

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE BEZDRÁTOVÉHO PŘENOSOVÉHO SYSTÉMU  
NA USRP

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

MICHAL VACLÍK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## NÁVRH A REALIZACE BEZDRÁTOVÉHO PŘENOSOVÉHO SYSTÉMU NA USRP

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL VACLÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ KRAJSA, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor  
Teleinformatika

**Student:** Michal Vaclík

**ID:** 154225

**Ročník:** 3

**Akademický rok:** 2014/2015

## NÁZEV TÉMATU:

**Návrh a realizace bezdrátového přenosového systému na USRP**

## POKYNY PRO VYPRACOVÁNÍ:

Pomocí vybraného vývojového nástroje realizujte obousměrný přenosový systém založený na softwarově definovaném rádiu, konkrétně na USRP1. Navržené řešení bude modulární s možností změny základních parametrů jako je kódování, modulace, přenosová rychlost.

## DOPORUČENÁ LITERATURA:

[1] ROUPHAEL, Tony J. RF and digital signal processing for software-defined radio: a multi-standard multi-mode approach. Amsterdam: Newnes, 2009, ix, 383 s. ISBN 978-0-7506-8210-7.

**Termín zadání:** 9.2.2015

**Termín odevzdání:** 2.6.2015

**Vedoucí práce:** Ing. Ondřej Krajsa, Ph.D.

**Konzultanti bakalářské práce:**

**doc. Ing. Jiří Mišurec, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

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

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh a realizace bezdrátového přenosového systému na USRP“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Ondřeji Krajsovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)



Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Purkynova 118, CZ-61200 Brno  
Czech Republic  
<http://www.six.feec.vutbr.cz>

## PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....

podpis autora(-ky)



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# OBSAH

Úvod	11
<b>1 Bezdrátová komunikace</b>	<b>12</b>
1.1 Digitální modulace (klíčování)	12
1.1.1 Gausovské klíčování s minimálním zdvihem (GMSK)	13
1.1.2 Klíčování fázovým posuvem (PSK)	13
1.1.3 Kvadraturní amplitudová modulace (QAM)	15
1.1.4 Ortogonální multiplex s frekvenčním dělením	16
1.2 Synchronizace	17
1.2.1 Synchronizace v OFDM systémech	17
1.2.2 Metody obnovení nosné vlny	18
1.2.3 Obnova časování symbolů	19
1.3 Kanálová kódování	19
1.3.1 Konvoluční kódy	20
1.3.2 Prokládání (Interleaving)	20
<b>2 Koncepte softwarového rádia</b>	<b>21</b>
2.1 Vývojový kit USRP1	21
2.2 Software GNU Radio	22
2.3 Použití v OS Linux	23
2.3.1 Instalace GNU Radio z binárních balíčků	23
2.3.2 Instalace ze zdrojových souborů	24
2.3.3 Spuštění, tvorba aplikace a kompilace	24
2.3.4 Dodatečná podpora USRP1	25
<b>3 Tvorba aplikací v GRC</b>	<b>26</b>
3.1 Základní bloky v GNU Radio Companion	27
3.1.1 Bloky UHD™ zařízení	27
3.1.2 Bloky bytových operátorů	28
3.1.3 Bloky grafického výstupu	28
3.2 Modulátory signálu	29
3.2.1 GMSK modulátor	29
3.2.2 Konstelační modulátory PSK a QAM	30
3.2.3 OFDM modulátor	31
3.2.4 Vysílání pomocí USRP	33
3.3 Návrh přijímačů a demodulace signálu	33
3.3.1 Demodulace frekvenčně klíčovaného signálu (GMSK)	34

3.3.2	Demodulace symbolů v konstelaci ( $M$ -PSK) . . . . .	35
3.3.3	Demodulace symbolů v konstelaci ( $M$ -QAM) . . . . .	36
3.3.4	Příjem a demodulace OFDM signálu . . . . .	36
3.4	Zabezpečení pomocí FEC API . . . . .	38
3.4.1	Implementace konvolučního kodéru . . . . .	38
<b>4</b>	<b>Modulární přenosový systém</b>	<b>39</b>
4.1	Zpracování dat ve vysílači . . . . .	39
4.2	Implementace přijímače . . . . .	40
4.3	Programování modularity . . . . .	41
4.3.1	Řízení OFDM přenosů . . . . .	43
4.4	Dosažené výsledky . . . . .	44
4.4.1	Přenos s užitím frekvenčních modulací . . . . .	45
4.4.2	Přenos s užitím fázových modulací . . . . .	45
4.4.3	Přenos s užitím QAM modulací . . . . .	47
4.4.4	Přenos s užitím OFDM . . . . .	47
4.4.5	Poznátky z používání SW GNU Radio/GRC . . . . .	48
	<b>Závěr</b>	<b>50</b>
	<b>Literatura</b>	<b>51</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>52</b>
	<b>Seznam příloh</b>	<b>54</b>
	<b>A Obsluha skriptů</b>	<b>55</b>
	<b>B Obsah příloženého DVD</b>	<b>58</b>



# SEZNAM OBRÁZKŮ

1.1	Vliv parametru BT na spektrální hustotu výkonu . . . . .	14
1.2	Stavové diagramy QAM při stejné kvalitě přenosového prostředí . . .	15
1.3	Princip ortogonality OFDM nosných . . . . .	16
1.4	Příklad alokace pilotních nosných . . . . .	18
1.5	Blokové schéma smyčky fázového závěsu dle [9], [10] . . . . .	19
3.1	Přenosový systém pomocí SDR . . . . .	26
3.2	Zapojení GMSK modulátoru v prostředí GNU Radio Companion . .	29
3.3	Zapojení QPSK modulátoru v prostředí GNU Radio Companion . . .	31
3.4	Zapojení OFDM modulátoru v prostředí GNU Radio Companion . .	32
3.5	Patříčně „zeslabený“ signál vstupující do USRP bloku . . . . .	33
3.6	Vnitřní struktura GMSK demodulátoru . . . . .	34
3.7	Zpracování signálu před vstupem do PSK/QAM demodulátoru . . . .	35
3.8	Princip činnosti přijímače OFDM signálu a dekodéru dat . . . . .	37
4.1	Struktura generovaného paketu . . . . .	40
4.2	Měřicí pracoviště pro testování systému . . . . .	45
4.3	Zřetelné konstelace přijímaného PSK signálu před demodulací . . . .	46
4.4	Srovnání konstelace při přenosu pomocí USRP a při simulaci . . . . .	47
4.5	Spektrum OFDM signálu při vzorkování rychlostí 4 MS/s . . . . .	48

## SEZNAM TABULEK

A.1	Parametry skriptů pro obsluhu vysílače a přijímače s jednou nosnou .	55
A.2	Přidané parametry skript pro obsluhu přijímače s jednou nosnou . . .	55
A.3	Parametry skriptu pro obsluhu OFDM vysílače a přijímače . . . . .	56

# ÚVOD

V dnešní době je digitální komunikace neodmyslitelnou součástí moderního života. Ať již jde o komunikaci z účelů pracovních či soukromých, schopnost zůstat ve spojení s rodinou, přáteli, ale i spolupracovníky je často neodmyslitelným standardem. Zcela nový rozměr pak přináší možnosti bezdrátových komunikačních systémů, které svojí podstatou zajišťují daleko větší pohodlí, širší oblasti nasazení a hlavně alternativu k systémům, kde spojení mechanickými vodiči není možné – ať už jde o příjem satelitního televizního vysílání, mobilní sítě pro telefonní a datová připojení a další.

Svoji podstatou se však od „drátových komunikací“ značně liší a vedle svých nepopíratelných výhod s sebou nesou i mnoho komplikací, ať už jde o složitější návrh takového systému, zpravidla menší spolehlivost bezdrátových přenosů, tak i větších nároků na bezpečnost komunikací, jejichž nosných médiiem je volný prostor.

Ve své bakalářské práci se zabývám návrhem bezdrátového přenosového systému založeném na softwarově definovaném rádiu (SDR). Tato technologie je perspektivním řešením v oblastech, kde je kladen nárok na univerzálnost řešení a implementaci nezávislou na hardwaru, na kterém má být řešení postaveno. Na rozdíl od proprietárních a většinou drahých jednoúčelových řešení nabízí SDR zcela opačný přístup a tím i daleko širší možnosti uplatnění. To je však vykoupeno vyšší náročností přípravy takového řešení, ať už jde o znalosti návrháře takového systému, ale i zařízení, na kterém pak probíhá digitální zpracování signálů.

K programování softwarového rádia ve své práci užívám vývojový framework GNU Radio, zejména jeho grafické prostředí GNU Radio Companion. Tím se možnosti práce s SDR nabízí i vývojářům, kteří se nechtějí zabývat psaním vlastních programovacích kódů. V tomto případě je v práci vytvořeno několik dílčích bloků, které tvoří jednotlivé modulátory a kodér konvolučního kódu. Vytvořené bloky jsou pak použity k tvorbě vysílače i přijímače pro použití s vývojovými kity USRP1 firmy Ettus Research. Vytvořený model je doplněn o možnosti změny základních parametrů přenosového systému a nabídne jednoduchou změnu kódování, modulací a přenosové rychlosti.

Poslední část práce je pak věnována zhodnocení vytvořeného systému a jeho schopností, kvalitu přenosů a poznatkům z provozu navrženého řešení. Dále jsou diskutovány možnosti optimalizace systému a navrženo řešení vzniklých problémů, případně možnosti potlačení jejich nežádoucích vlivů, které se při práci s vývojovými kity v prostředí GNU Radia nepodařilo zcela odstranit.

# 1 BEZDRÁTOVÁ KOMUNIKACE

Bezdrátovou komunikací, resp. bezdrátovým přenosem označujeme způsob přenosu informace či dat jiným způsobem než fyzickým propojením (elektrickým vodičem). Podle typu nosného média komunikaci dělíme na optickou, rádiovou a sonickou. Přenosovým prostředím může být vzduch nebo např. vakuum a vzdálenost mezi komunikujícími subjekty může být od jednotek metrů (komunikace počítačových periférií) až po desítky tisíc kilometrů (satelitní komunikační systémy).

Pro potřeby realizace bezdrátového přenosového systému na platformě USRP se omezíme pouze na rádiový přenos, který jediný bude použit v praktické realizaci. Následující podkapitoly jsou věnovány popisu parametrů, které definují nebo ovlivňují rádiový přenos nebo některé jeho výsledné vlastnosti a jsou měněny při sestavování komunikačního systému v této práci. Přestože je obsáhlejší popis částí a jejich vlastností tvořící bezdrátový komunikační systém mimo rozsah a zaměření tohoto textu, v práci uvádím alespoň některé základní vlastnosti modulací, kterých poté bude užito v praktické realizaci. Zbylé podkapitoly uvádí do problematiky kódování a synchronizace.

## 1.1 Digitální modulace (klíčování)

Pro rádiový přenos je nezbytné namodulovat signály na vysokofrekvenční vlny. Pokud je modulačním (řídícím) signálem signál číslicový, jedná se o digitální modulaci, někdy též označovanou jako klíčování (anglicky keying). V případě digitálních modulací zůstává nosný signál spojitý (analogový), např. harmonická funkce sinus, modulační signál je však číslicový, např. některý linkový kód. Amplitudová a fázová charakteristika modulovaného signálu tak nabývá jen omezeného počtu hodnot. Stejně jako v případě modulací analogových je možno ovlivňovat amplitudu, frekvenci a fázi nosné vlny – podle ovlivňovaného parametru lze rozlišit tři základní druhy digitálních modulací:

1. amplitudové klíčování (ASK)
2. frekvenční klíčování (FSK)
3. fázové klíčování (PSK).

Kromě základních typů modulací existují i tzv. složené digitální modulace, které kombinují dvě základní metody digitálních modulací, např. kvadraturní amplitudová modulace kombinuje amplitudové a fázové klíčování. V praxi se pak zpravidla setkáváme s tzv. vícestavovými digitálními modulacemi, kde jeden stav modulovaného signálu přenáší více bitů současně. Počet stavů modulace se označuje číslem před jejich názvem –  $M$ -stavové fázové klíčování lze tedy zapsat jako  $M$ -PSK, např.

osmistavová PSK modulace se zapíše jako 8-PSK, analogicky pak šestnáctistavová QAM jako 16-QAM. Počet přenášených bitů  $b$  v každém stavu je dán vztahem

$$b = \log_2(M), \quad (1.1)$$

kde  $M$  je počet stavů vícestavové digitální modulace, např. 16-QAM přenáší každým stavem  $\log_2(16) = 4$  bity. Mezi základní parametry digitálních modulací patří přenosová rychlost (označovaná jako bitová rychlost)  $f_b$  s dobou trvání jednoho bitu  $T_b$ , doba trvání jednoho symbolu (stavu nosné vlny)  $T_s$  a symbolová rychlost (také označovaná jako modulační rychlost)  $M$ -stavové modulace daná vztahem

$$f_s = \frac{f_b}{\log_2(M)}. \quad (1.2)$$

Za nejpodstatnější kvalitativní parametry modulačních technik uvažujme bitovou chybovost BER a symbolovou chybovost SER, které se snažíme minimalizovat zlepšením poměru C/N, tedy odstupem výkonu nosného signálu od výkonu šumu.

### 1.1.1 Gausovské klíčování s minimálním zdvihem (GMSK)

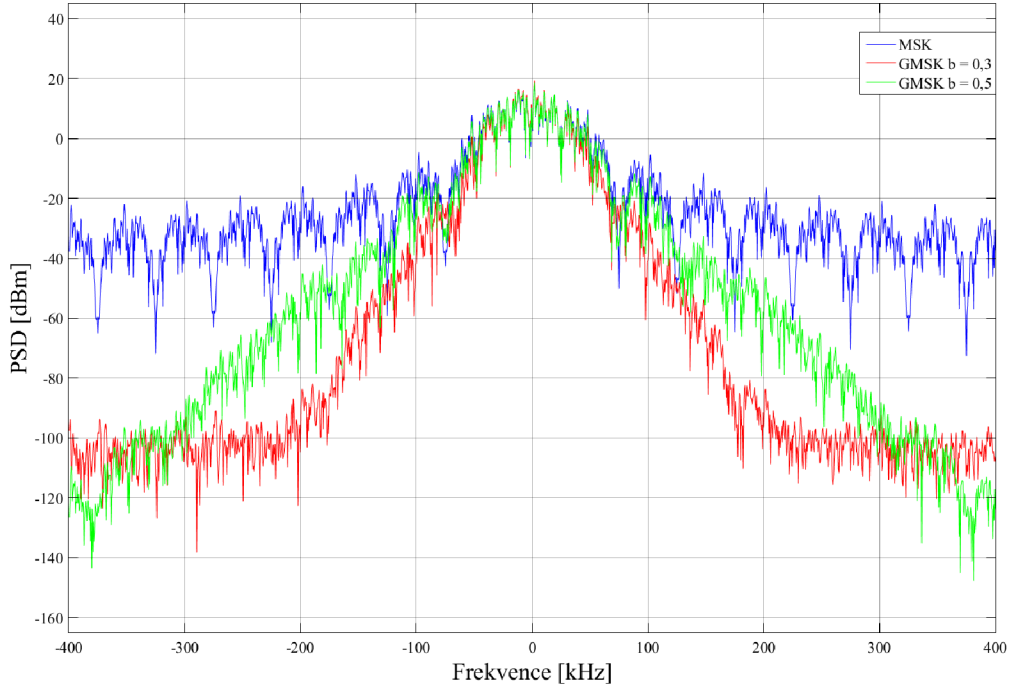
GMSK je rozšířená modulační (klíčovací) technika používaná např. v mobilních systémech GSM a CDPD. Od standardního frekvenčního klíčování s minimálním zdvihem (MSK) se liší aplikací Gaussova filtru (tvarováním pulsu) na datový tok před vstupem do frekvenčního modulátoru. Tím dojde k snížení výkonu v postranním pásmu, čímž se redukuje rušení mezi nosnými v přilehlých frekvenčních kanálech. GMSK vyniká svojí vysokou spektrální účinností a realizace modulátoru je možná pomocí sériově řazeného FSK modulátoru s oscilátorem VCO a s předmodulační propustí GLPF (Gaussian Low-Pass Filter).

Základním parametrem modulace GMSK je relativní šířka pásma  $b = BT_b$ , kde  $B$  je šířka pásma Gaussovy dolní propusti pro pokles o 3 dB a  $T_b$  je bitová perioda modulačního signálu. Vliv šířky Gaussovy dolní propusti na využití spektrum signálu je znázorněno na obr. 1.1. Při zmenšování parametru  $BT_b$  se postranní laloky výrazně zmenšují, avšak dochází k změnám tvaru modulačních impulzů, čímž mohou vznikat mezisymbolové interference.

Volba relativní šířky pásma je kompromisem mezi dostatečně potlačenými postranními laloky spektra a povolenou mezisymbolovou interferencí ISI, např. pro systém GSM byla zvolena hodnota  $BT_b = 0,3$  a pro systém DECT hodnota  $BT_b = 0,5$ . Pro  $BT_b \rightarrow \infty$  zaniká vliv filtrace a jedná se tedy o nefiltrovanou MSK modulaci.

### 1.1.2 Klíčování fázovým posuvem (PSK)

Fázové klíčování, někdy také klíčování fázovým posuvem (anglicky Phase-shift keying) je metodou digitální modulace, která pro přenos informace používá změny fáze



Obr. 1.1: Vliv parametru BT na spektrální hustotu výkonu

nosné vlny. U základní dvoustavové modulace BPSK nabývá fáze dvou diskretních stavů, tedy např.  $0^\circ$  a  $180^\circ$ . Příslušné signálové prvky modulované vlny lze pak dle [11] vyjádřit vztahy:

$$s_{\text{BPSK}}(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t), \text{ pro } 0 \leq t \leq T_b \text{ binární 1} \quad (1.3)$$

$$s_{\text{BPSK}}(t) = -\sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t), \text{ pro } 0 \leq t \leq T_b \text{ binární 0}, \quad (1.4)$$

kde  $\sqrt{2E_b/T_b}$  je amplituda modulovaného signálu a  $E_b$  je energie modulovaného signálu na 1 bit a  $T_b$  je bitová perioda. U více-stavových  $M$ -PSK modulací má nosná vlna konstantní amplitudu, ale může nabývat jeden z  $M$  možných stavů fáze. Každý stav fáze nosné vlny poté přenáší informaci o  $n = \log_2 M$  bitech a lze jej v časové oblasti popsat vztahem [11]

$$s_{\text{MPSK}}(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left[2\pi f_c t + (i-1)\frac{2\pi}{M}\right], \text{ pro } 0 \leq t \leq T_s, i = 1, 2, \dots, M \quad (1.5)$$

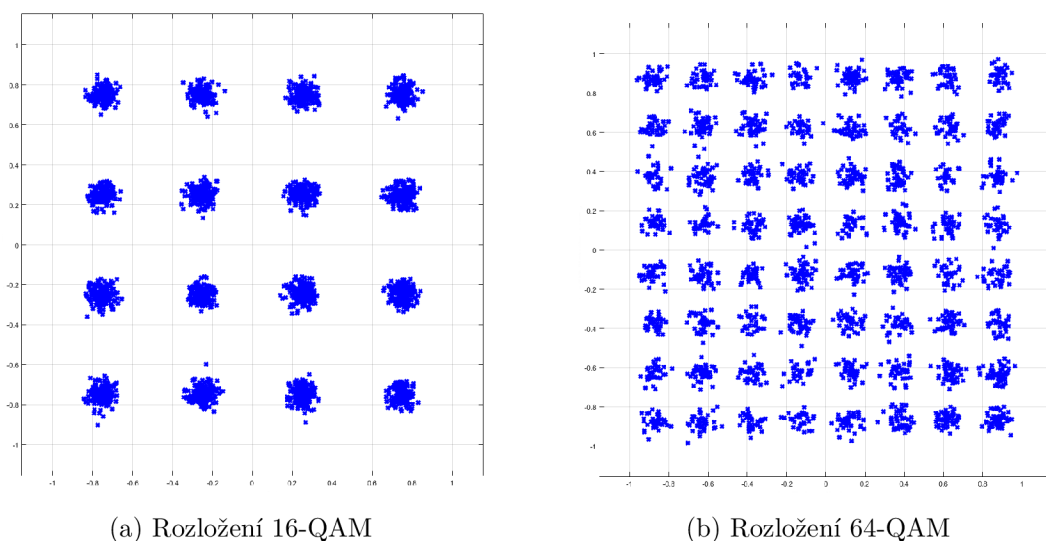
kde  $E_s$  je energie na jeden symbol a  $T_s$  je symbolová perioda. Mezi běžně používané formáty fázového klíčování patří binární fázové klíčování, kvadrurní fázové klíčování (QPSK) s mapováním 4 symbolů do konstelačního diagramu, kde jeden symbol (stav nosné) přenáší právě 2 bity. S rostoucím počtem stavů se zvyšují nároky na přijímač a je nutné zajistit dostatečný SNR. V práci jsou použity techniky fázového

klíčování realizovány s užitím diferenčního kódování, bitová slova jsou pak mapována změnám o patřičnou hodnotu fáze.

### 1.1.3 Kvadrurní amplitudová modulace (QAM)

Kvadrurní amplitudová modulace je pásmově úsporná a efektivní modulační technika. Aby nedošlo ke kompresi signálu a tím i jeho degradaci, k provozu této modulační techniky je nutná operace v lineární oblasti výkonového zesilovače. QAM obvykle využívá dvojici sinusových signálů (kvadrurní nosné nebo kvadrurní komponenty) s konstantním kmitočtem, které jsou vzájemně fázově posunuté o  $90^\circ$ . Tyto nosné jsou nezávisle na sobě amplitudově modulovány a poté sečteny. Výsledný signál má podobu signálu modulovaného (klíčovaného) jak fázovým posuvem, tak amplitudovým posuvem. Tím je dosaženo lepšího využití stavového diagramu, který může nabývat 16, 64, ale i 256 v závislosti na kvalitě přenosového prostředí.

QAM lze podobně jako PSK modulovaný signál znázornit pomocí konstelačních diagramů. Nejpoužívanější typ diagramu má konstelaci (rozložení symbolů) čtvercovou a pro  $M = 16$  a  $M = 64$  se simulovaným AWGN je zobrazen na obr. 1.2.



Obr. 1.2: Stavové diagramy QAM při stejné kvalitě přenosového prostředí

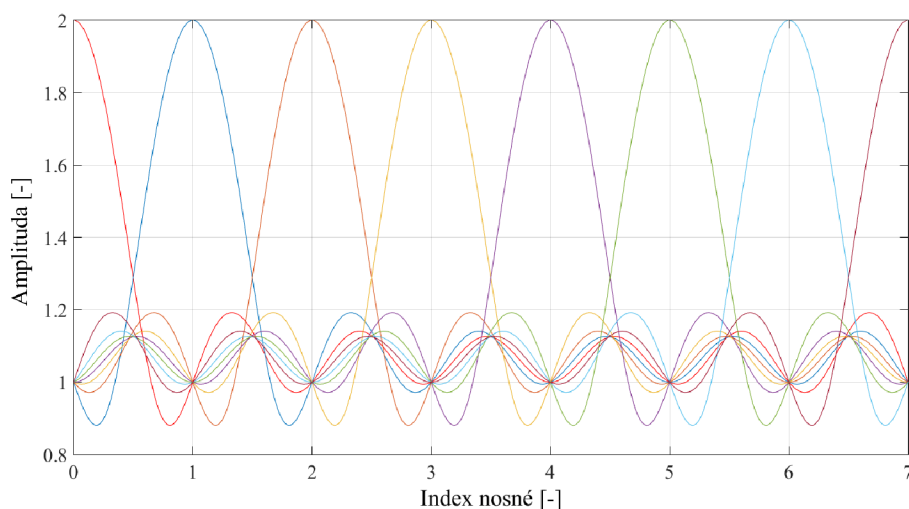
Z obr. 1.2 je tedy patrné, že s rostoucím počtem stavů, které přijímač musí rozlišit, rostou i nároky na jeho přesnost (konstelační diagram „houstne“ směrem k počátku – amplituda se snižuje).  $M$ -QAM signál lze zapsat dle [5] ve tvaru:

$$s_i(t) = \sqrt{\frac{2E_0}{T}} a_i \cos(2\pi f_c t) + \sqrt{\frac{2E_0}{T}} b_i \sin(2\pi f_c t), 0 \leq t \leq T, \quad (1.6)$$

kde  $f_c$  je kmitočet nosné vlny,  $E_0$  je energie signálu s nejmenší amplitudou,  $a_i, b_i$  jsou celá čísla určující polohu bodů v konstelačním diagramu.

### 1.1.4 Ortogonální multiplex s frekvenčním dělením

OFDM je modulační technikou, která při vysokofrekvenčním přenosu využívá velké množství (stovky až tisíce) nosných vln. Tato technika našla široké uplatnění při přenosu digitálního audio/video signálu, přenosů dat na krátké vzdálenosti (sítě Wi-Fi), ale i vysokorychlostních WAN sítích jako WiMAX nebo LTE. OFDM zajišťuje vysokou odolnost proti mezisymbolovým přeslechům při vícecestném šíření. Toho je docíleno použitím delší doby trvání symbolů nebo vložení mezery mezi symboly (tzv. ochranné intervaly). Modulační technika OFDM prodlužuje dobu trvání symbolu rozdělením vstupního bitového toku na několik paralelních toků, z nichž každý nezávisle moduluje jednu subnosnou vlnu pomocí vektorové modulace (např. QPSK,  $M$ -QAM). Sousední subnosné vlny udržují mezi sebou konstantní kmitočtový odstup  $\Delta f$ . I přes relativně malý odstup (řádově kHz) je problém interference jednotlivých signálů odstraněn díky ortogonalitě nosných vln, jejichž průběh je znázorněn na obr. 1.3<sup>1</sup>.



Obr. 1.3: Princip ortogonality OFDM nosných

Praktickou činnost OFDM modulátoru lze shrnout do tří kroků. Modulátor nejprve rozdělí vstupní posloupnosti symbolů na bloky po  $M$  symbolech, které jsou mapovány např. pomocí QPSK nebo  $M$ -QAM modulací, výpočtem IFFT převede každý blok z frekvenční do časové oblasti a výstup IFFT převádí na sériovou posloupnost prvků, kterou následně posílá na výstup. Princip demodulátoru lze do jisté míry označit za opačný – na jeho vstupu je signál rozdělen do paralelních větví, pomocí algoritmu FFT převeden do frekvenční oblasti, následuje přiřazení bitových

<sup>1</sup>Vykreslovací funkce převzata z WWW: <http://www.mathworks.com/matlabcentral/fileexchange/29816-ofdm-subcarriers>



kombinací jednotlivým symbolům v detektoru a nakonec opět proběhne převedení paralelních toků na sériový tok dat.

## 1.2 Synchronizace

V digitálních komunikačních systémech je správná synchronizace vysílače a přijímače nezbytným předpokladem pro korektní přenos. Synchronizací se většinou řeší problematika obnovy kmitočtu a fáze nosné a časování symbolů. K změnám fáze a kmitočtu nosné může docházet vlivem nestability frekvence oscilátoru a vlivem Dopplerova jevu při vzájemném pohybu vysílače a přijímače. Fázová odchylka mezi vysílanou a přijímanou vlnou je dána i konečnou dobou šíření elektromagnetické vlny a charakteristikou přenosového kanálu. Pro synchronní demodulaci musí být v přijímači použit obvod pro obnovu nosné vlny CR (Carrier Recovery), označované také jako obvody synchronizace nosné [3]. Pro přijímaný signál poté platí dle [9]:

$$r(t) = A\nu(t - \tau) \cos [2\pi(f_c - f_d)(t - \tau) + \theta] + n(t), \quad (1.7)$$

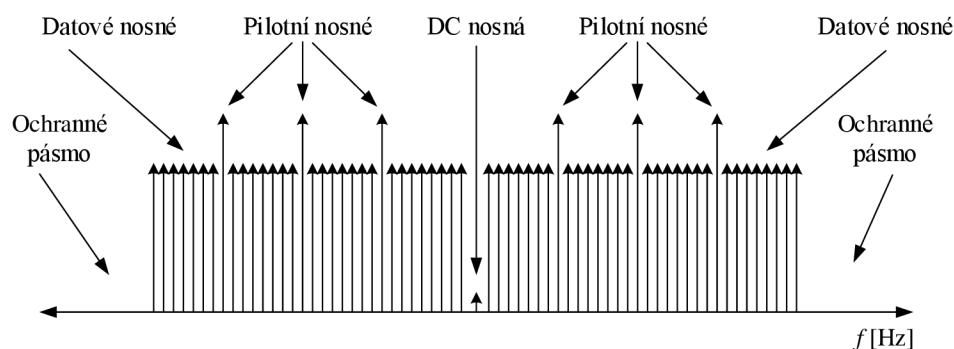
kde  $\nu(t - \tau)$  je signál v základním pásmu,  $f_d$  je kmitočtová odchylka nosné od nominálního kmitočtu  $f_c$ ,  $\tau$  je zpoždění a  $n(t)$  je šum. Fázové odchylky lze pak družít do jednoho členu [9]: Synchronizace zpravidla probíhá na třech úrovních, a to:

1. Obnova nosné vlny,
2. obnova časování symbolů,
3. synchronizace rámců.

V následujících podkapitolách jsou stručně popsány metody obnovení časování symbolů a obnovení nosné vlny.

### 1.2.1 Synchronizace v OFDM systémech

Signál dopadající na vstupu přijímače v OFDM systému je vlivem jeho šíření obvykle do jisté míry deformován. Předpokladem pro správnou demodulaci signálu je zajištění synchronizace vysílače a přijímače, který musí správně rozpoznat začátek každého OFDM symbolu. Tento proces je obvykle složen z procesu nalezení správné doby trvání symbolu a frekvenční odchylky, která se projeví vznikem interferencí mezi nosnými, čímž dojde k porušení podmínky ortogonality. Jednou z možností řešení této problematiky je doplnění přijímače o obvod pro odhad (estimátor) kmitočtového a časového offsetu, který používá pomocné referenční symboly, jež jsou proloženy mezi datovými, a pilotní tóny, které jsou modulovány mezi jednotlivými nosnými užívanými pro přenos uživatelských dat. [11] Způsob alokace pilotních



Obr. 1.4: Příklad alokace pilotních nosných

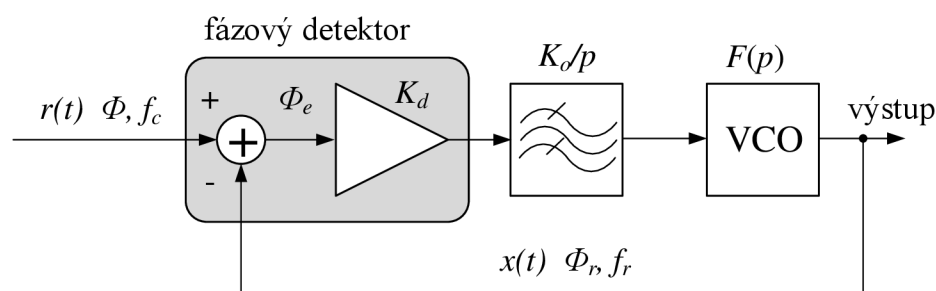
nosných je specifický pro daný komunikační systém. Příklad proložení pilotních nosných mezi datové je zobrazen na obr. 1.4. Pokud je použita koherentní demodulace subnosných, je nutné korigovat kmitočtové zkreslení kanálu a ekvalizace kanálu je nezbytná. Za tímto účelem se v estimátoru kanálu pro každou subnosnou provádí odhad přenosové charakteristiky kanálu (užitím kombinace známých pilotních tónů a referenčních symbolů). Odhad je poté užit v ekvalizéru, v němž se každá subnosná násobí individuálním korekčním faktorem.

### 1.2.2 Metody obnovení nosné vlny

Jednou z možných metod obnovování nosných vln je použití jednoho nebo více nemodulovaných pilotních signálů, jež mají specifický vztah k nosné. Pilotní signály pak mohou být vysílány souvisle např. metodou označovanou *Tone in Band (TIB)* nebo v systémech časového multiplexu např. metodou *Pilot Symbol-Assisted Modulation (PSAM)*, kde dochází k prokládání pilotních symbolů definovaným způsobem s informačními datovými symboly.

Druhou z metod používaných pro obnovení nosné vlny, která je vhodná pro systémy s nepotlačenou nosnou vlnou a s modulací s oběma postranními pásmy a potlačenou nosnou vlnou, tedy např. formáty *M-PSK* a *M-QAM*, je získání referenční nosné vlny přímo z přijímaného signálu. Protože je veškerý vysílaný výkon soustředěn do užitečných složek modulovaného výkonu, jejich výkonová účinnost je větší než u systémů používajících pilotní signál. Základem většiny metod synchronizace je použití různých modifikací smyčky fázového závěsu PLL (Phase-Locked Loop), resp. DPLL (Digital Phase-Locked Loop) [9].

Pro fázovou a kmitočtovou synchronizaci lze implementovat PPL obvod, jehož blokové schéma je znázorněno na obr. 1.5. Obvod je tvořen fázovým detektorem, filtrem smyčky (filtr typu dolní propust) a napětím řízeným oscilátorem (VCO).



Obr. 1.5: Blokové schéma smyčky fázového závěsu dle [9], [10]

Na vstup fázového detektoru je přiveden vstupní signál  $r(t) = A \cos[2\pi f_c t + \phi]$  a signál z VCO  $x(t) = \cos[2\pi f_r t + \phi_r(t)]$ . Rozdíl fází těchto signálů je v detektoru převeden na rozdílů odpovídající napětí. Výstup fázového detektoru je filtrován dolní propustí obvykle prvního nebo druhého řádu – tento filtr určuje rychlost ustálení smyčky. Napětím řízený oscilátor poté generuje periodický signál, jež je úměrný řídicímu napětí na jeho vstupu.

### 1.2.3 Obnova časování symbolů

Při použití synchronizačních obvodů pro obnovu časování symbolů rozlišujeme systémy dle jejich užití multiplexního kanálu pro přenos informace o hodinovém signálu. Ten je možno přenášet užitím časového nebo kmitočtového multiplexu. Užití multiplexu s sebou nese jednoduchost realizace takovýchto synchronizačních obvodů a obnovený časový signál je kvalitnější. Nevýhodou je však snížená energetická účinnost přenosového systému [9].

Systémy užívající kmitočtového multiplexu přenášejí synchronizační signál na pomocné subnosné, která je následně v přijímači obnovena filtrací. Systémy s časovým multiplexem vkládají do přenášené datové posloupnosti známé sekvence symbolů, na základě kterých je pak obnovován hodinový signál – tzv. *Data-Aided* systémy [11] [9]. Jejich použití je nezbytné v případech použití modulací se spojitou fází. Je-li naopak k obnově časování symbolů použit přímo demodulovaný signál, jedná se o tzv. *Non-Data-Aided* systémy [9].

## 1.3 Kanálová kódování

Kanálové kódování je metodou zabezpečení kanálem přenášených bitů proti chybám vzniklým vlivem nedokonalosti tohoto kanálu. Toho je docíleno zavedením jisté míry redundance (nadbytečnosti) do původního datového toku přidáním zabezpečovacích

bitů. Zabezpečovací (nadbytečné) bity neovlivní původně přenášenou informaci, dekódér na straně přijímače je však může využít k detekci a dle možností použitého kódování případně opravě chyb vzniklých při přenosu. O možnostech kódování pro bezdrátové kanály je více uvedeno v [1].

### 1.3.1 Konvoluční kódy

V práci je využito převážně konvolučních kódů, které kromě detekce chybného bitu dokáží chyby v přenosu opravit. Kodér je většinou realizován posuvným registrem dané délky a několika součtových členů realizující operaci exkluzivního součtu. Počet sčítacích členů je dán generujícími mnohočleny stupně  $S$ , které udávají pravidla tvorby výstupních bitových signálů. Dalšími určujícími vlastnostmi jsou parametry délky rámce vstupních  $m$  a výstupních  $n$  bitů. Vzájemný poměr vstupních a výstupních bitů udává kódový poměr, délka působení  $K$  kódu je dána vztahem  $K = (S + 1)m$ , kde  $S$  je stupeň generujícího mnohočlenu.

### 1.3.2 Prokládání (Interleaving)

Prokládání je principiálně jednoduchou technikou ochrany bitového toku proti shlukovým chybám, které mohou vzniknout například vlivem interferencí působících v rádiovém prostředí. Činnost prokladače spočívá v změně pořadí vysílaných bitů. Bitový tok je ve vysílači ukládán do paměti po řádcích a vyčítán z paměti po sloupcích. Opačný proces je pak aplikován na straně přijímače. Případná vzniklá chyba posloupnosti bitového toku (shluk chybných bitů v řadě) je nyní rozprostřena do delších intervalů, tudíž ji lze korigovat aplikací protichybového kódování. Prokládání obvykle probíhá na úrovni bitů a na úrovni symbolů současně, převážně pak v kombinaci s vhodným opravným kódem.

## 2 KONCEPCE SOFTWAREVÉHO RÁDIA

Softwarově definované rádio (v textu také zkráceně SDR) je flexibilní technologií v oblasti rádiových přenosů a digitálního zpracování signálů. Základní myšlenkou a hlavní výhodou je otevřenost a možnost překonfigurování takto realizovaného systému. Funkci rádiového přijímače a vysílače provádí pro tyto účely určený hardware (v závislosti na požadované kvalitě a funkci lze volit i jiná „neoficiální“ zařízení jako jsou TV tunery apod.) a jeho řízení pak zastává specializovaný software, popř. v něm vytvořená aplikace. Takto lze např. nastavovat frekvence, komunikační protokoly, aktivovat a deaktivovat modulátory a demodulátory, aplikovat filtry apod. Pomocí takto programovaných aplikací lze na platformě softwarově definovaného rádia relativně rychle nasadit širokou škálu aplikací, ať už jde např. o jednoduchý přenos souboru, hlasového signálu mezi dvěma stanicemi, ale např. i přístupová (rádiová) část mobilních sítí GSM, UMTS nebo monitor rádiového provozu v sítích Wi-Fi.

Vývojem SDR kitů se zabývají společnosti SRL, projekt WiNRADiO, Ettus Research a další. V práci je použita původní (první) verze vývojového kitu společnosti Ettus Research USRP1 (Universal Software Radio Peripheral). Následující podkapitoly jsou věnovány popisu hardwarové a softwarové části tvořící celek softwarově definovaného rádia.

### 2.1 Vývojový kit USRP1

Vývojový kit USRP1 je prvním produktem firmy Ettus Research zaměřeným na zpracování rádiových signálů. Základním vybavením USRP1 jsou 2 duální 12-ti bitové analogově-digitální převodníky schopné pracovat rychlostí 64 MS/s (milionů vzorků za sekundu) a 2 duální 14-ti bitové digitálně-analogové převodníky pracující rychlostí 128 MS/s. Dále je možno využít dvou anténních vstupně-výstupních konektorů. Pro připojení k hostitelskému počítači je použito rozhraní USB verze 2.0, díky kterému je zařízení schopno streamovat data rychlostí až 16 MS/s. Tyto vstupy a výstupy jsou připojeny k programovatelnému hradlovému poli Cyclone od firmy Altera.

USRP jednotka disponuje také možností vnitřního rozšíření o dvě tzv. daughterboards. Díky těmto přídatným kartám je možno na vývojovém kitu provozovat až dva zcela oddělené vysílací a přijímací řetězce nebo jednoduše zajistit funkčnost ve dvou frekvenčních pásmech. Rozšiřující karty lze volit dle požadavků na typ operace zařízení (přijímač, vysílač, přijímač i vysílač), rozsahu pracovních frekvencí, popř. dle požadovaných frekvenčních pásem, ve kterých má přípravek pracovat. Pro rea-

lizaci komunikačního systému je nutno zvolit rozšiřující kartu, která je schopna na dané frekvenci přijímat i vysílat. Vhodnou volbou by tak mohla být např. daughterboard s označením WBX, která pracuje v širším frekvenčním pásmu od 50 MHz až 2,2 GHz při výstupním výkonu až 100 mW.

## 2.2 Software GNU Radio

GNU Radio je volně dostupný software s otevřeným zdrojovým kódem, který slouží jako prostředí pro tvorbu skriptů a programů pro zpracování signálů. Program obsahuje softwarové řešení filtrů, kanálových kodérů, modulačních a demodulačních jednotek, synchronizační prvky, ekvalizéry a spoustu dalších prvků, které se užívají v rádiových systémech. Program lze využít přímo k programování hardwarových jednotek nebo také jako simulační prostředí při digitálním zpracování signálů. Pro přístup k hardwarovým prvkům kitu USRP1 je užíván UHD ovladač (USRP Hardware Driver™). Tento ovladač je pro všechna zařízení z produktové kategorie USRP stejný – je použit i u novějších modelů kitů USRP řad N, B, E a lze jej užít i při implementaci aplikací pomocí programů Matlab s rozšířením Simulink, LabVIEW, popřípadě při samostatném programování kitů bez aplikace zajišťující vývojové prostředí.

GNU Radio s jednotlivými prvky aplikace pracuje v blokovém uspořádání, přičemž bloky reprezentují jednotlivé prvky aplikace, které lze vzájemně propojovat a mezi propojenými bloky pak přenášet data. Z datových typů lze komunikaci mezi bloky (prvky) realizovat např. v bytech, číselných formátech (celá čísla, čísla s pohyblivou řádovou čárkou, komplexní čísla), vektory a další. Aplikací vytvořenou v prostředí softwaru GNU Radio je pak tzv. „flowgraph“ programovaný v jazyce C++ nebo Python. Graficky lze tuto aplikaci přirovnat ke grafu signálové cesty jdoucí přes prvky zpracovávající tento signál.

Při tvorbě flowgraphů lze využít nadstavby GNU Radio Companion, která GNU Radio rozšiřuje o grafické uživatelské rozhraní. To lze využít při tvorbě projektů bez znalosti programování v použitých jazycích C++ či Python – tvorba flowgraphů probíhá blokovým uspořádáním a propojováním použitých prvků. Toto uspořádání je potom sestaveno a následně přeloženo do programového kódu programu GNU Radio.

Software je dostupný pro operační systémy Windows, Mac OS X i Linux. Také v něm vytvořená aplikace (flowgraph) není závislá na hostitelském operačním systému.

## 2.3 Použití v OS Linux

Přestože je GNU Radio multiplatformní aplikací, v práci je provozována v prostředí operačního systému Linux, konkrétně v distribuci Ubuntu 14.04.1. Jedná se o distribuci speciálně upravenou pro provoz vývojového prostředí GNU Radio, tudíž je zde GNU Radio společně s grafickým rozhraním již předinstalováno a lze jej okamžitě začít používat pro tvorbu aplikací.

Tato distribuce vychází v několika postupně aktualizovaných verzích ročně, nelze tak vždy zaručit aktuálnost všech v distribuci obsažených aplikací. Tento problém se týká i integrace programu GNU Radio. Pokud je v distribuci obsažena starší verze softwaru nebo GNU Radio není dostupné vůbec, lze jej doinstalovat několika způsoby. Ty jsou dále popsány v následujících podkapitolách.

### 2.3.1 Instalace GNU Radio z binárních balíčků

Nejjednodušším a také nejspolehlivějším způsobem instalace SW GNU Radio je využití softwarového balíčkového správce. Konkrétní správce balíčků se liší v závislosti na použité distribuci OS Linux. Pro distribuce založených na jádře Debianu, tedy Ubuntu a jeho variace, využívající balíčkového manažera *Advanced Packaging Tool (APT)*, lze automatizovanou instalaci spustit příkazem<sup>1</sup>:

```
# apt-get install gnuradio
```

Na distribucích pracujících s RPM formátem binárních balíčků je nutno použít adekvátního správce (např. *Yellowdog Updater, Modified (yum)*). Instalaci lze pak podobně spustit voláním instalační funkce balíčkového správce, tedy příkazem:

```
# yum install gnuradio
```

Na jiných typech distribucí OS Linux, které pracují s jinými správci binárních balíčků lze pak spustit automatizovanou instalaci voláním příslušného správce pro instalaci balíčku **gnuradio**. Instalace z předkompilovaných binárních balíčků všechna nastavení a konfigurace provádí za uživatele během instalačního procesu, odpadá tak nutnost starat se o specifikace zdroje balíčku, cílového adresáře apod. Konfigurační soubory jsou pak dostupné v adresáři `/etc/gnuradio/` a předprogramované bloky včetně příkladů aplikací pak v `/usr/share/local/gnuradio/`.

---

<sup>1</sup>při použití standardního uživatelského účtu je potřeba dočasně zvýšených privilegií, před samotným příkazem tedy vložíme `sudo`

## 2.3.2 Instalace ze zdrojových souborů

Druhou a zdatelně náročnější možností instalace GNU Radio je instalace ze zdrojových kódů. Ty je možné stáhnout na oficiálních stránkách projektu GNU Radio dostupných na [www.gnuradio.org](http://www.gnuradio.org). Pro použití zařízení z rodiny USRP, je nutno před instalací SW GNU Radio prvně nainstalovat ovladač UHD. Samotný proces kompilace, sestavení a instalace GNU Radia je možné provést dle návodu<sup>2</sup>. Podrobnější rozbor instalace je nad rámec práce.

## 2.3.3 Spuštění, tvorba aplikace a kompilace

Po úspěšné instalaci popř. spuštění distribuce s předinstalovaným prostředím GNU Radio ještě před samotnou tvorbou aplikací ověříme použitou verzi. To lze provést v terminálu OS Linux příkazem:

```
~$ gnuradio-companion --version
```

Pro účely této práce je použitou verzí k dnešnímu datu aktuální verze 3.7.7.1. Podobně z terminálu spustíme i GNU Radio Companion, tedy příkazem:

```
~$ gnuradio-companion
```

Tvorba flowgraphů probíhá vkládáním bloků z pravého prvkového menu a spojováním jejich výstupů se vstupy následujících bloků. Nastavení parametrů jednotlivých bloků je možné zobrazit dvojitým kliknutím na blok, kde je mj. nutné nastavit odpovídající datové typy vstupů/výstupů tak, aby vzájemně odpovídaly datovému typu vstupu/výstupu bloku s ním propojovaného. Po dokončení flowgraphu je možné jej vygenerovat stisknutím F5 a následně spustit stisknutím F6. Ukončení běhu flowgraphu lze provést klávesou F7.

Při běhu flowgraphu jsou do konzolového okna, resp. terminálového okna signalizovány chybové stavy, ke kterým může docházet během zpracování signálu. Těmito signály jsou identifikovány problematické stavy, nikoli jejich konkrétní počet – při déle trvajících chybě se konzolové okno velmi rychle zaplní chybovými symboly, počet těchto symbolů však neodpovídá přesnému počtu chyb. Tyto symboly jsou:

- *u* – problém působí USRP jednotka
- *a* – problém audio zařízení (zvuková karta)
- *O* – „overrun“; host (PC) nestíhá zpracovat přijímaná data, typicky nedostatečný výkon CPU
- *U* – „underrun“; host (PC) nestíhá poskytovat, resp. generovat data pro USRP

Typicky se pak mohou vyskytovat kombinace těchto symbolů, tedy např. nedostatek audio vzorků pro zvukovou kartu je signalizováno kombinací „aUaU“ apod.

---

<sup>2</sup>dostupné z WWW: <http://gnuradio.org/redmine/projects/gnuradio/wiki/BuildGuide>



### 2.3.4 Dodatečná podpora USRP1

Před samotným použitím USRP1 bude pravděpodobně nutné do aktuálních verzí OS Linux (testované distribuce Kali 1.0.9a a Ubuntu 14.0.1) nutné přidat podporu pro původní USRP1 kit stažením obrazu jeho firmware pomocí automatizovaného skriptu. To lze provést zadáním následujících příkazů:

```
# cd /usr/share/uhd/images/3
# ./uhd_images_downloader.py
```

Po stažení jsou binární obrazy instalovány do `/usr/share/uhd/images/` a je možno je použít k programování kitů USRP1 pomocí GRC. Připojené USRP jednotky mohou být adresovány pomocí IP adres (USRP řady N) nebo svých sériových čísel. Ta lze zjistit odesláním požadavku na UHD ovladač příkazem:

```
$ uhd_find_devices
```

Jeho výstupem je pak výpis všech připojených USRP jednotek včetně sériových čísel, např.:

```
-----
-- UHD Device 0
-----
Device Address:
    type: usrp1
    name:
    serial: E4R22Y2U
-----
-- UHD Device 1
-----
Device Address:
    type: usrp1
    name:
    serial: E4R22Y4U
```

Adresou této konkrétní USRP jednotky (pokud nelze použít IP) v GRC je pak řetězec `serial=E4R22Y2U`.

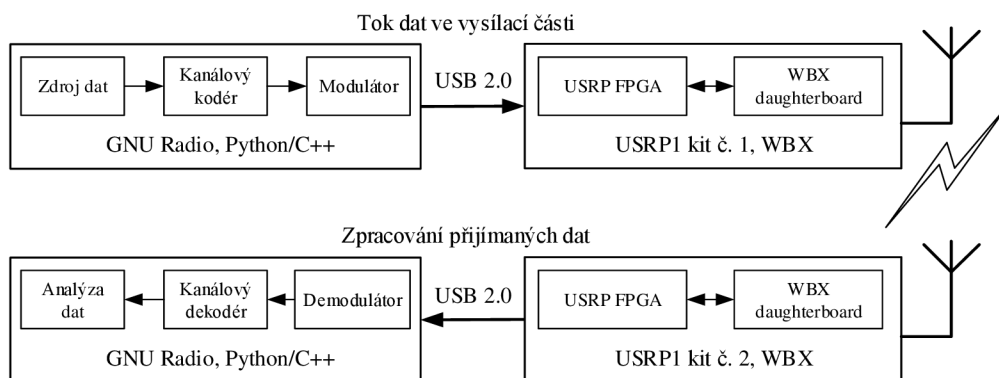
---

<sup>3</sup>Uvedená cesta je použita při automatizované instalaci a na některých distribucích se může lišit

### 3 TVORBA APLIKACÍ V GRC

Možností programování softwarových rádií je více, konkrétně v případě USRP jednotek se nabízí programování samotného FPGA například použitím vývojového prostředí Xilinx ISE. V bakalářské práci však tohoto způsobu není využito a veškerá tvorba přenosového systému je realizována programováním softwarových skriptů v jazyce Python (automatizovaný proces generování kódu v GRC). Principem činnosti takové aplikace je tedy vhodně nastavený zdroj a cíl signálu, který na své cestě vstupuje do několika bloků realizující digitální zpracování. Bloky, které se na procesu DSP podílí, jsou pak tvořeny funkcemi jazyka C++, ke kterým přistupuje interpret jazyka Python. FPGA USRP nikdy není přepsáno, lze tedy říci, že USRP pomocí UHD poskytuje aplikaci DAC převodník číslicového signálu na analogový v případě vysílače a převod analogového signálu na číslicový v případě přijímače, jehož vzorky jsou pak vstupními daty flowgraphu v GRC.

Samotný běh dat ve flowgraphu aplikace může být několika formátů. Kromě základních datových typů, které GRC rozlišuje, lze na formát dat pohlížet jako na neformátovaný tok, který vstupuje do jednotlivých bloků a po zpracování putuje na jejich výstup, nebo také na data předem definované délky, které blokům vymezují rozsah jejich činnosti – typicky tedy data paketového formátu. Takový blok poté uvažuje délku  $N$  vstupních dat a svůj operační cyklus provádí pouze na  $N$  bytech, číslech apod. Před samotnou tvorbou přenosového systému je tedy třeba zvážit typ použití takového systému, jehož tvorba může být v některých případech jednodušší většinou při modulaci dat, synchronizaci, zabezpečení apod.



Obr. 3.1: Přenosový systém pomocí SDR

Obecný přenosový systém lze nezávisle na způsobu jeho realizace v GRC vždy shrnout do výše zobrazených bloků, které tvoří vysílač a přijímač. Obrázek naznačuje schéma jednosměrného spoje mezi dvěma body: Zdrojová data jsou v binární

podobě v bytovém formátu fixní délky<sup>1</sup> a dále vstupují do FEC kodéru, kde je bitová posloupnost zabezpečena přidáním redundantní informace fixní délky. Takto formátovaný rámeček poté vstupuje do modulátoru, jehož výstupem je v čase vzorkovaný signál v podobě komplexních čísel, která jsou předávána USRP jednotkám. V přijímači je tok dat inverzní (předpoklady pro správný příjem USRP jednotkou jsou rozebrány v kap. 3.3) – po demodulaci a odstranění informace kanálového kodéru mohou být data jistým způsobem zpracována (uložení do souboru, zobrazení bitové posloupnosti apod.). Protože samotné USRP v konfiguraci s jednou WBX deskou je vybaveno jedním vysílacím rozhraním a jedním přijímacím rozhraním (TX/RX a RX2), která mohou pracovat v plně duplexním režimu na jiných frekvencích, pro realizaci obousměrného komunikačního systému se nabízí užití FDD, tedy duplexního přenosu s frekvenčním dělením a aplikaci doplnit o větev DSP bloků realizující i přijímač, resp. vysílač.

## 3.1 Základní bloky v GNU Radio Companion

Jak bylo popsáno v úvodu kapitoly, zpracování signálů probíhá pouze na jeho vzorkovaném průběhu. Činnost USRP kitů lze tedy shrnout jako cílový či zdrojový blok, jež se stará pouze o ADC a DAC a informace s hostitelskou PC přenáší pouze ve formě vzorků. Na signálu v digitální podobě je prováděno DSP užitím různých typů bloků. Následující podkapitola je věnována základnímu popisu jednotlivých bloků, které v práci používám pro realizaci přenosových větví v obou směrech.

### 3.1.1 Bloky UHD<sup>TM</sup> zařízení

Po správné instalaci UHD<sup>TM</sup> ovladače se v prostředí GRC zpřístupní bloky UHD: USRP Source<sup>2</sup> a UHD: USRP Sink<sup>3</sup>. Nejdůležitějším parametrem ovlivňujícím provoz USRP jednotky je **Sample Rate**, neboli vzorkovací kmitočet vzorkování. Jeho hodnota přímo odpovídá použité šířce pásma jak vysílače, tak přijímače. Další nastavitelné parametry UHD<sup>TM</sup> bloků se odvíjí od použité daughterboard (WBX). Jsou jimi hlavně RF frekvence a zesílení. Parametr **Center Freq [Hz]** lze dle specifikace k WBX možno volit z rozsahu 68,75 MHz až 2,2 GHz, parametr zesílení **Gain [dB]** pak lze pro vysílací řetězec (TX gain) použít z rozsahu 0 – 25 dB (krok 0,05 dB) a

---

<sup>1</sup>Bytový formát je způsobem komunikace mezi bloky v GRC, avšak podoba bytu může být předefinována (počet, „směr LSB/MSB“) apod.

<sup>2</sup>Blok typu *Source* je vždy zdrojem dat s nulovým počtem vstupů

<sup>3</sup>Blok typu *Sink* je vždy koncovým bodem toku dat s nulovým počtem výstupů

přijímací řetězec (RX gain) z rozsahu 0 až 31,5 dB (krok 0,5 dB). Posledním parametrem je textový řetězec **Antenna**, který nastavuje použitý rádiový vstup/výstup na WBX desce.

### 3.1.2 Bloky bytových operátorů

Pro realizaci datové komunikace v GRC je třeba správně rozlišovat bytové formáty, ve kterých si bloky předávají data. Základním předpokladem je správné užití bytových operátorů pro případnou konverzi dat na vstupu a výstupu bloků. Data mohou být v tzv. „Packed“, nebo „Unpacked“ formě. Uvažujme např. požadavek na zápis binární posloupnosti

```
[1, 0, 1, 1, 0, 1]
```

do souboru. Obsahem souboru bude binárně (unpacked) reprezentovaný obsah ve tvaru:

```
[00000001 00000000 00000001 00000001 00000000 00000001].
```

Je tedy patrné, že zápis stejné informace v odlišných formátech může značně změnit výslednou velikost binárně reprezentovaného obsahu. V případě, že z každé osmice, která obsahuje pouze jeden „užitečný“ bit (typicky výstup dvoustavového modulátoru), chceme pracovat pouze s jedním bitem lze využít převodníky mezi těmito tvary, pro výše popsaný příklad blok *Unpacked to Packed* či *Pack K Bits*. S patřičným nastavením těchto bloků lze tedy z každého bytu vyčíst pouze  $K$  bitů, které poté tvoří nový výstupní byte obsahující  $K$  užitečných bitů.

### 3.1.3 Bloky grafického výstupu

Mimo mnohé bloky realizující DSP je GNU Radio Companion v základu vybaven mnoha bloky realizující grafický výstup. Lze tak v jakémkoli bodě flowgraphu (někdy s patřičnou konverzí na komplexní, popř. číslo s pohyblivou řádovou čárkou) graficky zobrazit tok dat a signálu v časové či jeho frekvenční doméně.

Kromě grafických výstupů lze pomocí grafických rozhraní<sup>4</sup> *QT* a *WX* snadno realizovat vstupy a předávat tak aplikaci příkazy v reálném čase. To je výhodné zejména v případě potřeby průběžné změny zesílení signálu, přeladění frekvence USRP apod.

Pro realizaci systému je v práci hojně využito bloků *QT GUI Frequency Sink* pro zobrazení frekvenční reprezentace signálu a bloku *QT GUI Constellation Sink* pro snadné vykreslení konstelačních bodů v IQ rovině.

<sup>4</sup>v jednom flowgraphu nelze současně generovat prvky *QT* a *WX* rozhraní

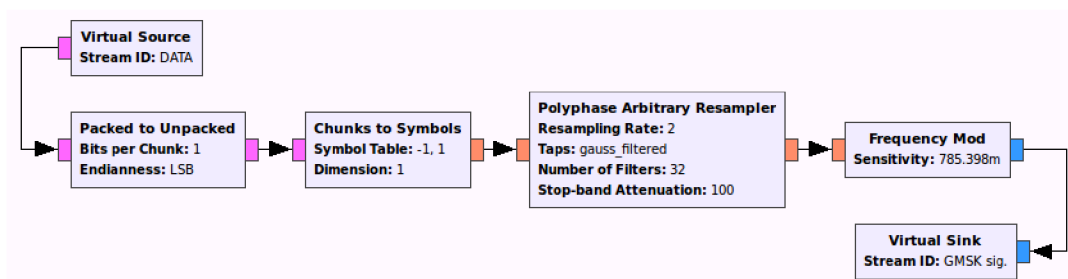
## 3.2 Modulátory signálu

Modulátory a demodulátory jsou základními stavebními prvky přenosového systému. Jejich korektní implementace značně ovlivní výsledné možnosti (přenosová rychlost, chybovost, požadovaná šířka pásma) celého systému. GNU Radio (GRC) nabízí již několik hotových bloků, které realizují funkci různých (de)modulátorů, ať již jde o jednodušší analogové FM modulace, digitální modulace s jednou nosnou, ale i OFDM. Protože je však práce koncipována jako naučný průvodce vývojovým prostředím, modulátory jsou tvořeny blokově ze základních stavebních prvků obsažených v prostředí GRC (děliče bitového toku, filtry apod.).

Výhodou nasazení již hotových bloků (de)modulátorů je výsledná jednoduchost flowgraphu. Nabízí se tak velmi snadná změna vlastností přenosového systému (změna jednoho bloku znamená např. změnu modulačního schématu), značnou nevýhodou je však neúplná dostupnost všech parametrů, které by mohly ovlivnit jeho činnost, obzvláště pak v rádiovém prostředí. Jejich využití je tedy vhodné spíše pro potřeby simulací. Popis níže použitých bloků vychází z jejich oficiální dokumentace. Pokud je dokumentace bloků neúplná, jeho funkce vychází ze známých metod a algoritmů na které se v textu odvolávám. Všechny užití bloky včetně jejich konfigurace jsou pak složeny do jednoho hierarchického bloku, který je dostupný na přiloženém DVD.

### 3.2.1 GMSK modulátor

V kapitole 1.1.1 je popsána podstata frekvenčního klíčování s minimálním kmitočtovým zdvihem. Vhodnou filtrací NRZ modulačních pulzů pomocí Gaussovy dolní propusti před vstupem do frekvenčního modulátoru vznikne modulátor GMSK s výrazně vylepšenou spektrální účinností. Realizaci GMSK modulátoru v GRC lze



Obr. 3.2: Zapojení GMSK modulátoru v prostředí GNU Radio Companion

provést pomocí bloků, které zajistí korektní formátování bitového toku na vstupu, filtraci pravoúhlých pulzů a frekvenční modulaci. K tomu slouží bloky

- Packed to Unpacked,

- Chunks to Symbols,
- Polyphase Arbitrary Resampler,
- Frequency Mod.

Na vstupu prvního bloku jsou vstupní byty rozděleny do bitové posloupnosti, ze které jsou po jednom mapovány do symbolů. Na tyto symboly je aplikována sada filtrů, jejíž definice vychází z parametrů nástroje pro tvorbu FIR filtrů<sup>5</sup>. Parametry Gaussova filtru lze definovat voláním funkce pro návrh filtru pomocí:

```
firdes.gaussian(gain, samples_per_symbol, bt, ntaps),
```

kde *gain*, *samples\_per\_symbol*, *bt*, *ntaps* jsou proměnné zastupující uvažované parametry použitého filtru. V tomto případě je  $BT_B = 0,3$  a počet vzorků připadající na jeden symbol  $SPS = 2$ . Filtrací pravoúhlých pulzů takto definovaným filtrem vznikne filtrovaný modulační signál, který poté vstupuje do frekvenčního modulátoru.

Ten je posledním blokem řetězce s nastavením jediného parametru. Jeho vliv je podrobněji popsán v oficiální dokumentaci, pro požadovaný index modulace (normovaný kmitočtový zdvih – deviační poměr)  $h = 0,5$  lze odvodit, že

$$Sensitivity = \frac{\pi}{2} \frac{1}{Samples\ per\ Symbol} \quad (3.1)$$

Výsledné zapojení sestaveného GMSK modulátoru je znázorněno na obr. 3.2.

### 3.2.2 Konstelační modulátory PSK a QAM

Tvorbu modulátorů založených na jednotlivých konstelačních diagramech je vhodné realizovat voláním několika již hotových funkcí v prostředí GNU Radia. Při návrhu tak odpadá starost např. s výpočtem souřadnic symbolů v konstelaci, mapováním bitových slov jednotlivým symbolům apod. Tyto informace je výhodné uchovávat ve vhodně formátovaných proměnných, tedy objektu *Constellation Rect. Object*. K výpočtu souřadnic symbolů konstelace lze z prostředí `digital` použít funkci `constellation_mpsk().points()`, kde *mpsk* svými hodnotami (*bpsk*, *qpsk*) reprezentuje rozložení odpovídající použité modulační technice, a která dle zvoleného typu konstelačního diagramu modulačního schématu vrací komplexní souřadnice. Po definici konstelace je možno „skládat“ modulátor z následujících bloků:

- Packed to Unpacked,
- Map,
- Differential Encoder,

---

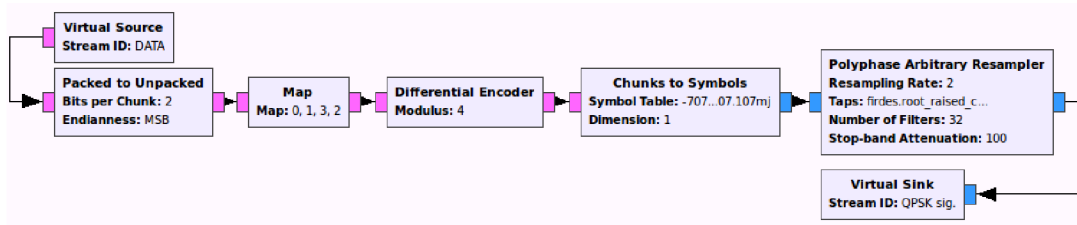
<sup>5</sup>Popis nástroje pro návrh FIR filtrů dostupný na WWW: [https://gnuradio.org/doc/doxygen/classgr\\_1\\_1filter\\_1\\_1firdes.html](https://gnuradio.org/doc/doxygen/classgr_1_1filter_1_1firdes.html)

- Chunks to Symbols,
- Polyphase Arbitrary Resampler.

Sériovým řazením ve vhodném datovém formátu vznikne konstelační modulátor, který na vstupu očekává bytově orientovaná data, které po  $m$ -ticích posílá mapovacímu bloku. Po procesu mapování jsou data diferenčně kódována a následně převedena na symboly v komplexní rovině. Posledním stupněm je filtrace, neboli tvarování pulsů. Tento proces provádí RRC interpolační filtr, který dle vztahu

$$Symbol\ Rate = \frac{Sample\ Rate}{Samples\ per\ Symbol} \quad (3.2)$$

definuje symbolovou rychlost, kde *Sample Rate* je vzorkovací kmitočet a *Samples per Symbol* definuje počet vzorků náležících na jeden symbol konstelace. Potřebným parametrem filtru je mj. tzv. roll-off factor, taktéž označovaný jako *Excess Bandwidth* či parametr  $\alpha$ . Modulátor složený z výše popsanych bloků je zobrazen na obr. 3.3.



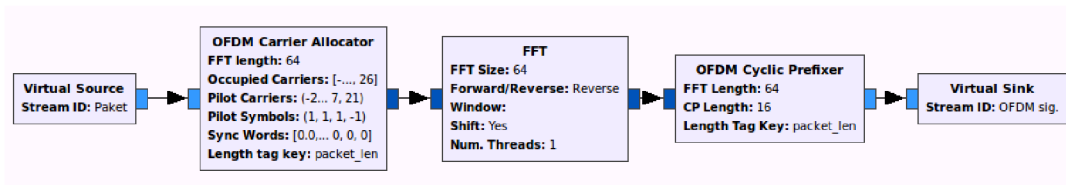
Obr. 3.3: Zapojení QPSK modulátoru v prostředí GNU Radio Companion

Obrázek 3.3 znázorňuje příklad realizace QPSK modulátoru. Patřičnou úpravou konstelačního objektu *Constellation Rect. Object* a mapovacího bloku lze uskutečnit i  $M$ -QAM modulátor. Volanou funkcí z prostředí `digital` pro sestavení konstelace je pak `digital.qam.qam_constellation()`, která je definována pro minimální počet stavů  $M = 16$  a její kompletní dokumentace je dostupná na manuálových stránkách<sup>6</sup>.

### 3.2.3 OFDM modulátor

Pro syntézu OFDM modulovaného signálu lze opět využít hned několik bloků. Nejjednodušším, pro simulace vhodným, blokem je *OFDM Mod*, který nabídne změnu pouze základních parametrů, kterými jsou počet nosných, délka FFT a použitá vnitřní modulace. Dalším použitelným blokem je *OFDM Transmitter*, jež je hierarchickým blokem sdružujícím funkci níže popsanych bloků. Zajímavostí toho bloku je jeho použití pro paketově formátovaná data, tedy data předem definované délky ve vysílači, možnost použití libovolného počtu nosných a jejich kontrolovatelná alokace apod.

<sup>6</sup>Podrobný popis využití funkcí v Pythonu k sestavení QAM konstelace na WWW: [http://gnuradio.org/doc/doxygen/page\\_digital.html#digital\\_python\\_helpers\\_qam](http://gnuradio.org/doc/doxygen/page_digital.html#digital_python_helpers_qam)



Obr. 3.4: Zapojení OFDM modulátoru v prostředí GNU Radio Companion

Protože jde o blok pro přenos paketových dat, předpokládá se na vstupu prvního bloku OFDM vysílače paket. Ten vznikne převodem neznačkovaného toku na značkovaný v bloku *Stream to Tagged Stream*. Sériově řazené bloky na výstupu značkovače si tak budou moci předávat informaci o délce dat, čehož využijí bloky pro formování paketů. Popis formování paketů je nad rámec kapitoly a je podrobněji popsán v referenčním manuálu<sup>7</sup>. Zpracování dat ve vysílači je realizováno bloky

- OFDM Carrier Allocator,
- FFT,
- OFDM Cyclic Prefixer

a je zobrazeno na obr. OFDM. Záhlaví paketu a jeho data jsou tvořena symboly (libovolné  $M$ -PSK či  $M$ -QAM modulace) a sloučena do jednoho komplexního toku, jehož délku nese proměnná `packet_len`. Symboly vstupují do prvního bloku, kde jsou mapovány na jednotlivé nosné. Blok *OFDM Carrier Allocator* umožňuje specifikovat rozsahy nosných, na kterých jsou přenášeny datové symboly (Occupied Carriers) a pilotní symboly (Pilot Carriers, Pilot Symbols). Nastavované hodnoty pro tento blok jsou vektory vektorů, jejichž notace podléhá specifickým pravidlům<sup>8</sup>. Uvažujme následující zápis:

```
occupied_carriers = ((-2, -1, 1, 3), (-3, -1, 1, 2))
pilot_carriers = ((-3, 2), (-2, 3))
```

Z výše uvedeného příkladu tedy vyplývá, že první OFDM symbol přenáší 4 datové symboly, k čemuž jsou použity nosné s indexy  $-2$ ,  $-1$ ,  $1$  a  $3$ . Podobně pro druhý OFDM symbol stejné délky jsou využity nosné  $-3$ ,  $-1$ ,  $1$  a  $2$ . Pro první OFDM symbol zůstaly nosné s indexy  $-3$  a  $2$  neobsazené, je tedy možné je využít pro přenos pilotních symbolů, v případě druhého OFDM symbolu budou použity nosné s indexy  $-2$  a  $3$ . Nosná s indexem  $0$  je stejnosměrná nosná, která nebývá obsazena. Dostupné nosné napravo od stejnosměrné nosné jsou indexovány z rozsahu  $1$  až

<sup>7</sup>Popis paketového přenosu v GRC dostupný na WWW: [http://gnuradio.org/doc/doxygen/page\\_packet\\_data.html](http://gnuradio.org/doc/doxygen/page_packet_data.html)

<sup>8</sup>více o vektorové notaci a indexování nosných dostupné z WWW: [http://gnuradio.org/doc/doxygen/page\\_ofdm.html](http://gnuradio.org/doc/doxygen/page_ofdm.html)

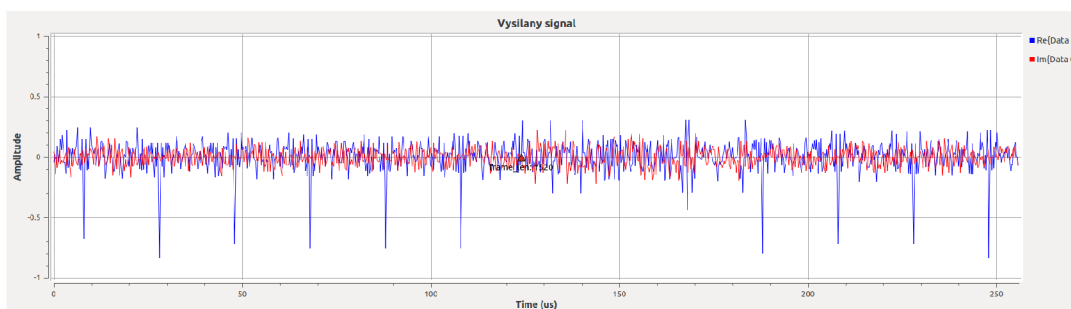


$N/2 - 1$ , kde  $N$  je délka FFT. Podobně pro nosné na nižších frekvencích platí rozsah  $-N/2$  až  $-1$ .

Po převedení signálu do časové oblasti je posledním krokem přidání cyklického prefixu, jehož délka bývá rovna čtvrtině nastavené délky FFT transformace.

### 3.2.4 Vysílání pomocí USRP

Po správné modulaci je na výstupu všech modulátorů komplexní signál (komplexní čísla). Ten je však nutno předávat USRP jednotce ve formátu, který dokáže zpracovat, převést na analogový a vyzářit do prostoru. Základním předpokladem je ošetření možných minimálních a maximálních hodnot, kterých signál v časové oblasti může nabývat. Maximální výkmit signálu na vstupu USRP bloku musí být vždy z rozsahu  $-1$  až  $+1$ . V opačném případě dojde v DAC převodníku k ořezu signálu a vyzáření jeho „neúplného“ průběhu, který na straně přijímače nebude možné zpracovat (dojde k značné deformaci spektra). Tento problém nastává zejména u OFDM signálů,



Obr. 3.5: Patříčně „zeslabený“ signál vstupující do USRP bloku

jehož průběh hodnot v závislosti na parametru PAPR může nabývat velkých rozdílů. Před „připojením“ bloku USRP vysílače je vhodné výstup modulátoru připojit k blokům *QT GUI Time Sink*, resp. *WX GUI Scope Sink* a splnění výše popsaného ověřit z grafických průběhů. Pokud se signál blíží mezím tolerance, je nutné před blok USRP vysílače zařadit násobič, který velikost signálu násobí předem danou konstantou  $K < 1$ . Vhodně zvolenou konstantou je pak možné signál patříčně zeslabit. V případě OFDM signálu je nutno volit  $K = 0,03$ , v případě modulací s jednou nosnou je za „bezpečnou“ hranici považována hodnota  $K = 0,5$  až  $K = 0,7$ .

## 3.3 Návrh přijímačů a demodulace signálu

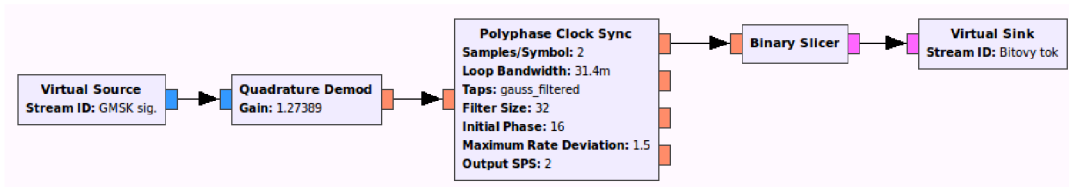
Příjem signálu v práci zastává druhý USRP kit (spojení bod-bod). Jeho zpracování a následná demodulace je však znatelně složitější proces. Nejenže je přijímací strana

zatížena nedokonalostmi samotného rádiového hardwaru, které je nutno kompenzovat, působí zde však i (nežádoucí) vlivy přenosového kanálu. Přijímaný signál je tedy nutno před vstupem do bloku demodulátorů patřičně „upravit“ a rekonstruovat. Tento proces může být navíc ovlivněn i okolními vlastnostmi prostředí, ve kterém je systém provozován (vzdálenost, rušení, vícecestné šíření).

V následujících podkapitolách se zaměřuji na řešení výše popsaných nedostatků a tvorbu bloků demodulátorů. Řešení přijímačů je zpravidla složitější proces než návrh vysílače (vysílače většinou nepřiznivé vlivy neřeší a kompenzaci nechávají pouze na přijímači), proto je tato problematika v blocích GRC většinou ošetřena „uvnitř“ hierarchických bloků demodulátorů. Pro názornost řešení však přijímače tvořím vhodným skládáním jednoúčelových bloků se záměrem dosažení cílů již existujících bloků.

### 3.3.1 Demodulace frekvenčně klíčovaného signálu (GMSK)

Výchozí GMSK demodulátor obsažený v GRC je velmi dobře použitelným již v základu a nevyžaduje jakýchkoli úprav. Použití již hotového bloku samozřejmě bude o něco omezujícím v možnostech nastavení parametrů demodulátoru, nicméně jeho výchozí hodnoty byly experimentálně navrženy tak, aby byly použitelné na většině hardwarových zařízeních, převážně z řad USRP. Činnost již hotového demodulátoru lze pak replikovat pomocí bloků *Quadrature Demod*, *Polyphase Clock Sync* a *Binary Slicer*. Na vstup demodulátoru je přiveden komplexními čísly reprezentovaný signál



Obr. 3.6: Vnitřní struktura GMSK demodulátoru

v základním pásmu a dochází k jeho frekvenční demodulaci. Původní signál byl tvarován pomocí Gaussovy dolní propusti, pro obnovu časování na straně přijímače je opět použit blok *Polyphase Clock Sync* realizovaný bankou filtrů shodných s filtry pro tvarování pulzů ve vysílači. Jejich konfigurace je aplikaci předem známa a lze tak snadno zaručit shodu filtrů v přijímači i vysílači.

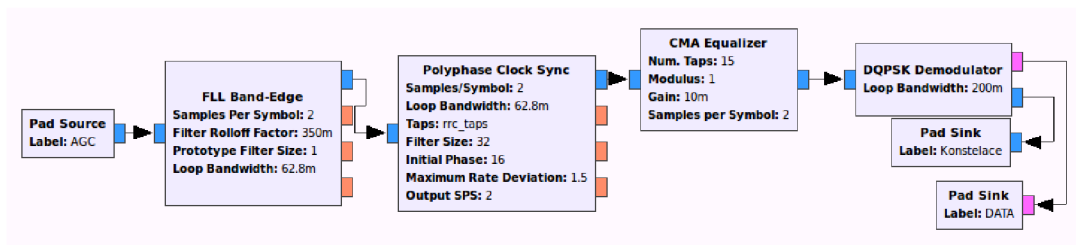
Posledním krokem je obnoveným pulzům v přijímači přiřazení jejich původní bitové hodnoty, což zajišťuje blok *Binary Slicer*. Na výstupu jsou pak byty (osmice bitů) obsahující 1 bit informace.

### 3.3.2 Demodulace symbolů v konstelaci ( $M$ -PSK)

Demodulátor fázově klíčovaného signálu lze opět postavit z velmi obecných bloků GRC. Jsou jimi

- Polyphase Clock Sync,
- CMA/LMS-DD Equalizer,
- Constellation Receiver,
- Differential Decoder,
- Map,
- Unpack K Bits.

Kromě samotné demodulace (vyčítání symbolů z konstelace, mapování bitových slov apod.) je však nutné signál patřičně obnovit. Prvním krokem je obnova časování, tedy hledání vhodného okamžiku pro vzorkování přijatého signálu a tím omezení mezisymbolových přeslechů. K tomu využijeme blok *Polyphase Clock Sync* využívající banky polyfázových filtrů konfigurovaných dle shodných parametrů ve vysílači. Na vstupu bloku je pak přiveden signál s určitým počtem vzorků připadajících na jeden symbol – PCS bloku slouží pouze jako odhad, požadovaný počet vzorků/symbol je dán parametrem `Output SPS`. Signál je také nutné patřičně zesílit – k tomuto účelu lze využít dvou typů ekvalizérů dostupných v GRC. První typ je zastoupen blokem *CMA Equalizer*. Jedná se o jednodušší způsob využívající algoritmus založený na konstantním modulu. Blok nezná žádné parametry konstelace a jejího rozložení, snaží se pouze „udržet“ symboly na jednotkové kružnici. Druhým možným způsobem je užití bloku implementující LMS ekvalizér řízený rozhodováním, který je možno implementovat v bloku *LMS-DD Equalizer*. Ten na rozdíl od CMA ekvalizéru vyžaduje znalost konstelace, kterou akceptuje jako proměnnou definovanou blokem *Constellation Rect. Object*. Použití LMS ekvalizéru je vhodné zejména pro složitější konstelace (16-QAM a více), je proto výpočetně náročnější a pro jednodušší konstelace (BPSK, QPSK) nemusí jeho použití být vhodnou volbou. Po



Obr. 3.7: Zpracování signálu před vstupem do PSK/QAM demodulátoru

korekci časování a ekvalizaci kanálu je potřeba ošetřit možné nepřesnosti frekvence a fáze přijímaného signálu. K tomuto typu synchronizace lze použít blok *Costas Loop* implementující smyčku 2. řádu. Blok vyžaduje pouze znalost PSK konstelace

(počet symbolů). Výše popsaný proces „předzpracování“ přijatého signálu lze realizovat i blokem *Constellation Receiver*, který taktéž implementuje smyčku pro fázovou korekci. Protože je však vhodné mít představu o stavu konstelace, se kterou blok *Constellation Receiver* pracuje (samotný blok nemá možnost, jak tuto informaci „předávat zpět“), je užít blok Costasovy smyčky a blok grafické reprezentace konstelačního diagramu (QT GUI Constellation Sink, resp. WX GUI Constellation Sink) paralelně s výše zmíněným blokem *Constellation Receiver*, který provádí dekódování konstelace. Poslední částí demodulace je dekódování symbolů, mapování bitových slov na základě diferenčního kódování a vhodného formátování bitového toku, který je poslán na výstup demodulátoru.

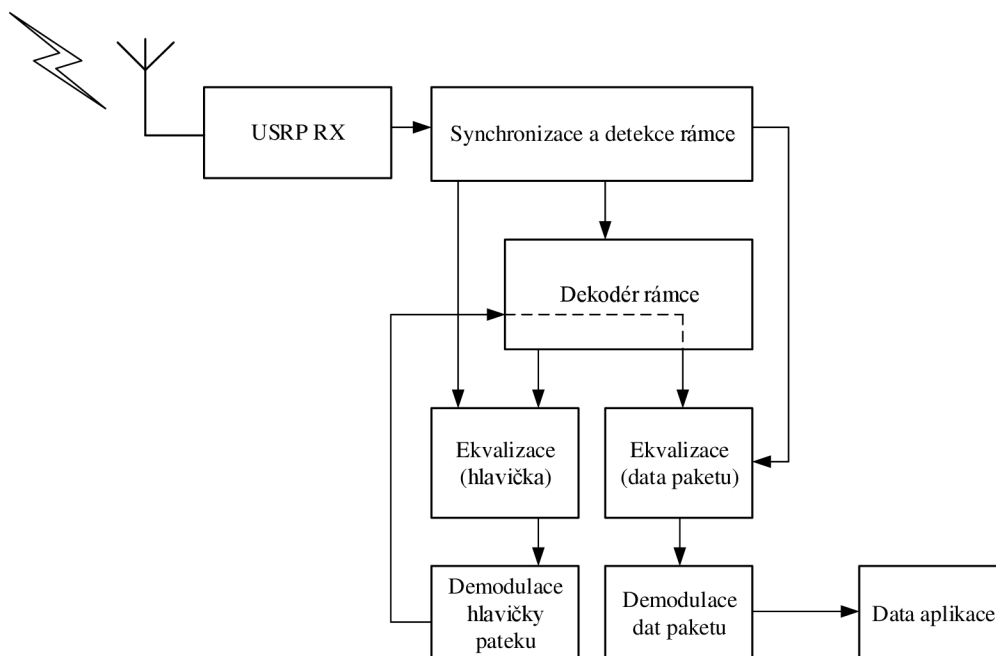
### 3.3.3 Demodulace symbolů v konstelaci ( $M$ -QAM)

Demodulace  $M$ -QAM signálu využívá podobný základ jako demodulátor fázově klíčovaného signálu. Prvním blokem přijímače je vložení FLL využívající okrajověpásmových filtrů (band-edge filtr). Blok frekvenční smyčky je použit pro obnovu nosné a zároveň korekci frekvenční odchylky. Většinu jeho parametrů lze odvodit z parametrů použitých ve vysílači, délka filtru by pak měla odpovídat nastavení filtru pro tvarování pulzů ve vysílači. Po frekvenční korekci následuje opět obnova časování symbolů v bloku PCS pomocí přizpůsobeného filtru. Ten se svými parametry také shoduje s RRC filtrem použitým ve vysílači ke tvarování pulzů a jeho stejnou konfiguraci lze použít i zde. Blok PCS je navržen pro přímý výstup do ekvalizéru. Na rozdíl od PSK modulovaného signálu je však nutno použít ekvalizér řízený rozhodováním, který je v GNU Radiu dostupný blokem *LMS-DD* (*Least Mean Squares - Decision Directed*). Ekvalizér vyžaduje znalost konstelace, které je pak přizpůsobeno jeho rozhodování. Souřadnice jednotlivých symbolů jsou definovány blokem *Constellation Rect. Object*, které po ekvalizaci využije i dekodér konstelace, jež je implementován blokem *Constellation Soft Decoder*. Po demapování symbolů v konstelaci je symbolům přiřazena hodnota bitová hodnota (není zde použito diferenčního kódování) a bity jsou posílány na výstupní port bloku demodulátoru.

### 3.3.4 Příjem a demodulace OFDM signálu

Příjem paketových dat modulovaných pomocí OFDM se skládá z několika dílčích kroků. Nejdůležitější informací pro správné dekódování dat v paketu je informace v jeho hlavičce, která nese typický údaj o délce dat, délce paketu atd. Přijímač nejprve musí detekovat začátek rámce, k čemuž je v GNU Radio dostupný Schmidl-

Coxův algoritmus<sup>9</sup>. Po jemné frekvenční korekci a správné detekci rámce nastává dekódování hlavičky paketu, která může být modulována několika způsoby, a to i zcela odlišně od uživatelských dat (payload) paketu. Na signál je tedy aplikována dopředná FFT transformace a na základě odhadu kvality kanálu probíhá ekvalizace. Paralelní tok dat je poté převeden na sériový a následně probíhá dekódování konstelace (demodulace). Protože má aplikace informaci o způsobu generování hlavičky paketu, je možné za běhu aplikace dle podobně předdefinovaných pravidel data analyzovat a obnovit informaci v nich obsaženou (tuto činnost vykonává tzv. parser hlavičky). Tím je možné zpětně v přijímači obnovit informaci uloženou v hlavičce paketu, kterou pomocí asynchronního předávání zpráv může zpracovat blok *Header/Payload Demux*.



Obr. 3.8: Princip činnosti přijímače OFDM signálu a dekodéru dat

Z hlavičky paketu je nyní aplikaci známa délka paketu. Tuto informaci si mohou opět předávat bloky provádějící dekódování dat paketu (payload). Proces dekódování je tvořen podobnými bloky jako dekodér hlavičky, avšak už pracují s délkou uživatelských dat paketu, nikoli jeho hlavičky. Prvním krokem je tedy opět dopředná FFT transformace, následuje ekvalizace a převod paralelních toků dat na sériový. Dekódování konstelace je posledním krokem, který je závislý na typu použité modulace pro data paketu.

<sup>9</sup>více informací o principu synchronizačního algoritmu dostupné z WWW: <http://cdcpc.ce.ncu.edu.tw/classbackup/wofdms03/ch5/Schmid197.pdf>

OFDM paketový demodulátor implementuje bloky odděleně pro obě části paketu. Jejich činnost je oddělená, avšak nemůže dojít k dekodování dat paketu, pokud nebyla přečtena jeho hlavička, neboť tato informace je dekodérem dat vyžadována<sup>10</sup>. Blokový diagram na obr. 3.8 obecně znázorňuje praktickou činnost takového demodulátoru.

## 3.4 Zabezpečení pomocí FEC API

Od GNU Radio verze 3.7 byl modul `gr-fec` integrován jako standardní součást vývojového prostředí, který nabídne možnost využívat detekci, případně i dopřednou korekci chyb v datových komunikacích. Implementace opravných kódů pro „obecné“ použití v rámci flowgraphu je však relativně složitá a dle typu dat se může značně lišit. Podrobný popis FEC API je dostupný v dokumentaci, v práci se zaměřuji jen na dva typy možných nasazení.

Následující kapitola je věnována popisu dvěma vybraným způsobům dopředného kanálového kódování, které lze realizovat pomocí FEC API.

### 3.4.1 Implementace konvolučního kodéru

Jak bylo zmíněno v kap. 3, datový tok může být dvojího typu, od čehož se odvíjí použití daného FEC kodéru. Pro jednodušší, tzv. „Streaming deployment“ neboli datový tok nekonečné délky, který je v práci využit při implementaci GMSK, PSK a QAM modulátorů, je nasazení konvolučních kodérů realizováno bloky *FEC Extender Encoder*, resp. *FEC Extender Decoder*. Oba bloky akceptují vstupní proměnnou, tzv. *Encoder/Decoder Variable*, která kodéru definuje jeho parametry a funkčnost. Protože se předpokládá nekonečný proud dat, kodér operuje na definovaném počtu vstupních bitů, který lze nastavovat celočíselnou hodnotou parametru `Frame Bits`. Bohužel v době psaní práce není FEC API ještě zcela kompletní a nabídne tak pouze fixní rozsah  $K = 7$  působení konvolučního kodéru a kódový poměr  $1/R$ , kde  $R = 2$ .

Podobná realizace FEC kódování je možná i v případě použití značkování toku dat (tzv. Tagged Stream), pouze informace o bitové délce kódovaného bloku obsažená v proměnné kodéru `Frame Bits` nastavuje maximální velikost, kterou paket nesmí překročit. Samotné kódování pak probíhá výhradně na základě informací o paketu, kterou nese značka (tag). Tuto značku si poté bloky předávají a ke správnému dekodování ji použije i samotný dekodér. Implementace obou typů kodéru i dekodéru je opět v hierarchickém bloku na příloženém DVD – podstatou užití hierarchického bloku je opět sloučení několika funkčních bloků do jednoho univerzálního, který umožní přístup pouze k vyžadovaným parametrům (de)kodéru.

---

<sup>10</sup>K dekodování ve skutečnosti dojde, avšak při práci s chybnou výchozí informací dekodér selže.

## 4 MODULÁRNÍ PŘENOSOVÝ SYSTÉM

Po samotném zhotovení dílčích částí, které vytvoří větve přenosového systému, je možno začít hierarchické bloky skládat do vrstevové struktury s užitím proměnných parametrů, která zajistí korektní implementaci všech požadovaných funkcí. Přenosový systém by tak měl zajistit následující:

- Změna modulačního schématu,
- změna parametrů kodéru,
- změna přenosové rychlosti.

Protože jsou všechny operace (DSP) přenosového systému realizovány samotným skriptem (nikoli v FPGA), při provozu systému může docházet k nesprávné funkci aplikace vlivem nedostatečného výkonu procesoru hostitelské stanice. Z tohoto důvodu je kladen důraz pouze na korektní funkčnost simplexního přenosu mezi dvěma vývojovými kity, obousměrný provoz je pak možný doplněním aplikace pro zpracování dat v „opačném“ směru s užitím frekvenčního duplexu. Následující část je věnována popisu tvorby vysílací a přijímací části, programování modularity systému a zhodnocení jeho provozních vlastností.

### 4.1 Zpracování dat ve vysílači

Začátkem vysílacího řetězce vždy musí být zdroj dat, ideálně pak takových, které lze na straně přijímače korelovat s přijatými daty a zhodnotit tak kvalitu přenosu. K tomuto účelu se v GRC nabízí generátor pseudonáhodných posloupností, které jsou generovány zpětnovazebním posuvným registrem. Zajištěním jeho stejné konfigurace a zvolením stejné délky na straně vysílače i přijímače je tvořena bitová posloupnost délky  $2^S - 1$ , kde  $S$  je stupeň generujícího polynomu. Generovaná data na výstupu GLFSR bloku vstupují do hierarchického bloku kanálového kodéru, který byl vytvořen v kap. 3.4.1, kde jsou rozdělena na úseky předem definované délky a zabezpečeny pomocí konvolučního kodéru. Zabezpečená „uživatelská“ data je před vstupem do modulátoru nutno formovat tak, aby byl přijímač schopen rozpoznat jejich začátek a délku. K tomu lze použít blok *Packet Encoder*<sup>1</sup>, který před samotná data vkládá preambuli a synchronizační směs. Ta na straně přijímače poslouží zejména jako identifikátor začátku uživatelských dat, proto by měla mít tomuto účelu přizpůsobenou hodnotu. Paketový kodér je kombinací několika dílčích funkcí, které navíc zajišťují prokládání bitového toku nulovými vzorky, aby výsledná délka byla násobkem 128 vzorků. Struktura takto vzniklého paketu je zobrazena na obr. 4.1.

---

<sup>1</sup>neplatí při použití OFDM

Paketová data nyní mohou vstupovat do zvoleného bloku modulátoru, který byl vytvořen v předchozích kapitolách. Modulovaný signál je následně předán násobiči konstantou (obvykle  $k = 0,5$ , kde dochází k snížení jeho maximálních úrovní. Výsledný signál je směřován do bloku obsluhující USRP jednotku, kde dochází k jeho transpozici na vysokofrekvenční nosnou vlnu a vyzáření do prostoru.

Preamble	Synchr. vektor	Hlavička	Data	CRC32
4 oktety	1 – 8 oktetů	2×2 oktety	0 – 4096 bytů	4 oktety

Obr. 4.1: Struktura generovaného paketu

Při běhu skriptu vysílače je vykreslován vzorkovaný průběh vysílaného signálu. Je vhodné sledovat úroveň maximálních výchylek, v případě překročení maximální povolené hodnoty je nutné snížit konstantu násobiče pomocí parametru `multiply`.

## 4.2 Implementace přijímače

Na straně přijímače je proces zpracování opačný. Nejprve je signál přijatý druhou USRP jednotkou převeden do základního pásma a vstupuje do zesilovače, který je realizován blokem *AGC* (*Automatic Gain Control*), ze kterého vystupuje do demodulátoru. Bloky demodulátoru jsou zpravidla složitější, neboť v nich mj. dochází také k obnově časování, ekvalizaci a synchronizaci. Popis skladby těchto bloků pro používaná modulační schémata je popsán v předchozí kapitole. Na výstupu demodulátoru jsou datovým proudem jednotlivé  $m$ -tice bitů, které formují rámec, resp. paket (nikoli však paket značkováného toku). Dekodér paketu provádí rámcovou (paketovou) synchronizaci a v bitovém toku na svém vstupu hledá tzv. „Access Code“, neboli synchronizační vektor, který je tvořen bitovou posloupností. Tu je možno nechat vygenerovat automaticky (méně spolehlivé při užití reálného rádia), vhodnější je však zvolit PN posloupnost s pokud možno nejnižší autokorelací. Za synchronizační směsí je přečten obsah hlavičky, která se skládá ze dvou polí – každé pole má až 16 bitů a obsahuje délku paketu, druhé pole je kopií prvního. Použití paketizovaného toku je nezbytné z důvodu funkce přijímací větve – jakýkoli signál dopadající na anténu přijímače tvoří vstupní data aplikace. Na takový nežádoucí signál bude ve flowgraphu reagovat postupně ekvalizér a demodulátor, který bude i nežádoucí signál (chybně) demodulovat a bitový tok postupně projde celým flowgraphem a jakožto neznámá a nežádoucí bitová posloupnost výrazně zvýší parametr bitové chybovosti BER. V případě, že by došlo k demodulaci uživatelských dat, aplikace by nemohla



rozpoznat jejich počátek v bitovém toku, jakákoli informace by tak byla „plovoucí“ v nekonečné bitové posloupnosti.

Po správném přečtení paketu jsou extrahována přenášená data. Délka dat v paketu je konstantní a odpovídá násobku kódované zprávy na výstupu FEC kodéru (při nekorektní detekci paketu je ztrácen obsah takto definované délky). Jeden paket tedy přenáší jedno kódované slovo kodované s poměrem 1/2, které v přijímači vstupuje do dekodéru konvolučního kódu. Na jeho výstupu jsou jednotlivé „unpacked“ bity převedeny na osmice a po bytech zapsány do souboru.

Při běhu skriptu přijímače je vykreslováno spektrum přijímaného signálu a konstelační diagram (pouze PSK/QAM modulace). Je tedy možné sledovat úrovně přijímaného signálu ve frekvenční oblasti, stejně tak lze z konstelačního digramu rychle odhadnout kvalitu příjmu a schopnost demodulace symbolů.

### 4.3 Programování modularity

Cílem modularity přenosového systému je zajištění rychlé možnosti změny jeho parametrů, a to převážně modulačních schémat, parametrů kodéru<sup>2</sup> a přenosové rychlosti dat. Protože však samotný flowgraph neumožňuje bloky přepínat jinak než jejich přepojením (deaktivace stávajícího, vložení a aktivace nového), generovaný zdrojový kód je nutno doplnit o logické podmínky *if-else*, které přepojení požadovaného bloku uskuteční.

Aplikace flowgraphu (Python skript) je typicky spouštěna z okna příkazového interpreta, volba vlastností systému probíhá předáním vstupních argumentů do skriptu v podobě proměnných. GNU Radio rozlišuje více typů proměnných, mezi nejběžnější patří *Variable* a *Parameter*. První jmenovaný typ je použitelný zejména uvnitř skriptu (flowgraphu) a na rozdíl od parametrů jej nelze adresovat z příkazového interpreta. Adresovatelné parametry lze ve flowgraphu vytvořit i v prostředí GRC, kde je možné pro jejich pozdější použití možno definovat jejich název (id), popis, typ a výchozí hodnotu. Parametrů může být ve flowgraphu libovolné množství a přestože jimi zadaná hodnota původně neovlivňuje činnost flowgraphu, lze ji definovat pro pozdější využití při vyhodnocování logických podmínek apod.

Pro přepínání modulačních bloků je vytvořen parametr `mod` s výchozí hodnotou typu textového řetězce „gmsk“. Vyhodnocení tohoto parametru před spuštěním skriptu proběhne vždy, pokud tedy uživatel nezvolí jiný typ, výchozí použitou modulací při přenosu bude GMSK. Změnu přenosové rychlosti systému lze nepřímo ovlivňovat zavedením vícecestavových modulací. Z nastavení bloku *Polyphase Resampler*

---

<sup>2</sup>změna parametrů FEC kodéru pomocí FEC API je zatím omezená, s budoucími aktualizacemi FEC API se však bude postupně rozšiřovat

vyplývá, že symbolová (modulační) rychlost je rovna polovině rychlosti vzorkování, což ve výchozím stavu odpovídá bitové rychlosti 500 kbit/s. Tato rychlost je fixní pro dvoustavovou modulaci GMSK. Použití vícecestavové modulace PSK nabídne změnu přenosových rychlostí dle vztahu

$$v_b = v_{\text{sym}} \log_2(M), \quad (4.1)$$

kde  $M$  je počet stavů vícecestavové modulace. V prostředí GNU Radia nelze přenosovou rychlost nastavit celočíselně hodnotou v bit/s, proto je její volba řešena nepřímo, a to volbou počtu stavů modulační techniky – symbolová rychlost zůstává konstantní, zavedením více možných symbolů v modulačním schématu se rychlost zvětšuje dle vztahu 4.1. Poslední dva parametry, které mají vliv na modularitu systému a které interpret skriptu akceptuje, jsou dvě proměnné konvolučního kodéru. Prvním je kódový poměr `coderate`, který je definován jako  $1/R$ , kde  $R$  lze volit z rozsahu 1 - 2, přičemž kódový poměr 1/1 způsobí odpojení kodéru. Druhým parametrem je délka působení kódu `constraint`, jehož jediná použitelná hodnota v aktuální verzi FEC API je  $K = 7$ . S postupnými aktualizacemi FEC API by se však měly nabídnout i jiné hodnoty, které výše popsány parametry půjde adresovat.

Hierarchické bloky, které budou použity pro ošetření nároků na modularitu je nutno před použitím importovat do prostředí GNU Radia. Import nastane automaticky do adresáře `~/grc_gnuradio/` v domovském adresáři uživatele při spuštění flowgraphu tvořící daný blok, alternativně lze import hierarchického bloku vyžádat v hlavičce každého skriptu ve tvaru

```
from fec_encoder_hier import fec_encoder_hier
from dbpsk_mod_hier import dbpsk_mod_hier
from dqpsk_mod_hier import dqpsk_mod_hier
...
```

S hodnotou parametru `mod` pracují logické podmínky uvnitř skriptu a na základě řetězce v této proměnné dochází k přepojování bloků modulátorů. Toho je docíleno pomocí opakovaných podmínek `if`, které parametr vyhodnocují a podle jeho nastavení volí odpovídající hierarchický blok. Úsek kódu, který vloží vytvořený QPSK modulátor užívající diferenčního kódování je vypsán níže, analogicky pak probíhá zapojení všech ostatních modulátorů. Kompletní obsah skriptu je na příloženém DVD.

```
### pokud je mod = "psk" a symbols = 4
if mod == "psk" and symbols == 4:
    self.dqpsk_mod_hier = dqpsk_mod_hier(
        excess_bw = excess_bw ,
```

```

        nfilts = nfilts ,
        sps = sps ,
        samp_rate = samp_rate ,
    )

```

Nyní je blok modulátoru vložen a inicializován, avšak stále je třeba zajistit jeho propojení s ostatními bloky flowgraphu. Vstupem do bloku modulátoru je formovač paketů, výstupem pak násobič konstantou, jehož účel je popsán v kap. 3.2.4. Připojení těchto bloků je realizováno opět logickou podmínkou (opět je třeba připojovat pouze bloky, které jsou vyžadovány na základě parametru `mod`):

```

## podmínečně připojení DQPSK modulátoru
if mod == "psk" and symbols == 4:
    self.connect((self.dqpsk_mod_hier, 0), (self.multiply, 0))
    self.connect((self.pkt_enc, 0), (self.dqpsk_mod_hier, 0))

```

Obdobně jsou realizovány propojovací funkce pro všechny ostatní modulátory. Podrobněji o programování skriptů v jazyce Python je uvedeno v oficiálním manuálu projektu GNU Radio<sup>3</sup>.

Přehled všech parametrů, které akceptuje vysílač a přijímač využívající přenosy s jednou nosnou je uveden v tab. A.1, resp. A.2. Jak již bylo zmíněno, vysílač i přijímač OFDM signálu je realizován jediným (samostatným) skriptem. Jeho vstupní parametry jsou doplněny o specifické vlastnosti OFDM přenosů a dále jsou popsány v kap. 4.3.1. Příklady použití obou skriptů jsou popsány v příloze A.

### 4.3.1 Řízení OFDM přenosů

Realizace OFDM přenosů je řešena v samostatném flowgraphu, kde jsou vloženy bloky pro tvorbu vysílače i přijímače a které byly popsány v kapitolách 3.2.3 a 3.3.4. Rovněž je implementován konvoluční kodér i dekodér pro zabezpečení přenášených dat, jehož parametry lze volit stejným způsobem jako v předchozím flowgraphu realizující přenos na jedné nosné. Jako vnitřní modulace (modulační technika jednotlivých subnosných) je použita modulace BPSK/QPSK čímž je spolu s možnou realkoací nosných zachována možnost volby přenosové rychlosti systému (symbolová perioda zůstane konstantní). Fázové klíčování je zvoleno jako kompromis mezi dosažitelnou rychlostí systému a přizpůsobivostí nepřilíš kvalitnímu rádiovému prostředí. Z příkazového interpreta lze volitelně určit parametry specifické pro OFDM, a to délku FFT, počet a rozložení nosných, alokaci pilotních nosných, pilotních symbolů

<sup>3</sup>Online verze dostupná z WWW:<http://gnuradio.org/redmine/projects/gnuradio/wiki/TutorialsWritePythonApplications>

a s tím i nastavení synchronizačních slov, které přijímací část použije k odhadu a ekvalizaci kanálu.

Skript pro obsluhu OFDM byl pro přehlednost a vygenerován ze samostatného flowgraphu. Ten je znatelně složitější než při práci s přenosovými technikami využívající jedné nosné, navíc k samotným datům přistupuje odlišným způsobem. Tento skript používá zmiňovaný značkovaný tok dat, ze kterého jsou formovány pakety pomocí zcela odlišných funkcí než v případě přenosů s jednou nosnou. Tento přístup je nutný vzhledem k postupné snaze rozšíření paketově orientovaných flowgraphů, pro jejichž realizaci původně GNU Radio nebylo určeno zcela vůbec. OFDM je v tomto směru stále novinkou a možnosti této přenosové techniky jsou stále předmětem aktivního vývoje. Funkčnost skriptu by však ve výsledku neměla být nijak odlišná, pouze je realizována pomocí sofistikovanějších algoritmů, viz popis OFDM demodulátoru v kap. 3.3.4. Kompletní seznam nastavitelných parametrů pro OFDM vysílač/přijímač je uveden v tab. A.3.

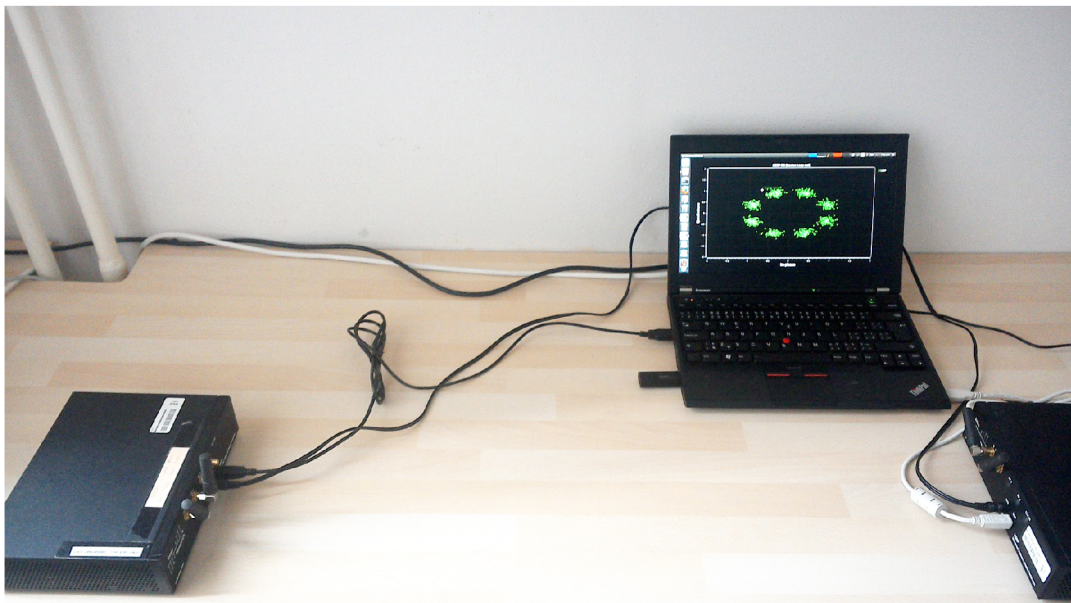
Při běhu OFDM skriptu je vykreslováno spektrum přijímaného signálu a zároveň jsou vypisována sekvenční čísla přijatých paketů. Ty lze použít k hrubému odhadu kvality přenosu a odhadu ztrátovosti.

## 4.4 Dosažené výsledky

Následující kapitola je věnována testování funkčnosti přenosového systému, měření chybovosti a popisu problémů, které jsou specifické pro tvorbu v GRC a užití vývojových kitů USRP1. K popisovaným problémům je dále navrženo řešení, resp. způsob omezení jejich vlivů, které však z technických důvodů nebylo možno realizovat, ať již z důvodů hardwarových nedostatků či principu činnosti programu GNU Radio. Testování funkčnosti přenosového systému probíhá s přímou viditelností obou kitů v konstantní vzdálenosti jednoho metru. Přenos je realizován v ISM pásmu na frekvenci 433,92 MHz (střední kmitočet ISM pásma) a pro minimalizaci možných interferencí pocházejících z jiných vysílačů je přenos uskutečněn v nočních hodinách, kde je předpokládáné využití pásma zcela minimální (sledováno pomocí spektrálního analyzátoru realizovaného pomocí USRP). Jako hostitelská stanice slouží notebook s dvoujádrovým procesorem Intel Core i5-3320M (2,60 GHz) a 16 GB DDR3 RAM. Testovací pracoviště je znázorněno na obr. 4.2. Testování probíhá opakováním přenosu pseudonáhodné posloupnosti. Přijatá bitová posloupnost je porovnávána s originálem<sup>4</sup>, jež je při spuštění vysílače vygenerován a uložen do souboru v binárním formátu.

---

<sup>4</sup>Sledována je zejména rychlost akvizice signálu a demodulace symbolů



Obr. 4.2: Měřící pracoviště pro testování systému

#### 4.4.1 Přenos s užitím frekvenčních modulací

Přenos signálu pomocí GMSK modulace vykazoval nejmenší chybovost při přenosu a oproti řešení s navrženým PSK modulátorem jej lze provozovat bez jakýchkoli přídatným bloků, které by zajišťovaly regeneraci signálu na výstupu USRP. Paketová synchronizace byla zcela bezproblémová a z uvažovaného počtu bitů byly vyhodnoceny všechny jako korektně přenesené. To může být dáno nejen velmi dobrými podmínkami pro přenos, ale i použitím zmíněného FEC kódování<sup>5</sup>. Přenos pomocí této modulační techniky v uvažovaném prostředí lze zhodnotit jako velmi kvalitní.

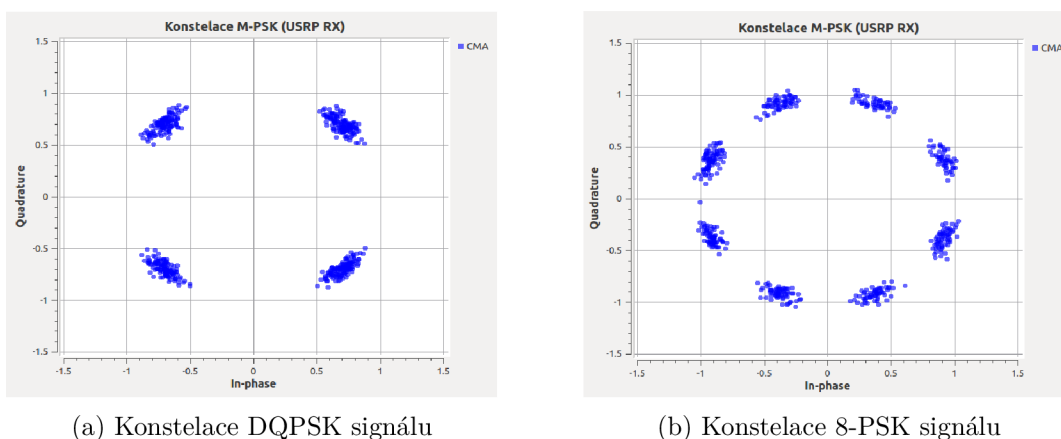
#### 4.4.2 Přenos s užitím fázových modulací

Při přenosu dat pomocí fázově klíčovaného signálu lze volit mezi základní dvoustavovou, čtyřstavovou a osmistavovou modulací. Při dvoustavové modulaci je každý bit reprezentován jedním symbolem, které jsou v IQ diagramu vzájemně posunuty o 180°. Fázové modulace jsou realizovány s použitím diferenčního kódování (neplatí pro 8-PSK), jednotlivé bity tedy nejsou vyjádřeny samotným stavem nosné, ale její změnou ve vztahu k předchozímu symbolu. Při přenosu mezi vývojovými kity nedoházelo k větším problémům, rychlost konvergence ekvalizéru byla dostatečná pro spolehlivý a bezchybný přenos.

---

<sup>5</sup>Vzhledem k realizaci bloků kodéru a dekodéru konvolučního kódu není možné měřit chybovost přenosu před vstupem do dekodéru

Zavedením vícestavové  $M$ -PSK dochází k zvyšování bitové přenosové rychlosti při zachování konstantní modulační rychlosti. Přenosový systém dokáže pracovat až s osmistavovou fázovou modulací, čímž se přenosová rychlost až třikrát zvýší oproti modulaci BPSK. Při provozu vícestavových modulací se však GNU Radio může chovat značně nestabilně, kdy sice dojde k přenosu, ale přenášená data nejsou korektně přičtena v paketovém dekodéru. Tento problém je způsoben nepříliš dobrou optimalizací funkce pro hledání počátku paketu v jeho dekodéru. Pokud nedojde k úspěšné paketové synchronizaci, data vstupující do jeho dekodéru jsou zahazována až do detekce preamble dalšího paketu. Tento problém je možné částečně potlačit použitím kratší délky paketu, kdy délka synchronizační sekvence bude několikanásobně delší než délka přenášených dat. Negativem tohoto řešení je však značně snížená efektivita přenosu. Na kvalitě přenosu se značně projeví i rychlost konvergence použitého ekvalizéru, převážně pak při použití osmistavové modulace. Měřením bylo zjištěno, že použití LMS ekvalizéru s rozhodovací vazbou je vhodnější – prvotní ztráta dat je až 6x menší. Tu je dále možné ještě víc potlačit použitím delší preamble před samotnými daty paketu.



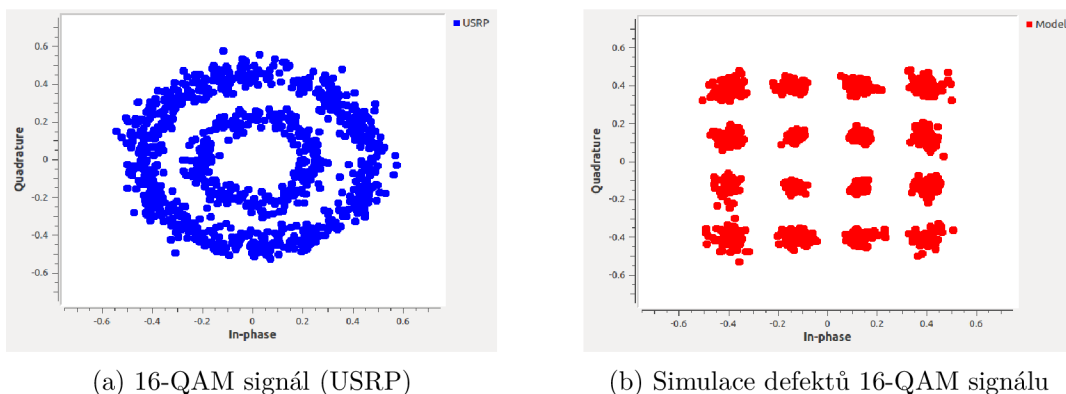
Obr. 4.3: Zřetelné konstelace přijímaného PSK signálu před demodulací

Obrázek 4.3 znázorňuje konstelace přijímaného DQPSK a 8-PSK signálu, jehož symboly jsou mapovány přenášeným dibitům, resp. tribitům. Při využití vícestavových modulací bylo nutné použít větší vzorkovací rychlost (2 MS/s, nebo 4 MS/s a převzorkování). Zatímco přenos pomocí diferenční dvoustavové fázové modulace lze hodnotit jako velmi kvalitní, přenos pomocí vícestavových modulací je zatížen zejména nepříliš optimální implementací paketového dekodéru. Jeho nekorektní funkčnost je zcela nepředvídatelná a vzhledem k nepříliš dobré dokumentaci bloku se nepodařilo modifikovat algoritmus zajišťující paketizaci dat. Vzhledem k poznatku, že nekorektní chování dekodéru nastává zcela náhodně i při přenosu s pouhým propojením bloků modulátoru a demodulátoru v GRC, lze zcela s jistotou vyloučit

hardwarovou závadu. Možným řešením do budoucna by mohlo být použití jiného algoritmu pro paketovou synchronizaci – původní implementace paketového toku je již z pohledu GNU Radia velmi zastaralá a do budoucna se počítá s vývojem algoritmů pro tvorbu paketů pouze značkováného toku (Tagged Stream).

### 4.4.3 Přenos s užitím QAM modulací

Implementace QAM přenosů pro datový přenos byla zcela neúspěšná. Přestože se přenos podařilo uskutečnit v simulovaném prostředí kanálového modelu, kde se úspěšně podařilo kompenzovat nedostatky případného rádiového prostředí, při přenosu pomocí USRP nebylo možné přijímaný signál demodulovat a dekodovat tak bitová slova. Symboly přijímaného signálu v IQ diagramu neustále rotovaly, což může být způsobeno užitím nedostatečné synchronizační techniky na straně přijímače. Bohužel po hlubším zkoumání a diskusi problému na oficiálním fóru lze konstatovat, že GNU Radio nenabízí sofistikovanější metody, kterými tomuto problému předcházet.



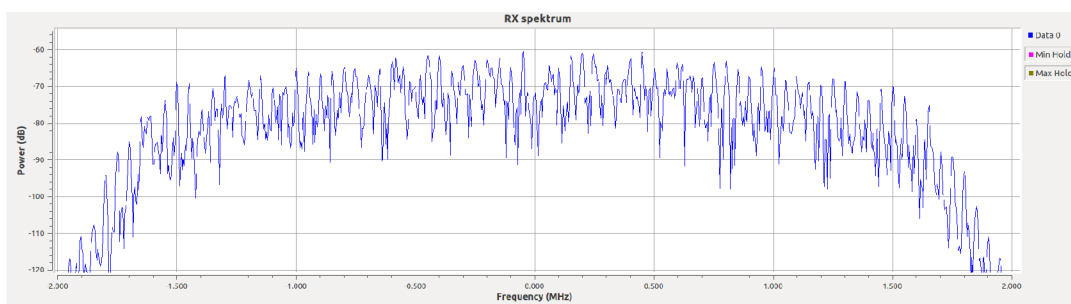
Obr. 4.4: Srovnání konstelace při přenosu pomocí USRP a při simulaci

Výše popsaná situace je zobrazena na obr. 4.4a. Obrázek 4.4b zobrazuje konstelaci signálu, který byl v simulaci zatížen AWGN srovnatelné úrovně, odchylkou frekvence nosné a časování symbolů. Z důvodu praktické nepoužitelnosti byl QAM demodulátor ze skriptu přijímače odstraněn.

### 4.4.4 Přenos s užitím OFDM

Při OFDM přenosu pomocí USRP1 vlivem nastavení malého vzorkovacího kmitočtu (zejména při nastavení 1 MS/s a méně, viz příloha A.2) dochází k špatné synchronizaci a detekci začátku paketu, což znehodnotí značnou část přenášených dat. Tento problém může být způsobem velkou relativní odchylkou mezi kmitočtem vysílače a

přijímače a nastavením většího vzorkovacího kmitočtu (rozšířením spektra) se tento nežádoucí jev stane méně patrným, což vede k značně lepší synchronizaci a detekci paketů pomocí Schmidl-Coxova algoritmu. Při nastavení rychlosti vzorků na 4 MS/s je synchronizace znatelně lepší, avšak za cenu většího požadavku na výkon hostitelské stanice a většího využitého kmitočtového pásma. Tento problém je pro USRP1 dle uživatelských ohlasů běžný a je značným nedostatkem možností nasazení OFDM přenosů na vývojový kit USRP1. Možnou příčinou může být taktéž proces vnitřního převzorkování, ke kterému dochází při zpracování signálu USRP1 jednotkou pro potřeby rychlosti DAC/ADC převodníků. Jejich vnitřní rychlosti jsou odlišné a je možné, že tímto procesem v signále vznikají nepřesnosti. Novější modely vývojového kitu nabízí lepší možnosti volby vzorkovacích rychlostí, jejichž vhodnou volbou lze přenosy s malým počtem nosných realizovat se znatelně lepší spektrální úsporou.



Obr. 4.5: Spektrum OFDM signálu při vzorkování rychlostí 4 MS/s

Příloha A.1 je příkladem výpisu konzolového okna, které slouží pro kontrolu sekvenčních čísel přijímaných paketů při přenosu mezi USRP kity. Vzorky jsou přenášeny s rychlostí 4 MS/s. Přestože je přenos relativně dobrý a paketová detekce je funkční pro delší souvislé celky, výpadek (špatný přenos) jednoho až dvou paketů způsobí ztrátu značného množství dat, které zcela znehodnotí zbytek přenášených dat.

#### 4.4.5 Poznátky z používání SW GNU Radio/GRC

Pro vývoj přenosového systému bylo postupně použito několik verzí vývojového prostředí. Původní vývoj začínal na verzi 3.6.5.1, vzhledem k značným rozdílům použitých bloků při přechodu na nové API ve verzi 3.7 byl celý skript přeprogramován s použitím GNU Radia v postupně uvolňovaných verzích až do aktuální verze 3.7.7.1. Nutno podotknout, že vzájemná kompatibilita mezi jednotlivými verzemi je značně diskutabilní. Nejenže jsou s vývojem nových verzí implementovány nové algoritmy pro stejnojmenné bloky, z nevysvětlitelných důvodů se např. při přechodu mezi verzemi 3.7.6.1 na verzi 3.7.7.1 značně zlepšily přenosové vlastnosti s použitím



OFDM. OFDM je sice předmětem aktivního vývoje, nicméně po diskusi s jedním z autorů vývojového prostředí bylo potvrzeno, že mezi verzemi nebyla implementována žádná změna. Odlišné chování systému tak vyvolává značné pochybnosti o budoucí znovupoužitelnosti navrženého systému.

GNU Radio bylo původně navrženo čistě jako framework pro zpracování signálu. Postupně je však pozorovatelná snaha o vytvoření základu pro přenos a analýzu datových komunikací. Nyní se však vývojové prostředí nachází na pomezí mezi dvěma odlišnými přístupy (značkováná a neznačkováná data). Od původního pohledu na data jako nekonečný tok informací je postupně přeházeno k paketově orientovanému přenosu, zejména pak s rozvojem OFDM. Tím jsou původně navržené přenosové (modulační) techniky zanechány v nepříliš vhodném stavu, absence robustních synchronizačních metod pro QAM jsou pak nepřekonatelným problémem, který značně degraduje uvažovaný přenosový systém s využitím této modulace. Hlavní oblastí využití GNU Radia tedy zůstává spíše simulace a analýza signálu, jako základ pro tvorbu komplexnějších přenosových systémů zatím prostředí není nejvhodnější.

## ZÁVĚR

Bakalářská práce je věnována popisu možností softwarově definovaného rádia a návrhu jeho aplikace pro funkčnost bezdrátového přenosového systému. Kladen je důraz na modularitu tohoto řešení a jeho jednoduchou adaptabilitu pro užití s různými parametry přenosu. Záměrem práce je demonstrace možností realizace přenosových systémů s využitím softwarově definovaného rádia a DSP prováděné specializovaným softwarem, k čemuž je práce navržena jako naučný průvodce.

Do problematiky bezdrátových přenosů je čtenář uveden formou teoretického podkladu, ať již v oblasti používaných modulačních technik, možností řešení synchronizace komunikujících uzlů a v neposlední řadě také kódování přenášené informace. Po úvodním seznámení s koncepcí softwarově definovaných radiových systémů (SDR) začínám tvorbu takového systému postupným vytvořením všech bloků realizujících uvažované modulátory signálu. Dále je popsáno řešení bloků demodulátorů, které jsou doplněny o bloky zajišťující synchronizaci (obnova časování symbolů, ekvalizace apod.) na straně přijímače. Pro jednoduchost manipulace se systémem je vytvořen blok realizující FEC kódér i dekodér. Pro tvorbu zmíněných bloků je užito tzv. flowgraphů, které realizují hierarchické bloky v prostředí GNU Radio Companion. Tímto je čtenář do jisté míry zbaven nutnosti pokročilého programování skriptů v jazycích C++ a Python, nicméně nároky na modularitu tohoto řešení bylo přesto nutné ošetřit alespoň jednoduchými logickými podmínkami. Tímto je zajištěna potřebná modularita řešení, před spuštěním aplikace na straně přijímače a vysílače je tak možno specifikovat modulační schéma, parametry užitého kódování a přenosovou rychlost.

Programování přenosového systému vychází z poznatků simulací přenosových systémů a následně je vytvořený systém nasazen na reálných hardwarových zařízeních. Popis systému, jeho funkčnost a vyhodnocení poznatků z přenosu realizovaných na vývojových kitech USRP1 je uvedeno v kapitole 4. Přestože se některé přenosy nepodařilo realizovat s naprostou spolehlivostí, v kapitole jsou diskutovány možné příčiny neúspěchu a navrženo jejich řešení.

Po dohodě s vedoucím bakalářské práce byl kladen důraz zejména na aspekty simplexního přenosu. V práci jsou popsány možnosti plně duplexního přenosu, které však z důvodů úspory výpočetního výkonu nejsou nasazeny na vývojovém kitu.

## LITERATURA

- [1] BIGLIERI, E. *Coding for Wireless Channels (Information Technology: Transmission, Processing and Storage)*. Springer, 2010. ISBN 978-1441954718.
- [2] HANUS, S. *Nové technologie mobilních komunikací pro integrovanou výuku VUT a VŠB-TUO*. Brno: VUT, 2013. Fakulta elektrotechniky a komunikačních technologií. ISBN 978-80-214-4824-7.
- [3] CHIUEH, T. D., TSAI, P.Y. *OFDM Baseband receiver design for wireless communications*. Singapore: Wiley-Blackwell, 2007. ISBN 978-0470822340.
- [4] MARŠÁLEK, R. *Teorie rádiové komunikace*. Brno: VUT, 2004. Fakulta elektrotechniky a komunikačních technologií.
- [5] ROUPHAEL, Tony J. *RF and digital signal processing for software-defined radio: a multi-standard multi-mode approach*. Amsterdam: Newnes, 2009. ISBN 978-0-7506-8210-7.
- [6] SMĚKAL, Z. *Analýza signálů a soustav*. Brno: VUT, 2012. Fakulta elektrotechniky a komunikačních technologií.
- [7] ŠILHAVÝ, P. *Modulace DMT (Discrete MultiTone* [online]. 2001, poslední aktualizace 31. 1. 2001 [cit. 11. 11. 2014]. Dostupné z URL: <<http://www.elektrorevue.cz/clanky/01006/index.html>>.
- [8] ŠILHAVÝ, P. *Modulace Filtered MultiTone (FMT), implementace ekvalizéru a analýza jeho optimálního řádu*. Elektrorevue, 2010. ISSN 1213-1539.
- [9] ŠIMŠA, J., KASAL, M., HANUS, S., PROKEŠ, A., WILFERT, O., ŠEBESTA, V., *Zpracování radiokomunikačních signálů*. Brno: VUT, Fakulta elektrotechniky a komunikačních technologií.
- [10] WONG, T.F. *Theory of Digital Communication* [online]. Dostupné z URL: <<http://www.wireless.ece.ufl.edu/twong/Notes/Comm/ch3.pdf>>
- [11] ŽALUD, V. *Moderní radiotechnika*. Praha: BEN, 2006, 1. vydání. ISBN: 80-7300-132-1

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

$f_{vz}$	vzorkovací kmitočet
$f_c$	Frekvence nosné
$T_b$	Bitová perioda; doba trvání bitu
$T_s$	Symbolová perioda; doba trvání symbolu
$E_b$	Energie modulovaného signálu na bit
DSP	číslicové zpracování signálů, z angl. Digital Signal Processing PSK Fázové klíčování; klíčování fázovým posuvem
GSM	Globální systém pro mobilní komunikaci, z franc. Groupe Spécial Mobile
MSK	Minimum shift keying; klíčování s minimálním zdvihem
QPSK	Kvadrurní klíčování fázovým posuvem
NRZ	Non-Return To Zero, linkový kód bez návratu k nulové úrovni
QAM	Kvadrurní amplitudová modulace
CR	Obvod pro obnovu nosné vlny, z angl. Carrier Recovery
TIB	Tone in Band
PSAM	Pilot Symbol-Assisted Modulation
$M$ -PSK	$M$ -ary Phase Shift Keying; $M$ -stavová PSK
$M$ -QAM	$M$ -ary Qadrature Amplitude Modulation; $M$ -stavová QAM
PLL	obvod Phase-Locked Loop
DPLL	Digital Phase-Locked Loop
UMTS	Universal Mobile Telecommunications System
SDR	Sowftwarově definované rádio, z angl. Software Defined Radio
USRP	Universal Software Radio Peripheral
FPGA	Programovatelné hradlové pole, z angl. Field Programmable Gate Array

WBX	Daughterboard pro USRP
UHD™	ovladač USRP, z angl. USRP Hardware Driver™
APT	balíkový manažer Advanced Packaging Tool
YUM	balíkový manažer Yellowdog Updater, Modified
GRC	Grafické prostředí GNU Radio Companion
MAC	Vrstva řízení přístupu k médiu, z angl. Media Access Control
XML	Značkovací jazyk, z angl. Extensible Markup Language
CRC	Mechanismus detekce chyby, z angl. Cyclic Redundancy Check
SNR	Veličina udávající odstup signálu od šumu, z angl. Signal to Noise Ratio
CMA	Ekvalizér na bázi konstantního modulu, z angl. Constant Module Algorithm
LMS-DD	Ekvalizér s rozhodovací smyčkou, z angl. Decision directed least mean square
SPS	Počet vzorků na symbol, z angl. Samples per symbol
PCS	Způsob hledání okamžiku vzorkování symbolu
FEC	Kódování s pomocí opravných kódů
FDD	Duplexní přenos s frekvenčním dělením
QT	Knihovna pro tvorbu grafických rozhraní
WX	Knihovna pro tvorbu grafických rozhraní
RRC	Root Raised Cosine filtr

# SEZNAM PŘÍLOH

A	Obsluha skriptů	55
B	Obsah přiloženého DVD	58

## A OBSLUHA SKRIPTŮ

Přenosový systém je spouštěn pomocí skriptu pro obsluhu vysílače a přijímače odděleně. V případě použití skriptu pro přenos na jedné nosné musí být parametry na straně přijímače i vysílače shodné. Jejich kompletní seznam je uveden v tab. A.1. Spuštění skriptu vysílače a vypsání jeho nápovědy (seznam akceptovatelných parametrů) je poté možné pomocí příkazu:

```
$ ./usrp_tx.py --help
$ ./usrp_ofdm.py --help
```

Spuštěním skriptu bez jakýchkoli parametrů se vysílač (přijímač) nakonfiguruje dle výchozích hodnot jednotlivých parametrů. Výchozí hodnoty pro jednotlivé parametry jsou uvedeny v tab. A.1 a A.2. Spuštění skriptu přijímače probíhá obdobným způsobem, navíc jsou akceptovány parametry pro nastavení šířky Costasovy smyčky a zesílení ekvalizéru.

Tab. A.1: Parametry skriptů pro obsluhu vysílače a přijímače s jednou nosnou

Parametr	ID	Výchozí hodnota	Typ
Modulace	--mod	gmsk	string
Počet stavů modulace	--symbols	2	int
Frekvence	--freq	433.92e6 Hz	float
Počet vz. na symbol	--sps	2	int
Konstanta násobiče	--multiply	0.5	float
Adresa USRP kitu	--address	serial=E4R22Y4U	string
Délka PN posloup.	--pn-len	32	int
Kódový poměr	--coderate	2	int
Délka působení	--constraint	7	int
Rychlost vzorků	--samp_rate	1 MS/s	int
Nastavení zesilovače	--gain	15	float

Tab. A.2: Přidané parametry skript pro obsluhu přijímače s jednou nosnou

Parametr	ID	Výchozí hodnota	Typ
Zesílení ekvalizéru	--eq-gain	10e-3	float
Šířka Costasovy smyčky	--phase-bw	300e-3	float

Řízení OFDM přenosů probíhá stejným způsobem jako přenosového systému s jednou nosnou. Dostupné parametry pro skript jsou uvedeny v tab. A.3.

Tab. A.3: Parametry skriptu pro obsluhu OFDM vysílače a přijímače

Parametr	ID	Výchozí hodnota	Typ
Frekvence [Hz]	--freq	433.92e6 Hz	float
Zesílení TX [dB]	--tx-gain	15	float
Zesílení RX [dB]	--rx-gain	15	float
Modulace subnosných	--mod	bpsk	string
FFT počet vz.	--fft	64	int
Adresace subnosných	--data-carriers	(-26, ..., 27)	-
Alokace pilot. tónů	--pilot-tones	(-21, -7, 7, 21)	-
Kódový poměr KK	--coderate	2	int
Délka působení KK	--constraint	7	int
Interpolační faktor	--interpolation	1	int
Délka paketu	--packet-len	96	int
Délka PN posloup.	--pn-len	30	int

Při běhu skriptu je zobrazováno spektrum přijímaného OFDM signálu a jsou vypisována sekvenční čísla paketů. Příklad posloupnosti pro popisovanou chybovost přenosu (detekce) je zobrazen v textovém poli A.1. Opět lze spuštěním skriptu s parametrem `help` vypsát nápovědu a dostupné parametry pro použití skriptu.

Výstup A.1: Paketové sekvence při přenosu s rychlostí vzorků 4 MS/s

Offset: 674368	Source: n/a	Key: packet_num	Value: 3486
Offset: 674752	Source: n/a	Key: packet_num	Value: 3487
Offset: 675136	Source: n/a	Key: packet_num	Value: 3488
Offset: 675520	Source: n/a	Key: packet_num	Value: 3489
Offset: 675904	Source: n/a	Key: packet_num	Value: 3490
Offset: 676288	Source: n/a	Key: packet_num	Value: 3491
Offset: 676672	Source: n/a	Key: packet_num	Value: 3492
Offset: 677056	Source: n/a	Key: packet_num	Value: 3493
Offset: 677440	Source: n/a	Key: packet_num	Value: 3494
Offset: 677824	Source: n/a	Key: packet_num	Value: 3495
Offset: 678208	Source: n/a	Key: packet_num	Value: 3496
Offset: 678592	Source: n/a	Key: packet_num	Value: 3498
Offset: 678976	Source: n/a	Key: packet_num	Value: 3499
Offset: 679360	Source: n/a	Key: packet_num	Value: 3501
Offset: 679744	Source: n/a	Key: packet_num	Value: 3502

Uvedená textová pole obsahují sekvenci 15 po sobě detekovaných paketů, které obsahují uživatelskou informaci. Z hodnot sekvenčních čísel (položka `packet_num Value`)



je patrné, že při rychlosti 1 MS/s je přenos prakticky nerealizovatelný. Možná příčina tohoto problému je popsána v kap. 4.4.4.

Výstup A.2: Paketové sekvence při přenosu s rychlostí vzorků 1 MS/s

Offset : 674368	Source : n/a	Key : packet_num	Value : 525
Offset : 674752	Source : n/a	Key : packet_num	Value : 643
Offset : 675136	Source : n/a	Key : packet_num	Value : 644
Offset : 675520	Source : n/a	Key : packet_num	Value : 800
Offset : 675904	Source : n/a	Key : packet_num	Value : 828
Offset : 676288	Source : n/a	Key : packet_num	Value : 882
Offset : 676672	Source : n/a	Key : packet_num	Value : 889
Offset : 677056	Source : n/a	Key : packet_num	Value : 992
Offset : 677440	Source : n/a	Key : packet_num	Value : 955
Offset : 677824	Source : n/a	Key : packet_num	Value : 1156
Offset : 678208	Source : n/a	Key : packet_num	Value : 1266
Offset : 678592	Source : n/a	Key : packet_num	Value : 1325
Offset : 678976	Source : n/a	Key : packet_num	Value : 1338
Offset : 679360	Source : n/a	Key : packet_num	Value : 1438
Offset : 679744	Source : n/a	Key : packet_num	Value : 1573

## B OBSAH PŘILOŽENÉHO DVD

Příložené DVD obsahuje elektronickou verzi práce ve formátu PDF a soubory pro práci s vytvořenou aplikací. Dále je obsažen výsledný skript v jazyce Python pro programování radio HW, zdrojové soubory v jazyce Python a jejich flowgraphy typu *.grc* pro práci s hierarchickými bloky. Flowgraphy jsou vytvořeny v SW GNU Radio Companion v aktuální verzi 3.7.7.1 a jím kompilované skripty jsou spustitelné v libovolném textovém editoru. Pro přehlednost a zachování formátování (interpret jazyka Python je na formátování velmi citlivý) je doporučeno skripty prohlížet pomocí programů PSpad nebo Notepad++. Kompletní obsah příloženého média je uveden níže:

Složka *grc\_hier\_bloky* obsahuje:

- *qam\_demod\_hier.grc*
- *bpsk\_demod\_hier.grc*
- *qpsk\_demod\_hier.grc*
- *epsk\_demod\_hier.grc*
- *qam\_mod\_hier.grc*
- *bpsk\_mod\_hier.grc*
- *qpsk\_mod\_hier.grc*
- *epsk\_mod\_hier.grc*
- *fec\_encode\_hier.grc*
- *fec\_decode\_hier.grc*
- *fec\_tagged\_encode\_hier.grc*
- *fec\_tagged\_decode\_hier.grc*

Složka *skripty* obsahuje hlavní skripty<sup>1</sup> pro obsluhu přenosů:

- *usrp\_tx.py* pro obsluhu vysílače
- *usrp\_rx.py* pro obsluhu přijímače
- *usrp\_ofdm.py* pro řízení OFDM přenosů

Složky *GRC models* obsahuje flowgraphy pro generování hlavních skriptů<sup>2</sup>.

- *usrp\_tx.grc* pro obsluhu vysílače
- *usrp\_rx.grc* pro obsluhu přijímače
- *usrp\_ofdm.grc* pro řízení OFDM přenosů

---

<sup>1</sup>testováno na OS Ubuntu 14.04.4

<sup>2</sup>v nově generovaném skriptu nebude ošetřena modularita