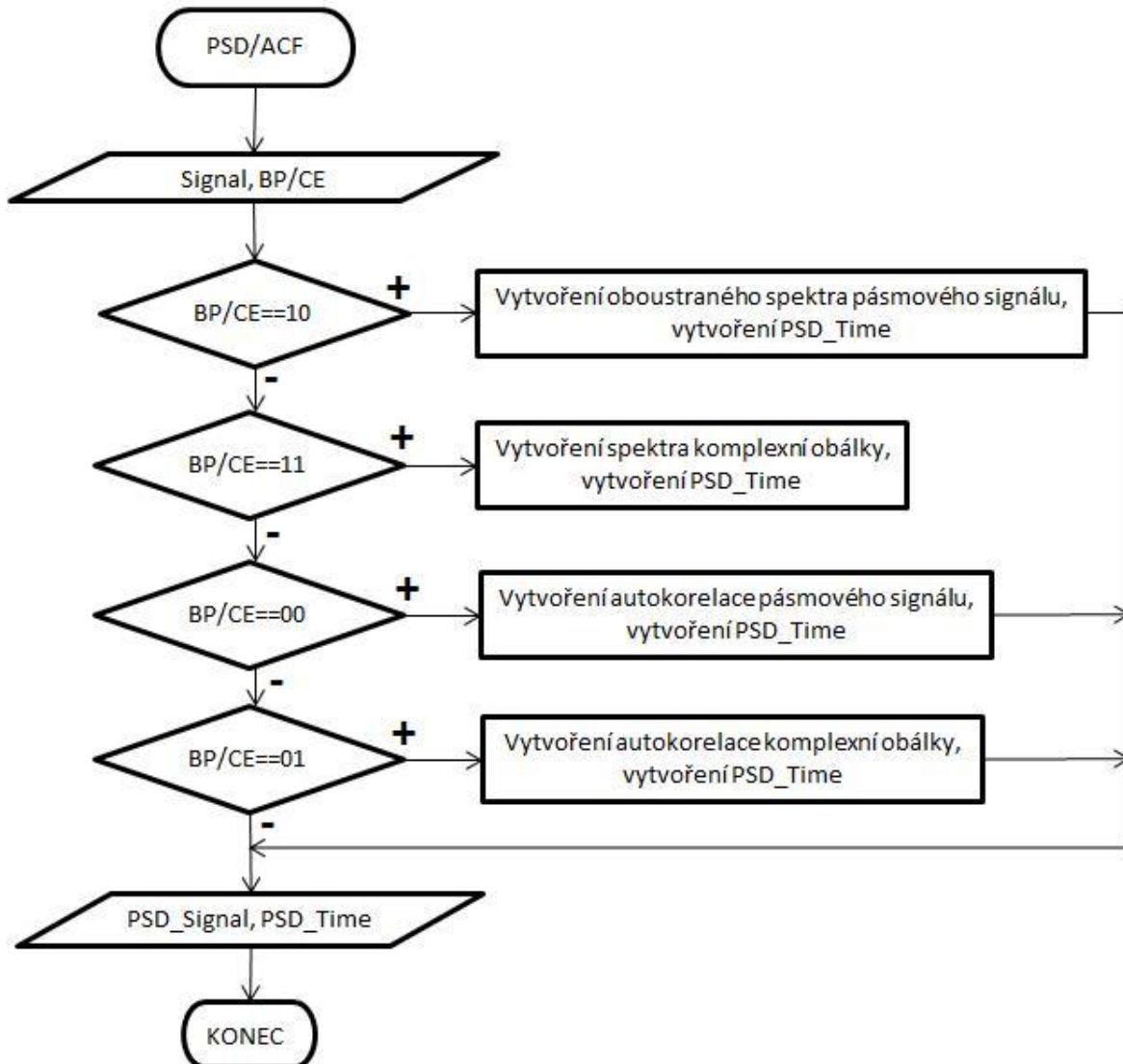


Příloha I - Vývojový diagram Niose_Signal.m





Příloha J - Vývojový diagram PSD_ACF.m





Příloha K - Zdrojový kód GNSS_generator.m

```
function varargout = GNSS_generator(varargin)
format long;
%GNSS_GENERATOR M-file for GNSS_generator.fig
% GNSS_GENERATOR, by itself, creates a new GNSS_GENERATOR or raises the
existing
% singleton*.
%
% H = GNSS_GENERATOR returns the handle to a new GNSS_GENERATOR or the handle to
the existing singleton*.
%
% GNSS_GENERATOR('Property','Value',...) creates a new GNSS_GENERATOR using the
given property value pairs. Unrecognized properties are passed via
varargin to GNSS_generator_OpeningFcn. This calling syntax produces a
warning when there is an existing singleton*.
%
% GNSS_GENERATOR('CALLBACK') and GNSS_GENERATOR('CALLBACK',hObject,...) call the
local function named CALLBACK in GNSS_GENERATOR.M with the given input
arguments.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

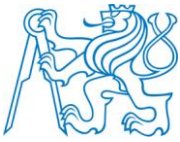
% Edit the above text to modify the response to help GNSS_generator

% Last Modified by GUIDE v2.5 26-Apr-2012 01:33:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GNSS_generator_OpeningFcn, ...
                  'gui_OutputFcn',  @GNSS_generator_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GNSS_generator is made visible.
function GNSS_generator_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
            command line (see VARARGIN)
% Choose default command line output for GNSS_generator
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes GNSS_generator wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = GNSS_generator_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
```



```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

function value_off(off)
set(off,'Value',0)
function visible_off(offed)
set(offed,'Visible','off')
function visible_on(oned)
set(oned,'Visible','on')
function value_on(on)
set(on,'Value',1)

% --- Executes on button press in pb_konec.
function pb_konec_Callback(hObject, eventdata, handles)
% hObject    handle to pb_konec (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
close all;
delete SIGNAL_SAVE

% --- Executes on button press in pb_vypocet.
function pb_vypocet_Callback(hObject, eventdata, handles)
% hObject    handle to pb_vypocet (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% konstanty
Nint=str2double(get(handles.edit_Tint,'String'));
if isnan(Nint)
    warndlg('Zadejte prosim ciselny format do Tint','Format Error');
    return;
end
PRN_satellite=(get(handles.pop_PRN_GPS,'Value'));
C_N0=str2double(get(handles.edit_C_N0,'String'));
Fd=str2double(get(handles.edit_Fd,'String'));
Tau=str2double(get(handles.edit_Tau,'String'));

% nacteni Fsa x Tsa
if(get(handles.rb_Fsa,'Value')==1)
    Fsa=str2double(get(handles.edit_Tsa_Fsa,'String'))*...
        get(handles.pop_Fsa,'UserData');
    if isnan(Fsa)
        warndlg('Zadejte prosim ciselny format do Tsa/Fsa','Format Error');
        return;
    else
        Tsa=1/Fsa;
    end
elseif(get(handles.rb_Tsa,'Value')==1)
    Tsa=str2double(get(handles.edit_Tsa_Fsa,'String'));
end
Fif=str2double(get(handles.edit_Fif,'String'))*get(handles.pop_Fif,'UserData');
if isnan(Fif)
    warndlg('Zadejte prosim ciselny format do Fif','Format Error');
    return;
end
% Tvorba signalu
switch get(handles.pop_nav_system,'Value')
% GPS
case 1
    switch get(handles.pop_GPS_pasmo,'Value')
% GPS L1 C/A
case 1
% nahrani message
```



```
if(get(handles.rb_load,'Value')==1)
    message_load=get(handles.pb_load, 'UserData');
    if message_load==0
        warndlg('Prosim nahrajte navigacni zpravu','Message Error');
        return;
    end
    if length(message_load)<Nint/20e-3
        warndlg(['V navigacni zprave chybi ',num2str((Nint/20e-3)-
length(message_load)),' bitu'],'Length Error');
        return;
    end
    if isnan(str2double(message_load))
        warndlg('Spatny format message','Format Error');
        return;
    else
        message=[];
        for m=1:1:Nint/20e-3
            if message_load(m)=='1' || message_load(m)=='0'
                message=[message str2double(message_load(m))];
            else
                warndlg(['Cislo na pozici ',num2str(m),' neni binar-
ni'],'Format Error');
                return;
            end
        end
    end
elseif(get(handles.rb_1,'Value')==1)
    message=ones(1,Nint/20e-3);
elseif(get(handles.rb_0,'Value')==1)
    message=zeros(1,Nint/20e-3);
elseif(get(handles.rb_10,'Value')==1)
    message=zeros(1,Nint/20e-3);
    for n=1:1:(length(message)+1)/2
        message(1+2*(n-1))=1;
    end
elseif(get(handles.rb_hand,'Value')==1)
    message_hand=(get(handles.edit_hand,'String'));
    message_hand=reshape(message_hand,1,length(message_hand));

    if length(message_hand)<Nint/20e-3
        warndlg(['V navigacni zprave chybi ',num2str((Nint/20e-3)-
length(message_hand)),' bitu'],'Length Error');
        return;
    end
    if isnan(str2double(message_hand))
        warndlg('Zadejte prosim ciselny format do Message rucni','Format Er-
ror');
        return;
    else
        message=[];
        for m=1:1:Nint/20e-3
            if message_hand(m)=='1' || message_hand(m)=='0'
                message=[message str2double(message_hand(m))];
            else
                warndlg(['Cislo na pozici ',num2str(m),' neni binar-
ni'],'Format Error');
                return;
            end
        end
    end
end
end
% Generace signalu
if(get(handles.rb_BP,'Value')==1)
    if(Fif*2<=(1/Tsa) && Fif~=0 && Fif*10>=(1/Tsa))
        [Signal Time]=GPS_L1_CA(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, 1, 1.023e6);
```




```
else
    warndlg('Fsa musi byt v intervalu (2*Fif,10*Fif)','Fsa Error');
    return;
end
elseif(get(handles.rb_CE,'Value')==1)
    if(2*get(handles.pop_Fsa,'UserData')<=(1/Tsa) &&
20*get(handles.pop_Fsa,'UserData')>=(1/Tsa))
        [Signal Time]=GPS_L1_CA(Tsa, 0, PRN_satellite, message, C_N0, Fd, Tau,
0, 1.023e6);
    else
        text=['Fsa musi byt v intervalu ('
num2str(2*get(handles.pop_Fsa,'UserData')) ',' num2str(20*get(handles.pop_Fsa,'UserData'))
')'];
        warndlg(text,'Fsa Error');
        return;
    end
end
% GPS L2 C
case 2
    % nahrani message
    if(get(handles.rb_load,'Value')==1)
        message_load=get(handles.pb_load,'UserData');
        if message_load==0
            warndlg('Prosim nahrajte navigacni zpravu','Message Error');
            return;
        end
        if length(message_load)<Nint/20e-3
            warndlg(['V navigacni zprave chybi ',num2str((Nint/20e-3)-
length(message_load)),' bitu'],'Length Error');
            return;
        end
        if isnan(message_load)
            warndlg('Spatny format message','Format Error');
            return;
        end
        else
            message=[];
            for m=1:1:Nint/20e-3
                if message_load(m)=='1' || message_load(m)=='0'
                    message=[message str2double(message_load(m))];
                else
                    warndlg(['Cislo na pozici ',num2str(m),' neni binar-
ni'],'Format Error');
                end
            end
            return;
        end
    end
    elseif(get(handles.rb_1,'Value')==1)
        message=ones(1,Nint/20e-3);
    elseif(get(handles.rb_0,'Value')==1)
        message=zeros(1,Nint/20e-3);
    elseif(get(handles.rb_10,'Value')==1)
        message=zeros(1,Nint/20e-3);
        for n=1:1:(length(message)+1)/2
            message(1+2*(n-1))=1;
        end
    elseif(get(handles.rb_hand,'Value')==1)
        message_hand=(get(handles.edit_hand,'String'));
        message_hand=reshape(message_hand,1,length(message_hand));

        if length(message_hand)<Nint/20e-3
            warndlg(['V navigacni zprave chybi ',num2str((Nint/20e-3)-
length(message_hand)),' bitu'],'Length Error');
            return;
        end
        if isnan(str2double(message_hand))
```



```
ror');
        warndlg('Zadejte prosim ciselny format do Message rucni','Format Er-
ror');
        return;
    else
        message=[];
        for m=1:1:Nint/20e-3
            if message_hand(m)=='1' || message_hand(m)=='0'
                message=[message str2double(message_hand(m))];
            else
                warndlg(['Cislo na pozici ',num2str(m),' neni binar-
ni'],'Format Error');
                return;
            end
        end
    end
    % Generace signalu
    if(get(handles.rb_BP,'Value')==1)
        if(Fif*2<=(1/Tsa) && Fif~=0 && Fif*10>=(1/Tsa))
            [Signal Time]=GPS_L2_C(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, 1, 511.5e3);
        else
            warndlg('Fsa musi byt v intervalu (2*Fif,10*Fif)','Fsa Error');
            return;
        end
        elseif(get(handles.rb_CE,'Value')==1)
            if(2*get(handles.pop_Fsa,'UserData')<=(1/Tsa) &&
20*get(handles.pop_Fsa,'UserData')>=(1/Tsa))
                [Signal Time]=GPS_L2_C(Tsa, 0, PRN_satellite, message, C_N0, Fd, Tau,
0, 511.5e3);
            else
                text=['Fsa musi byt v intervalu ( '
num2str(2*get(handles.pop_Fsa,'UserData')) ', ' num2str(20*get(handles.pop_Fsa,'UserData'))
')'];
                warndlg(text,'Fsa Error');
                return;
            end
        end
    % GPS L5
    case 3
        % nahrani message
        if(get(handles.rb_load,'Value')==1)
            message_load=get(handles.pb_load, 'UserData');
            if message_load==0
                warndlg('Prosim nahrajte navigacni zpravu','Message Error');
                return;
            end
            if length(message_load)<Nint/10e-3
                warndlg(['V navigacni zprave chybi ',num2str((Nint/10e-3)-
length(message_load)),' bitu'],'Length Error');
                return;
            end
            if isnan(message_load)
                warndlg('Spatny format message','Format Error');
                return;
            else
                message=[];
                for m=1:1:Nint/10e-3
                    if message_load(m)=='1' || message_load(m)=='0'
                        message=[message str2double(message_load(m))];
                    else
                        warndlg(['Cislo na pozici ',num2str(m),' neni binar-
ni'],'Format Error');
                        return;
                    end
                end
            end
        end
    end
end
```



```
end
elseif(get(handles.rb_1, 'Value')==1)
    message=ones(1,Nint/10e-3);
elseif(get(handles.rb_0, 'Value')==1)
    message=zeros(1,Nint/10e-3);
elseif(get(handles.rb_10, 'Value')==1)
    message=zeros(1,Nint/10e-3);
    for n=1:1:(length(message)+1)/2
        message(1+2*(n-1))=1;
    end
elseif(get(handles.rb_hand, 'Value')==1)
    message_hand=(get(handles.edit_hand, 'String'));
    message_hand=reshape(message_hand,1,length(message_hand));

    if length(message_hand)<Nint/10e-3
        warndlg(['V navigacni zprave chybi ', num2str((Nint/10e-3)-
length(message_hand))], ' bitu'], 'Length Error');
        return;
    end
    if isnan(str2double(message_hand))
        warndlg('Zadejte prosim ciselny format do Message rucni', 'Format Er-
ror');

        return;
    else
        message=[];
        for m=1:1:Nint/10e-3
            if message_hand(m)=='1' || message_hand(m)=='0'
                message=[message str2double(message_hand(m))];
            else
                warndlg(['Cislo na pozici ', num2str(m), ' neni binar-
ni'], 'Format Error');
                return;
            end
        end
    end
end
end
% Generace signalu
if(get(handles.rb_BP, 'Value')==1)
    if(Fif*2<=(1/Tsa) && Fif~=0 && Fif*10>=(1/Tsa))
        [Signal Time]=GPS_L5(Tsa, Fif, PRN_satellite, message, C_N0, Fd, Tau,
1, 10.230e6);
    else
        warndlg('Fsa musi byt v intervalu (2*Fif,10*Fif)', 'Fsa Error');
        return;
    end
elseif(get(handles.rb_CE, 'Value')==1)
    if(20*get(handles.pop_Fsa, 'UserData')<=(1/Tsa) &&
200*get(handles.pop_Fsa, 'UserData')>=(1/Tsa))
        [Signal Time]=GPS_L5(Tsa, 0, PRN_satellite, message, C_N0, Fd, Tau, 0,
10.230e6);
    else
        text=['Fsa musi byt v intervalu ( '
num2str(2*get(handles.pop_Fsa, 'UserData')) ', ' num2str(20*get(handles.pop_Fsa, 'UserData'))
')'];
        warndlg(text, 'Fsa Error');
        return;
    end
end
end
% GALILEO
case 2
    switch get(handles.pop_GALILEO_pasmo, 'Value')
        % GALILEO E1
        case 1
            % nahrani message
            if(get(handles.rb_load, 'Value')==1)
```



```
message_load=get(handles.pb_load, 'UserData');
if message_load==0
    warndlg('Prosim nahrajte navigacni zpravu','Message Error');
    return;
end
if length(message_load)<Nint/4e-3
    warndlg(['V navigacni zprave chybi ',num2str((Nint/4e-3)-
length(message_load)),' bitu'],'Length Error');
    return;
end
if isnan(message_load)
    warndlg('Spatny format message','Format Error');
    return;
else
    message=[];
    for m=1:1:Nint/4e-3
        if message_load(m)=='1' || message_load(m)=='0'
            message=[message str2double(message_load(m))];
        else
            warndlg(['Cislo na pozici ',num2str(m),' neni binar-
ni'],'Format Error');
            return;
        end
    end
    elseif(get(handles.rb_1,'Value')==1)
        message=ones(1,Nint/4e-3);
    elseif(get(handles.rb_0,'Value')==1)
        message=zeros(1,Nint/4e-3);
    elseif(get(handles.rb_10,'Value')==1)
        message=zeros(1,Nint/4e-3);
        for n=1:(length(message)+1)/2
            message(1+2*(n-1))=1;
        end
    elseif(get(handles.rb_hand,'Value')==1)
        message_hand=(get(handles.edit_hand,'String'));
        message_hand=reshape(message_hand,1,length(message_hand));

        if length(message_hand)<Nint/4e-3
            warndlg(['V navigacni zprave chybi ',num2str((Nint/4e-3)-
length(message_hand)),' bitu'],'Length Error');
            return;
        end
        if isnan(str2double(message_hand))
            warndlg('Zadejte prosim ciselny format do Message rucni','Format Er-
ror');
            return;
        else
            message=[];
            for m=1:1:Nint/4e-3
                if message_hand(m)=='1' || message_hand(m)=='0'
                    message=[message str2double(message_hand(m))];
                else
                    warndlg(['Cislo na pozici ',num2str(m),' neni binar-
ni'],'Format Error');
                    return;
                end
            end
        end
    if(get(handles.rb_BP,'Value')==1)
        if(Fif*2<=(1/Tsa) && Fif~=0 && Fif*10>=(1/Tsa))
            [Signal Time]=GALILEO_E1(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, 1, 1.023e6);
        else
            warndlg('Fsa musi byt v intervalu (2*Fif,10*Fif)','Fsa Error');
```



```
        return;
    end
    elseif(get(handles.rb_CE,'Value')==1)
        if(40*get(handles.pop_Fsa,'UserData')<=(1/Tsa) &&
400*get(handles.pop_Fsa,'UserData')>=(1/Tsa))
            [Signal Time]=GALILEO_E1(Tsa, 0, PRN_satellite, message, C_N0, Fd,
Tau, 0, 1.023e6);
        else
            text=['Fsa musi byt v intervalu ('
num2str(40*get(handles.pop_Fsa,'UserData')) ',' num2str(400*get(handles.pop_Fsa,'UserData'))
')'];
            warndlg(text,'Fsa Error');
            return;
        end
    end
    % GALILEO E5
    case 2
        if(get(handles.rb_BP,'Value')==1)
            if(Fif*2<(1/Tsa))
                % Signal=BP_GALILEO_E5(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, 1, 'Tc-doplnit');
            else
                % error
            end
        elseif(get(handles.rb_CE,'Value')==1)
            % Signal=CE_GALILEO_E5(Tsa, PRN_satellite, C_N0, Fd, Tau, 0, 'Tc-
doplnit');
        end
    end
    % GLONASS
    case 3
        switch get(handles.pop_GPS_pasmo,'Value')
            % GLONASS L1
            case 1
                if(get(handles.rb_BP,'Value')==1)
                    if(Fif*2<(1/Tsa))
                        % Signal=BP_GLONASS_L1(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, 1, 'Tc-doplnit');
                    else
                        % error
                    end
                elseif(get(handles.rb_CE,'Value')==1)
                    % Signal=CE_GGLONASS_L1(Tsa, PRN_satellite, C_N0, Fd, Tau, 0, 'Tc-
doplnit');
                end
            end
            % GLONASS L2
            case 2
                if(get(handles.rb_BP,'Value')==1)
                    if(Fif*2<(1/Tsa))
                        % Signal=BP_GGLONASS_L2(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, 1, 'Tc-doplnit');
                    else
                        % error
                    end
                elseif(get(handles.rb_CE,'Value')==1)
                    % Signal=CE_GLONASS_L2(Tsa, PRN_satellite, C_N0, Fd, Tau, 0, 'Tc-
doplnit');
                end
            end
        end
    end
    % ulozeni dat o signalu
    if get(handles.rb_BP,'Value')==1
        save(get(handles.edit_save,'String'),'Fsa','Fif','Nint','C_N0','Fd','Tau','Signal')
    else
        save(get(handles.edit_save,'String'),'Fsa','Nint','C_N0','Fd','Tau','Signal')
    end
end
```



```
% nacteni promennych
Load=load('SIGNAL_SAVE');
% prepnuti na rb_Signal
off=[handles.rb_Signal, handles.rb_ACF];
value_off(off);
on=[handles.rb_PSD];
value_on(on);
% vykresleni
positionFigure=get(handles.figure1,'Position');
positionFigure(1,3)=184;
set(handles.figure1,'Position',positionFigure);
plot(Time./get(handles.pop_Fif,'UserData'), 10*log10(Signal))
title(['Spektrum ' Load.Popisek_BP_CE]);
label_axes=get(handles.pop_Fif,'String');
ylabel('PSD [dB]');
xlabel(['f [' label_axes{get(handles.pop_Fif,'Value')} ']]');

% --- Executes on button press in rb_Signal.
function rb_Signal_Callback(hObject, eventdata, handles)
% hObject    handle to rb_Signal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

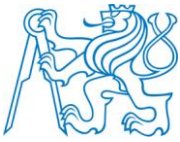
% Hint: get(hObject,'Value') returns toggle state of rb_Signal
off=[handles.rb_PSD, handles.rb_ACF];
value_off(off);
Load=load('SIGNAL_SAVE');
plot(Load.Time_BP_CE, Load.Signal_BP_CE)
title(Load.Popisek_BP_CE);
ylabel('A');
xlabel('t [s]');

% --- Executes on button press in rb_PSD.
function rb_PSD_Callback(hObject, eventdata, handles)
% hObject    handle to rb_PSD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_PSD
off=[handles.rb_Signal, handles.rb_ACF];
value_off(off);
if(get(handles.rb_BP,'Value')==1)
    PSD_BP=11;
elseif(get(handles.rb_CE,'Value')==1)
    PSD_BP=10;
end
Load=load('SIGNAL_SAVE');
[Signal_PSD_ACF Time]=PSD_ACF(Load.Signal_BP_CE, PSD_BP);
handles.figure1;
plot(Time./get(handles.pop_Fif,'UserData'), 10*log10(Signal_PSD_ACF))
title(['Spektrum ' Load.Popisek_BP_CE]);
label_axes=get(handles.pop_Fif,'String');
xlabel(['f [' label_axes{get(handles.pop_Fif,'Value')} ']]');
ylabel('PSD [dB]');

% --- Executes on button press in rb_ACF.
function rb_ACF_Callback(hObject, eventdata, handles)
% hObject    handle to rb_ACF (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_ACF
off=[handles.rb_Signal, handles.rb_PSD];
value_off(off);
if(get(handles.rb_BP,'Value')==1)
    PSD_BP=01;
```



```
elseif(get(handles.rb_CE, 'Value')==1)
    PSD_BP=00;
end
Load=load('SIGNAL_SAVE');
[Signal_PSD_ACF Time]=PSD_ACF(Load.Signal_BP_CE, PSD_BP);
handles.figure1;
plot(Time./get(handles.pop_Fif, 'UserData'), (Signal_PSD_ACF))
title(['Spektrum ' Load.Popisek_BP_CE]);
label_axes=get(handles.pop_Fif, 'String');
xlabel(['f [' label_axes{get(handles.pop_Fif, 'Value')} ']]);
ylabel('ACF');

% --- Executes on button press in rb_load.
function rb_load_Callback(hObject, eventdata, handles)
% hObject    handle to rb_load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_load
off=[handles.rb_1, handles.rb_0, handles.rb_10, handles.rb_hand];
value_off(off);
offed=[handles.text_1, handles.text_0, handles.text_10, handles.edit_hand];
visible_off(offed);
oned=[handles.pb_load];
visible_on(oned);

% --- Executes on button press in rb_1.
function rb_1_Callback(hObject, eventdata, handles)
% hObject    handle to rb_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_1
off=[handles.rb_load, handles.rb_0, handles.rb_10, handles.rb_hand];
value_off(off);
offed=[handles.text_0, handles.text_10, handles.edit_hand, handles.pb_load];
visible_off(offed);
oned=[handles.text_1];
visible_on(oned);
set(handles.pb_vypocet, 'Enable', 'on');

% --- Executes on button press in rb_0.
function rb_0_Callback(hObject, eventdata, handles)
% hObject    handle to rb_0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_0
off=[handles.rb_1, handles.rb_load, handles.rb_10, handles.rb_hand];
value_off(off);
offed=[handles.text_1, handles.text_10, handles.edit_hand, handles.pb_load];
visible_off(offed);
oned=[handles.text_0];
visible_on(oned);
set(handles.pb_vypocet, 'Enable', 'on');

% --- Executes on button press in rb_10.
function rb_10_Callback(hObject, eventdata, handles)
% hObject    handle to rb_10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_10
off=[handles.rb_1, handles.rb_0, handles.rb_load, handles.rb_hand];
value_off(off);
offed=[handles.text_0, handles.text_1, handles.edit_hand, handles.pb_load];
```



```
visible_off(offed);
oned=[handles.text_10];
visible_on(oned);
set(handles.pb_vypocet, 'Enable', 'on');

% --- Executes on button press in rb_hand.
function rb_hand_Callback(hObject, eventdata, handles)
% hObject    handle to rb_hand (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_hand
off=[handles.rb_1, handles.rb_0, handles.rb_10, handles.rb_load];
value_off(off);
offed=[handles.text_0, handles.text_10, handles.text_1, handles.pb_load];
visible_off(offed);
oned=[handles.edit_hand];
visible_on(oned);

function edit_hand_Callback(hObject, eventdata, handles)
% hObject    handle to edit_hand (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_hand as text
%        str2double(get(hObject,'String')) returns contents of edit_hand as a double

% --- Executes during object creation, after setting all properties.
function edit_hand_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_hand (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_C_N0_Callback(hObject, eventdata, handles)
% hObject    handle to edit_C_N0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_C_N0 as text
%        str2double(get(hObject,'String')) returns contents of edit_C_N0 as a double
if str2double(get(handles.edit_C_N0, 'String')) > 1000
    helpdlg('Pri vetsim C/N0 nez 1000 se sum v signalu neprojevi', 'Info C/N0');
    set(handles.edit_C_N0, 'String', '1000');
    return;
end

% --- Executes during object creation, after setting all properties.
function edit_C_N0_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_C_N0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_Tau_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Tau (see GCBO)
```




```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Tau as text
% str2double(get(hObject,'String')) returns contents of edit_Tau as a double

% --- Executes during object creation, after setting all properties.
function edit_Tau_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit_Tau (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_Fd_Callback(hObject, eventdata, handles)
% hObject handle to edit_Fd (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Fd as text
% str2double(get(hObject,'String')) returns contents of edit_Fd as a double

% --- Executes during object creation, after setting all properties.
function edit_Fd_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit_Fd (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in pop_Fd.
function pop_Fd_Callback(hObject, eventdata, handles)
% hObject handle to pop_Fd (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_Fd contents as cell array
% contents{get(hObject,'Value')} returns selected item from pop_Fd
switch get(handles.pop_Fd,'Value')
    case 1
        set(handles.pop_Fd,'UserData',1);
    case 2
        set(handles.pop_Fd,'UserData',1e3);
    case 3
        set(handles.pop_Fd,'UserData',1e6);
    case 4
        set(handles.pop_Fd,'UserData',1e9);
end

% --- Executes during object creation, after setting all properties.
function pop_Fd_CreateFcn(hObject, eventdata, handles)
% hObject handle to pop_Fd (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```



```
        set(hObject,'BackgroundColor','white');
    end

% --- Executes on selection change in pop_Fsa.
function pop_Fsa_Callback(hObject, eventdata, handles)
% hObject      handle to pop_Fsa (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_Fsa contents as cell array
%         contents{get(hObject,'Value')} returns selected item from pop_Fsa
switch get(handles.pop_Fsa,'Value')
    case 1
        set(handles.pop_Fsa,'UserData',1);
    case 2
        set(handles.pop_Fsa,'UserData',1e3);
    case 3
        set(handles.pop_Fsa,'UserData',1e6);
    case 4
        set(handles.pop_Fsa,'UserData',1e9);
end

% --- Executes during object creation, after setting all properties.
function pop_Fsa_CreateFcn(hObject, eventdata, handles)
% hObject      handle to pop_Fsa (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in pop_nav_system.
function pop_nav_system_Callback(hObject, eventdata, handles)
% hObject      handle to pop_nav_system (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_nav_system contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from pop_nav_system

if get(handles.rb_BP,'Value')==1
    BP_CE='BP';
else
    BP_CE='CE';
end
switch get(handles.pop_nav_system,'Value')
    case 1
        offed=[handles.pop_GALILEO_pasmo, handles.pop_GLONASS_pasmo, handles.pop_PRN_GALILEO];
        visible_off(offed);
        oned=[handles.pop_GPS_pasmo, handles.pop_PRN_GPS];
        visible_on(oned);
        path_save=[BP_CE '_GPS_' get(handles.pop_GPS_pasmo,'UserData') '_druzice'
num2str(get(handles.pop_PRN_GPS,'Value'))];
        set(handles.edit_save,'String',path_save);
        set(handles.pop_nav_system,'UserData','GPS');
        switch get(handles.pop_GPS_pasmo,'Value')
            case 1
                set(handles.edit_Fif, 'String', '2');
                set(handles.edit_Tsa_Fsa, 'String', '8.3');
            case 2
                set(handles.edit_Fif, 'String', '2');
                set(handles.edit_Tsa_Fsa, 'String', '8.3');
```



```
        case 3
            set(handles.edit_Fif, 'String', '20');
            set(handles.edit_Tsa_Fsa, 'String', '80.3');
        end
    case 2
        offed=[handles.pop_GPS_pasmo, handles.pop_GLONASS_pasmo, handles.pop_PRN_GPS];
        visible_off(offed);
        oned=[handles.pop_GALILEO_pasmo, handles.pop_PRN_GALILEO];
        visible_on(oned);
        path_save=[BP_CE '_GALILEO_' get(handles.pop_GALILEO_pasmo, 'UserData') '_druzice'
num2str(get(handles.pop_PRN_GALILEO, 'Value'))];
        set(handles.edit_save, 'String', path_save);
        set(handles.pop_nav_system, 'UserData', 'GALILEO');
        set(handles.edit_Fif, 'String', '10');
        set(handles.edit_Tsa_Fsa, 'String', '40.3');
    case 3
        offed=[handles.pop_GALILEO_pasmo, handles.pop_GPS_pasmo, handles.pop_PRN_GALILEO];
        visible_off(offed);
        oned=[handles.pop_GLONASS_pasmo, handles.pop_PRN_GPS];
        visible_on(oned);
        path_save=[BP_CE '_GLONASS_' get(handles.pop_GLONASS_pasmo, 'UserData') '_druzice'
num2str(get(handles.pop_PRN_GPS, 'Value'))];
        set(handles.edit_save, 'String', path_save);
        set(handles.pop_nav_system, 'UserData', 'GLONASS');
end

% --- Executes during object creation, after setting all properties.
function pop_nav_system_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop_nav_system (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

handles = guihandles(hObject);
nav_system=get(handles.pop_nav_system, 'Value');
switch(nav_system)
    case 1
        set(handles.pop_nav_system, 'UserData', 'GPS');
    case 2
        set(handles.pop_nav_system, 'UserData', 'GALILEO');
    case 3
        set(handles.pop_nav_system, 'UserData', 'GLONASS');
end

function edit_Tsa_Fsa_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Tsa_Fsa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit_Tsa_Fsa as text
%         str2double(get(hObject, 'String')) returns contents of edit_Tsa_Fsa as a double

% --- Executes during object creation, after setting all properties.
function edit_Tsa_Fsa_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_Tsa_Fsa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
```



```
        set(hObject,'BackgroundColor','white');
    end

% --- Executes on selection change in pop_PRN_GPS.
function pop_PRN_GPS_Callback(hObject, eventdata, handles)
% hObject    handle to pop_PRN_GPS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_PRN_GPS contents as cell array
%         contents{get(hObject,'Value')} returns selected item from pop_PRN_GPS

if get(handles.rb_BP,'Value')==1
    BP_CE='BP';
else
    BP_CE='CE';
end
path_save=[BP_CE ' ' get(handles.pop_nav_system,'UserData') ' '
get(handles.pop_GPS_pasmo,'UserData') '_druzice' num2str(get(handles.pop_PRN_GPS,'Value'))];
set(handles.edit_save,'String',path_save);

% --- Executes during object creation, after setting all properties.
function pop_PRN_GPS_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop_PRN_GPS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_Tint_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Tint (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Tint as text
%         str2double(get(hObject,'String')) returns contents of edit_Tint as a double

% --- Executes during object creation, after setting all properties.
function edit_Tint_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_Tint (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in pop_GPS_pasmo.
function pop_GPS_pasmo_Callback(hObject, eventdata, handles)
% hObject    handle to pop_GPS_pasmo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_GPS_pasmo contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from pop_GPS_pasmo

if get(handles.rb_BP,'Value')==1
    BP_CE='BP';
else
```



```
BP_CE='CE';
end
switch get(handles.pop_GPS_pasmo, 'Value')
case 1
    set(handles.edit_Fif, 'String', '2');
    set(handles.edit_Tsa_Fsa, 'String', '8.3');
    path_save=[BP_CE '_' get(handles.pop_nav_system, 'UserData') '_L1CA_druzice'
num2str(get(handles.pop_PRN_GPS, 'Value'))];
    set(handles.edit_save, 'String', path_save);
    set(handles.pop_GPS_pasmo, 'UserData', 'L1CA');
case 2
    set(handles.edit_Fif, 'String', '2');
    set(handles.edit_Tsa_Fsa, 'String', '8.3');
    path_save=[BP_CE '_' get(handles.pop_nav_system, 'UserData') '_L2C_druzice'
num2str(get(handles.pop_PRN_GPS, 'Value'))];
    set(handles.edit_save, 'String', path_save);
    set(handles.pop_GPS_pasmo, 'UserData', 'L2C');
case 3
    set(handles.edit_Fif, 'String', '20');
    set(handles.edit_Tsa_Fsa, 'String', '80.3');
    path_save=[BP_CE '_' get(handles.pop_nav_system, 'UserData') '_L5_druzice'
num2str(get(handles.pop_PRN_GPS, 'Value'))];
    set(handles.edit_save, 'String', path_save);
    set(handles.pop_GPS_pasmo, 'UserData', 'L5');
end

% --- Executes during object creation, after setting all properties.
function pop_GPS_pasmo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop_GPS_pasmo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

handles = guihandles(hObject);
nav_system=get(handles.pop_GPS_pasmo, 'Value');
switch(nav_system)
case 1
    set(handles.pop_GPS_pasmo, 'UserData', 'L1CA');
case 2
    set(handles.pop_GPS_pasmo, 'UserData', 'L2C');
case 3
    set(handles.pop_GPS_pasmo, 'UserData', 'L5');
end

function edit_Fif_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Fif (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit_Fif as text
%         str2double(get(hObject, 'String')) returns contents of edit_Fif as a double

% --- Executes during object creation, after setting all properties.
function edit_Fif_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_Fif (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
```



```
set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in pop_Fif.
function pop_Fif_Callback(hObject, eventdata, handles)
% hObject      handle to pop_Fif (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_Fif contents as cell array
%         contents{get(hObject,'Value')} returns selected item from pop_Fif
switch get(handles.pop_Fif,'Value')
    case 1
        set(handles.pop_Fif,'UserData',1);
    case 2
        set(handles.pop_Fif,'UserData',1e3);
    case 3
        set(handles.pop_Fif,'UserData',1e6);
    case 4
        set(handles.pop_Fif,'UserData',1e9);
end

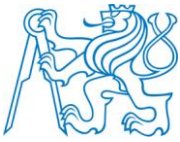
% --- Executes during object creation, after setting all properties.
function pop_Fif_CreateFcn(hObject, eventdata, handles)
% hObject      handle to pop_Fif (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in rb_CE.
function rb_CE_Callback(hObject, eventdata, handles)
% hObject      handle to rb_CE (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_CE
off=[handles.rb_BP];
value_off(off);
offed=[handles.text_Fif, handles.edit_Fif, handles.pop_Fif];
visible_off(offed);

switch get(handles.pop_nav_system,'Value')
    case 1
        path_save=['CE_' get(handles.pop_nav_system,'UserData') '_']
        get(handles.pop_GPS_pasmo,'UserData') '_druzice' num2str(get(handles.pop_PRN_GPS,'Value'));
        set(handles.edit_save,'String',path_save);
        set(handles.pop_nav_system,'UserData','GPS');
    case 2
        path_save=['CE_' get(handles.pop_nav_system,'UserData') '_']
        get(handles.pop_GALILEO_pasmo,'UserData') '_druzice'
        num2str(get(handles.pop_PRN_GALILEO,'Value'));
        set(handles.edit_save,'String',path_save);
        set(handles.pop_nav_system,'UserData','GALILEO');
    case 3
        path_save=['CE_' get(handles.pop_nav_system,'UserData') '_']
        get(handles.pop_GLONASS_pasmo,'UserData') '_druzice'
        num2str(get(handles.pop_PRN_GPS,'Value'));
        set(handles.edit_save,'String',path_save);
        set(handles.pop_nav_system,'UserData','GLONASS');
end
```



```
% --- Executes on button press in rb_BP.
function rb_BP_Callback(hObject, eventdata, handles)
% hObject    handle to rb_BP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_BP
oned=[handles.text_Fif, handles.edit_Fif, handles.pop_Fif];
visible_on(oned);
off=[handles.rb_CE];
value_off(off);

switch get(handles.pop_nav_system,'Value')
    case 1
        path_save=['BP_' get(handles.pop_nav_system,'UserData') '_']
        get(handles.pop_GPS_pasmo,'UserData') '_druzice' num2str(get(handles.pop_PRN_GPS,'Value'));
        set(handles.edit_save,'String',path_save);
        set(handles.pop_nav_system,'UserData','GPS');
    case 2
        path_save=['BP_' get(handles.pop_nav_system,'UserData') '_']
        get(handles.pop_GALILEO_pasmo,'UserData') '_druzice'
        num2str(get(handles.pop_PRN_GALILEO,'Value'));
        set(handles.edit_save,'String',path_save);
        set(handles.pop_nav_system,'UserData','GALILEO');
    case 3
        path_save=['BP_' get(handles.pop_nav_system,'UserData') '_']
        get(handles.pop_GLONASS_pasmo,'UserData') '_druzice'
        num2str(get(handles.pop_PRN_GPS,'Value'));
        set(handles.edit_save,'String',path_save);
        set(handles.pop_nav_system,'UserData','GLONASS');
end

% --- Executes on selection change in pop_GLONASS_pasmo.
function pop_GLONASS_pasmo_Callback(hObject, eventdata, handles)
% hObject    handle to pop_GLONASS_pasmo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_GLONASS_pasmo contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from pop_GLONASS_pasmo

% --- Executes during object creation, after setting all properties.
function pop_GLONASS_pasmo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop_GLONASS_pasmo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in pop_GALILEO_pasmo.
function pop_GALILEO_pasmo_Callback(hObject, eventdata, handles)
% hObject    handle to pop_GALILEO_pasmo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_GALILEO_pasmo contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from pop_GALILEO_pasmo

% --- Executes during object creation, after setting all properties.
function pop_GALILEO_pasmo_CreateFcn(hObject, eventdata, handles)
```



```
% hObject    handle to pop_GALILEO_pasmo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

handles = guihandles(hObject);
set(handles.pop_GALILEO_pasmo,'UserData','E1');

% -----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in rb_Tsa.
function rb_Tsa_Callback(hObject, eventdata, handles)
% hObject    handle to rb_Tsa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_Tsa
offed=[handles.pop_Fsa];
visible_off(offed);
oned=[handles.text_Tsa_s];
visible_on(oned);
off=[handles.rb_Fsa];
value_off(off);

% --- Executes on button press in rb_Fsa.
function rb_Fsa_Callback(hObject, eventdata, handles)
% hObject    handle to rb_Fsa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rb_Fsa
oned=[handles.pop_Fsa];
visible_on(oned);
offed=[handles.text_Tsa_s];
visible_off(offed);
off=[handles.rb_Tsa];
value_off(off);

% --- Executes on selection change in pop_PRN_GALILEO.
function pop_PRN_GALILEO_Callback(hObject, eventdata, handles)
% hObject    handle to pop_PRN_GALILEO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns pop_PRN_GALILEO contents as cell
array
%           contents(get(hObject,'Value')) returns selected item from pop_PRN_GALILEO

if get(handles.rb_BP,'Value')==1
    BP_CE='BP';
else
    BP_CE='CE';
end
path_save=[BP_CE ' ' get(handles.pop_nav_system,'UserData') ' '
get(handles.pop_GALILEO_pasmo,'UserData') ' '
num2str(get(handles.pop_PRN_GALILEO,'Value'))];
set(handles.edit_save,'String',path_save);
```




```
% --- Executes during object creation, after setting all properties.
function pop_PRN_GALILEO_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop_PRN_GALILEO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_save_Callback(hObject, eventdata, handles)
% hObject    handle to edit_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_save as text
%         str2double(get(hObject,'String')) returns contents of edit_save as a double

% --- Executes during object creation, after setting all properties.
function edit_save_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

handles = guihandles(hObject);
path_save='BP_GPS_LICA_druzice1';
set(handles.edit_save,'String',path_save);

% --- Executes on button press in pb_load.
function pb_load_Callback(hObject, eventdata, handles)
% hObject    handle to pb_load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.pb_vypocet, 'Enable', 'off');
[name_message,path_message]=uigetfile({'*.xls'; '*.xlsx'}, 'File Selector', 'Multiselect',
'off');
if name_message==0;
    return;
else
    message=xlsread([path_message,name_message]);
    [m n]=size(message);
    if m>n
        message=message';
    end
    set(handles.pb_load, 'UserData', message);
    set(handles.pb_vypocet, 'Enable', 'on');
end
```



Příloha L - Zdrojový kód GPS_L1_CA.m

```
function [Signal_PSD_ACF Time_PSD_ACF]=GPS_L1_CA(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, BP, Fc)
%% Konstanty
Tc=1/Fc;
Tm=20e-3;
Tint=fix((length(message)*Tm)/Tsa);
Time_BP_CE=(0:Tint-1)*Tsa;
%% Generace PRN kodu
PRN_code=Generate_PRN_GPS_L1_CA(PRN_satellite);
%% Namapovani PRN a message
message=((message.*2)-1);
PRN_code=((PRN_code.*2)-1);
%% zopakovani PRN a message do delky Nint
Tint_PRN_code=PRN_code(mod(fix(Time_BP_CE/Tc), length(PRN_code))+1);
Tint_message=message(mod(fix(Time_BP_CE/Tm), length(message))+1);
%% Posunutí PRN a message o Tau
Shift_Tint_PRN_code=[Tint_PRN_code(Tau+1:end) Tint_PRN_code(1:Tau)];
Shift_Tint_message=[Tint_message(Tau+1:end) Tint_message(1:Tau)];
%% BP nebo CE signal s Tau
Signal=Shift_Tint_message.*Shift_Tint_PRN_code;
if(BP==1)
    Signal_BP_CE=Signal.*sin(2*pi*(Fif+Fd).*Time_BP_CE);
    Popisek_BP_CE=('Pásmový signál GPS L1 C/A');
else
    Signal_BP_CE=Signal.*exp(2i*pi*Fd*Time_BP_CE);
    Popisek_BP_CE=('Komplexní obálka GPS L1 C/A');
end
%% Signal + noise
if(C_N0~=1000)
    Signal_BP_CE=Noise_Signal(Signal_BP_CE, C_N0, BP, Tsa);
end
%% ulozeni dat
save('SIGNAL_SAVE', 'Signal_BP_CE', 'Time_BP_CE', 'Popisek_BP_CE', 'Tsa');
%% PSD
[Signal_PSD_ACF Time_PSD_ACF]=PSD_ACF(Signal_BP_CE, 10+BP);
end
```



Příloha M - Zdrojový kód GPS_L2_C.m

```
function [Signal_PSD_ACF Time_PSD_ACF]=GPS_L2_C(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, BP, Fc)
%% Konstanty
Tc=1/Fc;
Tm=20e-3;
Nint=fix((length(message)*Tm)/Tsa);
Time_BP_CE=(0:Nint-1)*Tsa;
%% Generace PRN
[L2CM L2CL]=Generate_PRN_GPS_L2_C(PRN_satellite);
%% Tvorba L2C
L2CM=L2CM(mod(1:length(L2CL), length(L2CM))+1);
L2C=[L2CM; L2CL];
L2C=L2C(:)';
%% PRN a message na Time_BP_CE
L2C_Time=L2C(mod(fix(Time_BP_CE/(Tc/2)), length(L2C))+1);
message_Time=message(mod(fix(Time_BP_CE/Tm), length(message))+1);
%% Shift PRN a message
L2C_Time_Shift=[L2C_Time(Tau+1:end) L2C_Time(1:Tau)];
message_Time_Shift=[message_Time(Tau+1:end) message_Time(1:Tau)];
%% Namapovani L2C a message
L2C_Time_Shift=((L2C_Time_Shift.*2)-1);
message_Time_Shift=((message_Time_Shift.*2)-1);
%% PRN a message
L2C=L2C_Time_Shift.*message_Time_Shift;
%% BP nebo CE signal s Tau
if(BP==1)
    Signal_BP_CE=L2C.*sin(2*pi*(Fif+Fd).*Time_BP_CE);
    Popisek_BP_CE=('Pásmový signál GPS L2 C');
else
    Signal_BP_CE=L2C.*exp(2i*pi*Fd*Time_BP_CE);
    Popisek_BP_CE=('Komplexní obálka GPS L2 C');
end
%% Signal + noise
if(C_N0~=1000)
    Signal_BP_CE=Noise_Signal(Signal_BP_CE, C_N0, BP, Tsa);
end
%% ulozeni dat
save('SIGNAL_SAVE', 'Signal_BP_CE', 'Time_BP_CE', 'Popisek_BP_CE', 'Tsa');
%% PSD
[Signal_PSD_ACF Time_PSD_ACF]=PSD_ACF(Signal_BP_CE, 10+BP);
end
```



Příloha N - Zdrojový kód GPS_L5.m

```
function [Signal_PSD_ACF Time_PSD_ACF]=GPS_L5(Tsa, Fif, PRN_satellite, message, C_N0, Fd, Tau, BP, Fc)
%% Konstanty
Tc=1/Fc;
Tm=10e-3;
Nint=fix((length(message)*Tm)/Tsa);
Time_BP_CE=(0:Nint-1)*Tsa;
NH_code_I5=[0 0 0 0 1 1 0 1 0 1];
NH_code_Q5=[0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 1 1 1 0];
Tnh_I5=1e-3;
Tnh_Q5=2e-3;
%% Generace PRN kódu
[PRN_I5 PRN_Q5]=Generate_PRN_GPS_L5(PRN_satellite);
%% kody na Time_BP_CE
NH_I5_Time=NH_code_I5(mod(fix(Time_BP_CE/Tnh_I5), length(NH_code_I5))+1);
NH_Q5_Time=NH_code_Q5(mod(fix(Time_BP_CE/Tnh_Q5), length(NH_code_Q5))+1);
message_Time=message(mod(fix(Time_BP_CE/Tm), length(message))+1);
PRN_I5_Time=PRN_I5(mod(fix(Time_BP_CE/Tc), length(PRN_I5))+1);
PRN_Q5_Time=PRN_Q5(mod(fix(Time_BP_CE/Tc), length(PRN_Q5))+1);
%% message s NH code
message_NH_I5=xor(NH_I5_Time, message_Time);
message_NH_Q5=xor(NH_Q5_Time, message_Time);
%% Navigační zprava a PRN
I5=xor(PRN_I5_Time, message_NH_I5);
Q5=xor(PRN_Q5_Time, message_NH_Q5);
%% Shift PRN
Shift_I5=[I5(Tau+1:length(I5)) I5(1:Tau)];
Shift_Q5=[Q5(Tau+1:length(Q5)) Q5(1:Tau)];
%% Namapování I5 a Q5
I5=((Shift_I5.*2)-1);
Q5=((Shift_Q5.*2)-1);
%% modulace
CE_Signal=(I5+1i*Q5).*exp(2i*pi*Fd.*Time_BP_CE);
if BP==1
    Signal_BP_CE=real(CE_Signal.*exp(2i*pi*(Fif+Fd).*Time_BP_CE));
    Popisek_BP_CE=('Pásmový signál GPS L5');
else
    Signal_BP_CE=CE_Signal;
    Popisek_BP_CE=('Komplexní obálka GPS L5');
end
%% Signal + noise
if(C_N0~=1000)
    Signal_BP_CE=Noise_Signal(Signal_BP_CE, C_N0, BP, Tsa);
end
%% uložení dat
save('SIGNAL_SAVE', 'Signal_BP_CE', 'Time_BP_CE', 'Popisek_BP_CE', 'Tsa');
%% PSD
[Signal_PSD_ACF Time_PSD_ACF]=PSD_ACF(Signal_BP_CE, 10+BP);
end
```



Zdrojový kód GALILEO_E1.m

```
function [Signal_PSD_ACF Time_PSD_ACF]=GALILEO_E1(Tsa, Fif, PRN_satellite, message, C_N0, Fd,
Tau, BP, Fc)
%% Konstanty
Tc=1/Fc;
Tm=4e-3;
alpha=sqrt(10/11);
beta=sqrt(1/11);
Tsec=100e-3;
Nint=fix((length(message)*Tm)/Tsa);
Time_BP_CE=(0:Nint-1)*Tsa;
oneT_CBOC_E1B=[alpha+beta alpha-beta alpha+beta alpha-beta alpha+beta alpha-beta ...
-alpha+beta -alpha-beta -alpha+beta -alpha-beta -alpha+beta -alpha-beta];
oneT_CBOC_E1C=[alpha-beta alpha+beta alpha-beta alpha+beta alpha-beta alpha+beta ...
-alpha-beta -alpha+beta -alpha-beta -alpha+beta -alpha-beta -alpha+beta];
%% Generace PRN
[E1B_primar E1C_primar E1C_secondar]=Generate_PRN_GALILEO_E1(PRN_satellite);
%% kody E1B, E1C, message, oneT_CBOC na Time_BP_CE
E1C_secondar_Time=E1C_secondar(mod(fix(Time_BP_CE/(Tc)), length(E1C_secondar))+1);
E1C_primar_Time=E1C_primar(mod(fix(Time_BP_CE/(Tm)), length(E1C_primar))+1);
E1B_primar_Time=E1B_primar(mod(fix(Time_BP_CE/(Tc)), length(E1B_primar))+1);
message_Time=message(mod(fix(Time_BP_CE/(Tm)), length(message))+1);
oneT_CBOC_E1B_Time=oneT_CBOC_E1B(mod(fix(Time_BP_CE/(Tc/12)), length(oneT_CBOC_E1B))+1);
oneT_CBOC_E1C_Time=oneT_CBOC_E1C(mod(fix(Time_BP_CE/(Tc/12)), length(oneT_CBOC_E1C))+1);
%% kody E1B_Time a E1C_Time
E1C_Time=xor(E1C_primar_Time, E1C_secondar_Time);
E1B_Time=xor(E1B_primar_Time, message_Time);
%% namapovani E1B_Time a E1C_Time
E1C_Time=(E1C_Time.*2)-1;
E1B_Time=(E1B_Time.*2)-1;
%% CBOC
E1C_Time_CBOC=(E1C_Time.*oneT_CBOC_E1C_Time);
E1B_Time_CBOC=(E1B_Time.*oneT_CBOC_E1B_Time);
%% posunutí o Tau
E1C_Time_Shift=[E1C_Time_CBOC(Tau+1:end) E1C_Time_CBOC(1:Tau)];
E1B_Time_Shift=[E1B_Time_CBOC(Tau+1:end) E1B_Time_CBOC(1:Tau)];
%% modulace
Signal_BP_CE=(1/sqrt(2)).*(E1B_Time_Shift-E1C_Time_Shift);
if(BP==1)
Signal_BP_CE=Signal_BP_CE.*sin(2*pi*(Fif+Fd).*Time_BP_CE);
Popisek_BP_CE=('Pásmový signál GALILEO E1');
else
Signal_BP_CE=Signal_BP_CE.*exp(2i*pi*Fd*Time_BP_CE);
Popisek_BP_CE=('Komplexní obálka GALILEO E1');
end
%% Signal + noise
if(C_N0~=1000)
Signal_BP_CE=Noise_Signal(Signal_BP_CE, C_N0, BP, Tsa);
end
%% uložení dat
save('SIGNAL_SAVE', 'Signal_BP_CE', 'Time_BP_CE', 'Popisek_BP_CE', 'Tsa');
%% PSD
[Signal_PSD_ACF Time_PSD_ACF]=PSD_ACF(Signal_BP_CE, 10+BP);
end
```



Zdrojový kód Generate_PRN_GPS_L1_CA.m

```
function PRN_code=Generate_PRN_GPS_L1_CA(PRN_satellite)
PRN=zeros(1,1023);

num_Satellite=[2 3 4 5 1 2 1 2 3 2 3 5 6 7 8 9 1 2 3 4,...
               5 6 1 4 5 6 7 8 1 2 3 4 5 4 1 2 4;
               6 7 8 9 9 10 8 9 10 3 4 6 7 8 9 10 4 5 6 7,...
               8 9 3 6 7 8 9 10 6 7 8 9 10 10 7 8 10];

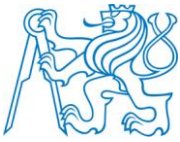
g1=ones(1,10);
g2=ones(1,10);

g21=num_Satellite(1,PRN_satellite);
g22=num_Satellite(2,PRN_satellite);

for m=1:1023
    last_G1=g1(10);
    first_G1=xor(g1(3),g1(10));
    first_G2=xor(g2(2),xor(g2(3),xor(g2(6),xor(g2(8),xor(g2(9),g2(10))))));
    selector=xor(g2(g21),g2(g22));

    g1=[first_G1 g1(1:9)];
    g2=[first_G2 g2(1:9)];

    PRN(1,m)=xor(selector,last_G1);
end
PRN_code=PRN;
end
```



Příloha O - Zdrojový kód Generate_PRN_GPS_L2_C.m

```
function [L2CM L2CL]=Generate_PRN_GPS_L2_C(PRN_satellite)
%% Konstanty
L2CM=zeros(1,10230);
L2CL=zeros(1,767250);
%% Inicializace L2CM a L2CL pomoci PRN_satellite
Initial_L2CM=[1 1 1 1 0 0 0 1 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 0 1 1 0 1 0 0];
[1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 1];
[0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0];
[0 0 0 1 1 0 1 1 0 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 1 0 0];
[1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1];
[1 1 1 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1];
[0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0];
[1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 0 1 1];
[0 0 0 1 0 0 1 1 1 1 0 1 1 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 1 0];
[1 1 1 0 1 1 0 1 1 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0];
[1 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0];
[0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0];
[0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1];
[0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 0 1];
[0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0];
[0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0];
[1 0 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0];
[0 1 0 0 0 0 1 0 1 1 0 1 0 1 0 1 0 0 0 1 1 1 1 0 0 0 1 0 1 0 1];
[0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1 0 1 1 1 0 0 1];
[0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1 0 1 1 0 1 1 0 1 1];
[1 1 1 0 1 0 1 0 0 1 1 1 1 1 0 0 0 1 1 1 0 0 0 1 1 0 1 1 1 1 1];
[1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1 0 0 0];
[1 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1 0 1 1 1 0 0 1 1 0 0];
[0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 0 0];
[1 1 1 0 0 1 0 1 1 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0];
[1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0 0 0 0 1 1 1 0 0 1];
[0 1 1 0 0 1 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 0 1 1 1 0 0 1 1 0 0];
[1 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1];
[1 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 1];
[0 0 0 1 0 1 0 0 0 0 1 1 1 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 0 1 1];
[1 0 1 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 1 1 1 1 1 0 0 0 0 1 1 1];
[1 1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 0];
[0 0 1 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1 0 0 0 1];
[1 1 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 1 0 1 0 1 1];
[1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0];
Initial_L2CL=[1 1 0 0 1 0 1 0 0 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 1 0 1 0];
[1 0 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 0 0 1 1 1 1 0 0 1 0 1 0];
[0 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0];
[1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0];
[0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 1 0 1];
[0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0];
[1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 1 1 1 1 0];
[0 1 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1];
[0 0 0 0 0 1 1 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0];
[1 0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 1 1 1 1 1 1 0];
[0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 1 0 1 0 1 0 1 0];
[0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 0 0];
[1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 0 1 1 0];
[0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1];
[0 0 0 1 0 0 1 1 0 1 0 1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 0 1];
[1 1 0 1 1 1 0 0 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0 1 0 0 0];
[1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0];
[0 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 0 0 1 0 0 0 0 1 1 1];
[1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1];
[0 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1];
```



```
[0 0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1];
[1 0 1 0 0 0 0 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 0 1 1 1 0];
[1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0 0 0 1 1];
[1 1 0 0 0 1 1 0 1 1 0 1 0 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1];
[1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0];
[1 1 1 0 1 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 1 1 1 0];
[1 1 1 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 1];
[0 1 0 0 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 1 1 0 0];
[0 0 1 0 1 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1 0 1 1 0 0 0 1];
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 1 0 0 1 0 0 1 0];
[0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0];
[0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0 0 1 1];
[1 0 0 0 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 0 1 1 1 0 0 0 0];
[1 1 1 0 0 1 1 1 1 0 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0];
[0 1 0 0 1 0 0 1 0 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 1 1 1];
[1 0 1 1 1 0 0 0 1 0 0 1 0 1 0 0 1 1 0 1 1 0 0 0 1 1 1];
[0 1 0 1 0 0 0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 1 1 0 1 1];
reg_L2CM_code=Initial_L2CM(PRN_satellite,1:end);
reg_L2CL_code=Initial_L2CL(PRN_satellite,1:end);
%% L2CM coder
for m=1:length(L2CM)
    last_reg_L2CM_code=reg_L2CM_code(end);
    reg_L2CM_code(3)=xor(last_reg_L2CM_code,reg_L2CM_code(3));
    reg_L2CM_code(6)=xor(last_reg_L2CM_code,reg_L2CM_code(6));
    reg_L2CM_code(8)=xor(last_reg_L2CM_code,reg_L2CM_code(8));
    reg_L2CM_code(11)=xor(last_reg_L2CM_code,reg_L2CM_code(11));
    reg_L2CM_code(14)=xor(last_reg_L2CM_code,reg_L2CM_code(14));
    reg_L2CM_code(16)=xor(last_reg_L2CM_code,reg_L2CM_code(16));
    reg_L2CM_code(18)=xor(last_reg_L2CM_code,reg_L2CM_code(18));
    reg_L2CM_code(21)=xor(last_reg_L2CM_code,reg_L2CM_code(21));
    reg_L2CM_code(22)=xor(last_reg_L2CM_code,reg_L2CM_code(22));
    reg_L2CM_code(23)=xor(last_reg_L2CM_code,reg_L2CM_code(23));
    reg_L2CM_code(24)=xor(last_reg_L2CM_code,reg_L2CM_code(24));
    reg_L2CM_code=[last_reg_L2CM_code reg_L2CM_code(1:end-1)];
    L2CM(m)=last_reg_L2CM_code;
end
%% L2CL coder
for m=1:length(L2CL)
    last_reg_L2CL_code=reg_L2CL_code(end);
    reg_L2CL_code(3)=xor(last_reg_L2CL_code,reg_L2CL_code(3));
    reg_L2CL_code(6)=xor(last_reg_L2CL_code,reg_L2CL_code(6));
    reg_L2CL_code(8)=xor(last_reg_L2CL_code,reg_L2CL_code(8));
    reg_L2CL_code(11)=xor(last_reg_L2CL_code,reg_L2CL_code(11));
    reg_L2CL_code(14)=xor(last_reg_L2CL_code,reg_L2CL_code(14));
    reg_L2CL_code(16)=xor(last_reg_L2CL_code,reg_L2CL_code(16));
    reg_L2CL_code(18)=xor(last_reg_L2CL_code,reg_L2CL_code(18));
    reg_L2CL_code(21)=xor(last_reg_L2CL_code,reg_L2CL_code(21));
    reg_L2CL_code(22)=xor(last_reg_L2CL_code,reg_L2CL_code(22));
    reg_L2CL_code(23)=xor(last_reg_L2CL_code,reg_L2CL_code(23));
    reg_L2CL_code(24)=xor(last_reg_L2CL_code,reg_L2CL_code(24));
    reg_L2CL_code=[last_reg_L2CL_code reg_L2CL_code(1:end-1)];
    L2CL(m)=last_reg_L2CL_code;
end
end
```




Příloha P - Zdrojový kód Generate_PRN_GPS_L5.m

```
function [PRN_I5 PRN_Q5]=Generate_PRN_GPS_L5(PRN_satellite)
%% constant
Time=(0:10230-1);
shift_reg_XA=ones(1, 13);
XA_code=zeros(1,890);
XBi_code=zeros(1,891);
XBq_code=zeros(1,891);
%% inicializace XBI a XBQ pomoci PRN_satellite
num_satellite=[
[0 1 0 1 0 1 1 1 1 0 0 1 0 0] [1 0 0 1 0 1 1 1 0 0 1 1 0 0];
[1 1 0 0 0 0 0 1 1 0 1 0 1] [0 1 0 0 0 1 1 1 1 0 1 1 0];
[0 1 0 0 0 0 0 0 0 1 0 0 0] [1 1 1 1 0 0 0 1 0 0 0 1 1];
[1 0 1 1 0 0 0 1 0 0 1 1 0] [0 0 1 1 1 0 1 1 0 1 0 1 0];
[1 1 1 0 1 1 1 0 1 0 1 1 1] [0 0 1 1 1 1 0 1 1 0 0 1 0];
[0 1 1 0 0 1 1 1 1 1 0 1 0] [0 1 0 1 0 1 0 1 0 1 0 1 0];
[1 0 1 0 0 1 0 0 1 1 1 1 1] [1 1 1 1 1 1 0 0 0 0 0 0 1];
[1 0 1 1 1 1 0 1 0 1 0 0 1] [0 1 1 0 1 0 1 1 0 1 0 1 0];
[1 1 1 1 1 0 0 1 0 1 0 1 1] [1 0 1 1 1 0 1 0 0 0 0 1 1];
[0 1 1 1 1 1 1 0 1 1 1 1 0] [0 0 1 0 0 1 0 0 0 0 1 1 0];
[0 0 0 0 1 0 0 1 1 1 0 1 0] [0 0 0 1 0 0 0 0 0 0 1 0 1];
[1 1 1 0 0 1 1 1 1 1 0 0 1] [0 1 0 1 0 1 1 0 0 0 1 0 1];
[0 0 0 1 1 1 0 0 1 1 1 0 0] [0 1 0 0 1 1 0 1 0 0 1 0 1];
[0 1 0 0 0 0 1 0 0 1 1 1 1] [1 0 1 0 0 0 0 1 1 1 1 1 1];
[0 1 1 0 1 0 1 0 1 1 0 1 0] [1 0 1 1 1 1 0 0 0 1 1 1 1];
[0 0 0 1 1 1 1 0 0 1 0 0 1] [1 1 0 1 0 0 1 0 1 1 1 1 1];
[0 1 0 0 1 1 0 0 0 1 1 1 1] [1 1 1 0 0 1 1 0 0 1 0 0 0];
[1 1 1 1 0 0 0 0 1 1 1 1 0] [1 0 1 1 0 1 1 1 0 0 1 0 0];
[1 1 0 0 1 0 0 0 1 1 1 1 1] [0 0 1 1 0 0 1 0 1 1 0 1 1];
[0 1 1 0 1 0 1 1 0 1 1 0 1] [1 1 0 0 0 0 1 1 1 0 0 0 1];
[0 0 1 0 0 0 0 0 0 1 0 0 0] [0 1 1 0 1 1 0 0 1 0 0 0 0];
[1 1 1 0 1 1 1 1 0 1 1 1 1] [0 0 1 0 1 1 0 0 0 1 1 1 0];
[1 0 0 0 0 1 1 1 1 1 1 1 0] [1 0 0 0 1 0 1 1 1 1 1 0 1];
[1 1 0 0 0 1 0 1 1 0 1 0 0] [0 1 1 0 1 1 1 1 1 0 0 1 1];
[1 1 0 1 0 0 1 1 0 1 1 0 1] [0 1 0 0 0 1 0 0 1 1 0 1 1];
[1 0 1 0 1 1 0 0 1 0 1 1 0] [0 1 0 1 0 1 0 1 1 1 1 0 0];
[0 1 0 1 0 1 1 0 1 1 1 1 0] [1 0 0 0 0 1 1 1 1 1 0 1 0];
[0 1 1 1 1 0 1 0 1 0 1 1 0] [1 1 1 1 1 0 1 0 0 0 0 1 0];
[0 1 0 1 1 1 1 1 0 0 0 0 1] [0 1 0 1 0 0 0 1 0 0 1 0 0];
[1 0 0 0 0 1 0 1 1 0 1 1 1] [1 0 0 0 0 0 1 1 1 1 0 0 1];
[0 0 0 1 0 1 0 0 1 1 1 1 0] [0 1 0 1 1 1 1 1 0 0 1 0 1];
[0 0 0 0 0 1 0 1 1 1 0 0 1] [1 0 0 1 0 0 0 1 0 1 0 1 0];
[1 1 0 1 0 1 0 0 0 0 0 0 1] [1 0 1 1 0 0 1 0 0 1 0 0 0];
[1 1 0 1 1 1 1 1 1 1 0 0 1] [1 1 1 1 0 0 1 0 0 0 1 0 0];
[1 1 1 1 0 1 1 0 1 1 1 0 0] [0 1 1 0 0 1 0 1 1 0 0 1 1];
[1 0 0 1 0 1 1 0 0 1 0 0 0] [0 0 1 1 1 1 0 1 0 1 1 1 1];
[0 0 1 1 0 1 0 0 1 0 0 0 0] [0 0 1 0 0 1 1 0 1 0 0 0 1]];
shift_reg_XBi=num_satellite(PRN_satellite, 1:13);
shift_reg_XBq=num_satellite(PRN_satellite, 14:26);
%% XA coder
for k=1:1:890
    last_shift_reg_XA=shift_reg_XA(end);
xor_shift_reg_XA=xor(shift_reg_XA(13),xor(shift_reg_XA(12),xor(shift_reg_XA(10),shift_reg_XA(9)
)))));
    shift_reg_XA=[xor_shift_reg_XA shift_reg_XA(1:12)];
    XA_code(1,k)=last_shift_reg_XA;
end
%% XBI a XBQ coder
for k=1:1:891
    last_shift_reg_XBi=shift_reg_XBi(end);
    last_shift_reg_XBq=shift_reg_XBq(end);
    xor_shift_reg_XBi=xor(shift_reg_XBi(13),xor(shift_reg_XBi(12), ...
        xor(shift_reg_XBi(8),xor(shift_reg_XBi(7),xor(shift_reg_XBi(6), ...
```



```
        xor(shift_reg_XBi(4), xor(shift_reg_XBi(1), shift_reg_XBi(3))))));
xor_shift_reg_XBq=xor(shift_reg_XBq(13),xor(shift_reg_XBq(12), ...
        xor(shift_reg_XBq(8),xor(shift_reg_XBq(7),xor(shift_reg_XBq(6), ...
        xor(shift_reg_XBq(4), xor(shift_reg_XBq(1), shift_reg_XBq(3))))));
shift_reg_XBi=[xor_shift_reg_XBi shift_reg_XBi(1:12)];
shift_reg_XBq=[xor_shift_reg_XBq shift_reg_XBq(1:12)];
XBi_code(1,k)=last_shift_reg_XBi;
XBq_code(1,k)=last_shift_reg_XBq;
end
%% XA, XBi a XBq prodlouzeni na Tint_code
XA_code_Tint=XA_code(mod(Time, length(XA_code))+1);
XBi_code_Tint=XBi_code(mod(Time, length(XBi_code))+1);
XBq_code_Tint=XBq_code(mod(Time, length(XBq_code))+1);
%% XBi a XBq code
PRN_I5=xor(XA_code_Tint, XBi_code_Tint);
PRN_Q5=xor(XA_code_Tint, XBq_code_Tint);
end
```



Příloha Q - Zdrojový kód Generate_PRN_GALILEO_E1.m

```
function [E1B_primär E1C_primär E1C_secundär]=Generate_PRN_GALILEO_E1 (PRN_satellite)
%% konstanty
E1C_reg1_primär=zeros(1,4092);
E1C_reg2_primär=zeros(1,4092);
E1B_reg1_primär=zeros(1,4092);
E1B_reg2_primär=zeros(1,4092);
initial_E1C_reg1_primär=ones(1,13);
initial_E1B_reg1_primär=ones(1,13);
%% Initial reg2 E1C a E1B
Initial_reg=[[1 1 0 1 1 1 0 0 0 0 0 1 1] [1 1 0 0 1 1 0 0 0 0 0 1 1] ...
            [0 0 1 1 1 0 0 0 0 0 0 1 0] [0 1 0 1 1 0 1 1 0 0 1 0];
            [1 0 0 1 1 1 1 1 1 1 0 0] [0 1 0 0 0 1 0 1 1 1 1 1] ...
            [0 0 1 1 1 0 0 0 0 0 0 1 0] [0 1 0 1 1 0 1 1 0 1 1 0 0 1 0]];
initial_E1C_reg2_primär=Initial_reg (PRN_satellite,1:13);
initial_E1B_reg2_primär=Initial_reg (PRN_satellite,14:26);
E1C_secundär=Initial_reg (PRN_satellite,27:end);
%% Generace primär code E1B a E1C
for m=1:1:4092
    last_E1B_reg1_primär=initial_E1B_reg1_primär(end);
    last_E1B_reg2_primär=initial_E1B_reg2_primär(end);
    last_E1C_reg1_primär=initial_E1C_reg1_primär(end);
    last_E1C_reg2_primär=initial_E1C_reg2_primär(end);

    first_E1B_reg1_primär=xor(initial_E1B_reg1_primär(4),xor(initial_E1B_reg1_primär(5),...
        xor(initial_E1B_reg1_primär(7),xor(initial_E1B_reg1_primär(9),...
        xor(initial_E1B_reg1_primär(10),initial_E1B_reg1_primär(13))))));

    first_E1B_reg2_primär=xor(initial_E1B_reg2_primär(5),xor(initial_E1B_reg2_primär(6),...
        xor(initial_E1B_reg2_primär(7),xor(initial_E1B_reg2_primär(8),...
        xor(initial_E1B_reg2_primär(12),initial_E1B_reg2_primär(13))))));

    first_E1C_reg1_primär=xor(initial_E1C_reg1_primär(4),xor(initial_E1C_reg1_primär(1),...
        xor(initial_E1C_reg1_primär(3),initial_E1C_reg1_primär(13))));

    first_E1C_reg2_primär=xor(initial_E1C_reg2_primär(4),xor(initial_E1C_reg2_primär(5),...
        xor(initial_E1C_reg2_primär(7),xor(initial_E1C_reg2_primär(9),...
        xor(initial_E1C_reg2_primär(10),initial_E1C_reg2_primär(13))))));
    initial_E1B_reg1_primär=[first_E1B_reg1_primär initial_E1B_reg1_primär(1:12)];
    initial_E1B_reg2_primär=[first_E1B_reg2_primär initial_E1B_reg2_primär(1:12)];
    initial_E1C_reg1_primär=[first_E1C_reg1_primär initial_E1C_reg1_primär(1:12)];
    initial_E1C_reg2_primär=[first_E1C_reg2_primär initial_E1C_reg2_primär(1:12)];
    E1B_reg1_primär(m)=last_E1B_reg1_primär;
    E1B_reg2_primär(m)=last_E1B_reg2_primär;
    E1C_reg1_primär(m)=last_E1C_reg1_primär;
    E1C_reg2_primär(m)=last_E1C_reg2_primär;
end
E1B_primär=xor(E1B_reg1_primär,E1B_reg2_primär);
E1C_primär=xor(E1C_reg1_primär,E1C_reg2_primär);
end
```



Příloha R - Zdrojový kód Noise_Signal.m

```
function Signal_BP_CE_Noise=Noise_Signal(Signal_BP_CE, C_N0, BP, Tsa)
    if BP==1
        P_Signal_BP_CE=var(Signal_BP_CE);
        Noise=randn(1,length(Signal_BP_CE));
        P_Noise =sqrt((P_Signal_BP_CE/(10.^(C_N0/10)))*(1/Tsa)/2);
        Signal_BP_CE_Noise =Signal_BP_CE + P_Noise.*(Noise);
    else
        P_Signal_BP_CE=var(Signal_BP_CE);
        Noise=(randn(1,length(Signal_BP_CE)))+(1i.*randn(1,length(Signal_BP_CE)));
        P_Noise=P_Signal_BP_CE/(10^(C_N0/10))*(1/Tsa);
        Signal_BP_CE_Noise=Signal_BP_CE + sqrt(P_Noise).*(Noise);
    end
end
```



Příloha S - Zdrojový kód PSD_ACF.m

```
function [Signal_PSD_ACF, Time_PSD_ACF]=PSD_ACF(Signal_BP_CE, PSD_BP)
Load=load('SIGNAL_SAVE');

switch PSD_BP
case 11
    Signal_PSD_ACF=pwelch(Signal_BP_CE, 100, [], [], 1/Load.Tsa, 'twosided');
    Time_PSD_ACF=(-fix(length(Signal_PSD_ACF)/2):fix((length(Signal_PSD_ACF)-
1)/2))*(1/Load.Tsa)/length(Signal_PSD_ACF);
case 10
    Signal_PSD_ACF=pwelch(Signal_BP_CE, 100, [], [], 1/Load.Tsa, 'twosided');
    Time_PSD_ACF=(-fix(length(Signal_PSD_ACF)/2):fix((length(Signal_PSD_ACF)-
1)/2))*(1/Load.Tsa)/length(Signal_PSD_ACF);
    Signal_PSD_ACF=fftshift(Signal_PSD_ACF);
case 01
    Signal_PSD_ACF=xcorr(Signal_BP_CE);
    Time_PSD_ACF=(-fix(length(Signal_PSD_ACF)/2):fix((length(Signal_PSD_ACF)-
1)/2))*(1/Load.Tsa)/length(Signal_PSD_ACF);
case 00
    Signal_PSD_ACF=xcorr(Signal_BP_CE);
    Time_PSD_ACF=(-fix(length(Signal_PSD_ACF)/2):fix((length(Signal_PSD_ACF)-
1)/2))*(1/Load.Tsa)/length(Signal_PSD_ACF);
end
end
```