VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

ADAPTIVNÍ OFDM

ADAPTIVE OFDM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. DAN NOWAK
AUTHOR

VEDOUCÍ PRÁCE doc. Ing. ROMAN MARŠÁLEK, Ph.D.
SUPERVISOR

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student:	Bc. Dan Nowak	***ID:***	78641
Ročník:	2	***Akademický rok:***	2010/2011

NÁZEV TÉMATU:

Adaptivní OFDM

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s technikami pro adaptivní nastavování parametrů systémů s více nosnými, zejména OFDM. Vypracujte podrobnou rešerši, týkající se problematiky zjišťování aktuálního stavu rádiového kanálu a problematiky adaptivního nastavování typu modulace a výkonu na jednotlivých nosných.

V prostředí MATLAB vytvořte funkční simulaci metody pro zjišťování stavu rádiového kanálu (přenos, obsazenost...) a několika metod pro adaptivní nastavování parametrů OFDM.

Vybrané metody implementujte ve vybraném komerčně dostupném softwarovém rádiu. Funkčnost ověřte měřením.

DOPORUČENÁ LITERATURA:

[1] GLISIC, S., LORENZO, B. Advanced Wireless Networks: Cognitive, Cooperative & Opportunistic 4G Technology, New York: J. Wiley & Sons, 2009

Termín zadání:	7.2.2011	***Termín odevzdání:***	20.5.2011

Vedoucí práce: doc. Ing. Roman Maršálek, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Cílem této práce je prostudovat možnosti adaptivního nastavení systému OFDM. Studie řeší způsob získání potřebných informací o jednotlivých frekvenčních subkanálech pomocí pilotních signálů a využití těchto informací k nastavení systému. Práce představuje princip water-fillingu pro přidělování výkonu subkanálům na základě jejich SNR a mechanismy adaptivní změny modulace pro různá využití systému OFDM. Tyto mechanismy jsou implementovány do modelu v prostředí MATLAB. Kód je pak převeden do grafické podoby v prostředí Simulink a s využitím USRP2 je signál vysílán reálným přenosovým prostředím.

Klíčová slova

OFDM, modulace, adaptivní, water-filling, SNR, BER, MATLAB, Simulink, USRP2

Abstract

The purpose of this paper is to study the possibilities of the OFDM system adaptive modulation. The study presents mechanism for acquiring the necessary data of every frequency subchannel using pilot signals and application of these data for system adjustment. The paper introduces the water-filling principle for subchannel power allocation on the basis of their SNR and adaptive modulation mechanisms for variety of OFDM usage. These mechanisms are implemented into a MATLAB model. The code is then transformed into graphical representation and using USRP2, the signal is transmitted by real world channel.

Keywords

OFDM, modulation, adaptive, water-filling, SNR, BER, MATLAB, Simulink, USRP2

NOWAK, D. *Adaptivní OFDM.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 43 s. Vedoucí diplomové práce doc. Ing. Roman Maršálek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Adaptivní OFDM jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících 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.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

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

V Brně dne ....

(podpis autora)

Obsah

Úvod **1**

1 Obecná charakteristika systému OFDM **2**

2 Odhad kanálu **3**

2.1 Blokový typ pilotního signálu 3

2.2 Hřebenový typ pilotního signálu 4

2.3 Odhad kanálu dle standardu IEEE 802.11 6

3 Adaptivní nastavování parametrů kanálů **7**

3.1 Přidělování výkonu a bitů jednotlivým kanálům 7

3.1.1 Maximalizace přenosové rychlosti 8

3.1.2 Minimalizace potřebného výkonu 9

3.2 Volba modulace v kanálech 10

3.2.1 Algoritmus pevného práhu 11

3.2.2 Algoritmus odhadu BER subpásma 12

3.2.3 Algoritmus konstantní datové rychlosti 12

4 Simulace v prostředí MATLAB **13**

4.1 Úvodní odhad kanálu 15

4.2 Algoritmus Greedy 18

4.2.1 Metoda přičítání (funkce GreedyAdd) 18

4.2.2 Metoda odečítání (funkce GreedySub) 19

4.3 Alokace výkonu pro metodu přičítání 20

4.4 Alokace bitů a vnitřní modulace 21

4.5 Vysílání a příjem OFDM symbolů, průchod kanálem 22

5 Implementace do USRP2 v prostředí Simulink **25**

5.1 Převedení kódu na model v prostředí Simulink 26

5.1.1 Pilotní signál a jeho vysílání 26

5.1.2 Odhad přenosu, Greedy a potřebný výkon 27

5.1.3 Alokace bitů a modulace 31
5.1.4 Přenos OFDM symbolů 34
5.1.5 Zaokrouhlování 34
5.1.6 Demodulace 36
5.2 Úspora výkonu při použití USRP2 37
6 Shrnutí 39
Seznam použité literatury 41
Seznam zkratek 43

Seznam obrázků

Obr. 1.1: Princip ortogonality OFDM [17] ... 2
Obr. 2.1: Blokový typ pilotního signálu [10] ... 3
Obr. 2.2: Hřebenový typ pilotního signálu [10] ... 4
Obr. 2.3: Blokové znázornění algoritmu LMS pro hřebenový typ pilotního signálu [6] ... 5
Obr. 3.1: Princip algoritmu water-filling [7] ... 7
Obr. 3.2: Grafické vyjádření vzorce 3.11 [7] ... 9
Obr. 3.3: Systémový model OFDM [8] ... 10
Obr. 4.1: Blokové schéma simulovaného modelu [10] ... 13
Obr. 4.2: Vývojový diagram simulace v prostředí MATLAB ... 14
Obr. 4.3: Nosné vstupující do IFFT [10] ... 15
Obr. 4.4: Úvodní odhad přenosu a útlumu na jednotlivých subnosných ... 16
Obr. 4.5: Konstelační diagram přijatého signálu bez ekvalizace ... 17
Obr. 4.6: Konstelační diagram přijatého signálu po použití zero-forcing ekvalizéru ... 17
Obr. 4.7: Rozdělení bitů na subnosné při metodě přičítání a počtu vysílaných bitů 150 ... 19
Obr. 4.8: Rozdělení bitů na subnosné při metodě odečítání a počtu vysílaných bitů 150 ... 20
Obr. 4.9: Úspora výkonu při různém počtu vysílaných bitů ... 21
Obr. 4.10: Konstelační diagramy modulací použitých v modelu [18] ... 22
Obr. 4.11: Konstelační diagram přijatého signálu bez ekvalizace ... 23
Obr. 4.12: Konstelační diagram přijatého signálu po ekvalizaci (pro 156 vysílaných bitů) ... 23
Obr. 4.13: Konstelační diagram přijatého signálu po ekvalizaci (pro 260 vysílaných bitů) ... 24
Obr. 5.1: USRP2 ... 25
Obr. 5.2: Parametry připojeného rádia zobrazující se v prostředí Simulink ... 26
Obr. 5.3: Krok 1 - přidání nul a CP ... 27
Obr. 5.4: Krok 1 – vysílač a přijímač USRP2 ... 27
Obr. 5.5: Krok 2 – odhad přenosu ... 28
Obr. 5.6: Krok 2 - detail bloku Upravy ... 28
Obr. 5.7: Krok 2 – Greedy algoritmus (1. část) ... 28
Obr. 5.8: Krok 2 - detail bloku If Action Subsystem (úroveň 1) ... 29
Obr. 5.9: Krok 2 - detail bloku Switch Case Action (úroveň 2) ... 30
Obr. 5.10: Krok 2 – výpočet celkového potřebného výkonu pro každou nosnou ... 30
Obr. 5.11: Krok 2 – použité paměťové bloky ... 31
Obr. 5.12: Krok 2 – generace dat k vysílání ... 31
Obr. 5.13: Krok 3 – Greedy algoritmus (2. část) ... 32
Obr. 5.14: Krok 3 – Alokace bitů a modulace ... 32
Obr. 5.15: Krok 3 – detail bloku Switch Case Action pro modulaci 16QAM ... 33
Obr. 5.16: Načítání symbolů ... 34
Obr. 5.17: Výběr symbolů, odstranění CP a nul ... 34
Obr. 5.18: Zaoukrouhlování symbolů ... 35
Obr. 5.19: Zaokrouhlovací mřížka pro modulaci 16QAM [17] ... 35
Obr. 5.20: Demodulace ... 36
Obr. 5.21: Výpočet úspory výkonu (úroveň 1) ... 37
Obr. 5.22: Výpočet úspory výkonu (úroveň 2) ... 37
Obr. 5.23: Výpočet úspory výkonu (úroveň 3) ... 38
Obr. 5.24: Úspora výkonu při různém počtu vysílaných bitů v prostředí Simulink ... 38

Úvod

Účelem této práce je prostudovat možnosti adaptivního nastavování parametrů systémů s více nosnými, především OFDM. Teoreticky popsané metody získávání informací o subkanálech pak budou převedeny do kódu v systému MATLAB, kde bude vytvořena simulace rádiového kanálu s vysíláním systému OFDM využívajícího tohoto adaptivního nastavování parametrů, především přidělování bitů jednotlivým subkanálům v závislosti na jejich přenosu. Toto nastavení sníží potřebný výkon k vyslání stejného počtu bitů při stejné chybovosti jako bez použití adaptivního nastavování parametrů.

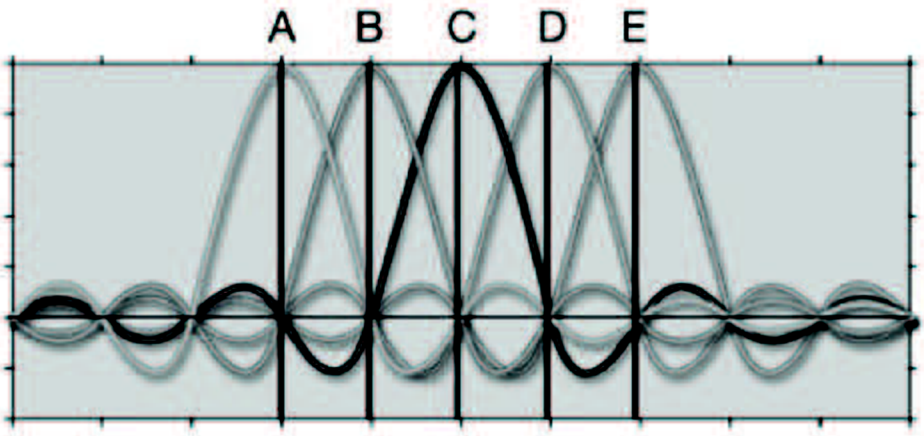
Další část práce je zaměřena na praktické využití algoritmů v softwarovém rádiu USRP2 pracujícím ve volném pásmu ISM. V prostředí Simulink je kód převeden do grafické podoby a pomocí softwarového rádia je signál vysílán na frekvenci 2,451234 GHz. Konečným cílem této práce je srovnání úspory výkonu s reálnými přenosovými prvky s teoretickými hodnotami vypočtenými v prostředí MATLAB.

1 Obecná charakteristika systému OFDM

Spolu s rozmachem bezdrátového vysílání v posledních dekádách rostlo i zahlcení frekvenčních pásem. Jednotlivá vysílání digitálního signálu na jedné nosné frekvenci jsou doprovázena nežádoucími vlivy šumu, interferencemi s jinými signály a mnohacestným šířením. Řešení těchto problémů zahrnují např. rozšíření kmitočtového pásma, což má za následek snížení SNR (poměr signál/šum), či snížení ISI (mezisymbolových interferencí) při vícecestném šíření signálu pomocí snížení přenosové rychlosti. Tato řešení ale nebyla dostatečná, proto byl vytvořen standard OFDM.

Modulační metoda OFDM (Orthogonal Frequency Division Multiplexing) spočívá v použití několika desítek až tisíců nosných kmitočtů. Nosné na těchto frekvencích jsou dále modulovány dle potřeby různě robustními modulacemi, od BPSK, přes QPSK, po M-QAM. Jednotlivé signály na nosných jsou vzájemně ortogonální (tj. jejich skalární součin je nulový), takže maximum každé nosné by se mělo překrývat s průchody nulou ostatních nosných [1] (viz obr. 1.1). Tento princip převádí jeden širokopásmový frekvenční kanál na množství úzkopásmových subkanálů, jejichž ortogonalita zaručuje šetření frekvenčního pásma a účinný přenos dat v bezdrátovém disperzním prostředí s vícecestným šířením signálu.

Díky vysoké datové propustnosti, flexibilitě a jednoduché implementaci se OFDM stalo základem v současných standardech jako DSL (Digital Subscriber Line), LTE Advanced (Long Term Evolution), DAB (Digital Audio Broadcasting), DVB (Digital Video Broadcasting), WLAN (Wireless Local Area Network) IEEE 802.11 a-g, a WiMAX (Worldwide Interoperability for Microwave Access) IEEE 802.16.



Obr. 1.1: Princip ortogonality OFDM [17]

2 Odhad kanálu

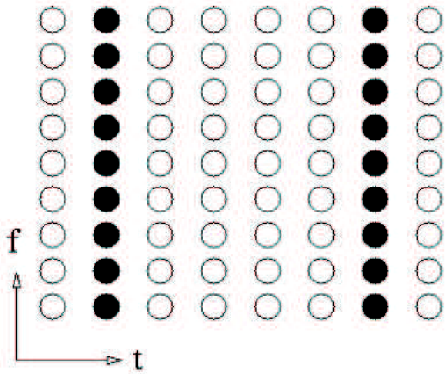
V prostředí s více uživateli mohou být jednotlivé subnosné přiřazeny různým uživatelům, takže je dostupné spektrum rozděleno mezi více stanic. Tento princip je základem OFDMA (OFDM Multiple Access). Energetická účinnost a vysoké přenosové rychlosti jsou důležitými pro tvorbu systémů založených na této přístupové metodě. Proto jsou využívány dva způsoby optimalizace:

- Rate adaptive optimalization, která maximalizuje celkovou rychlost při daném výkonu systému
- Margin adaptive optimization, která minimalizuje potřebný výkon při dané přenosové rychlosti systému

Optimální rozdělení prostředků vyžaduje přesný odhad stavu kanálu, který se provádí pomocí tréninkových sekvencí, tzv. pilotů. Pilot je referenční signál použitý vysílačem i přijímačem sloužící k identifikaci parametrů kanálu.

2.1 Blokový typ pilotního signálu

Blokový typ pilotního signálu využívá všechny nosné pro přenos tréninkové sekvence ve stejném okamžiku (viz obr. 2.1 – černé body jsou pilotní signály, bílé body užitečná data). Je zde využit algoritmus LS (Least Square – metoda nejmenších čtverců) nebo MMSE (Minimum Mean Square Error), popř. interpolace s rozhodovací zpětnou vazbou.



Obr. 2.1: Blokový typ pilotního signálu [10]

Algoritmus LS převzatý z [2], jehož vstupním signálem je X a výstupním signálem Y, je jednoduchý:

$$h_{LS} = X^{-1} \cdot Y, \tag{2.1a}$$

kde

$$X = diag\{x_0, x_1, x_2 \dots x_{N-1}\} \tag{2.1b}$$

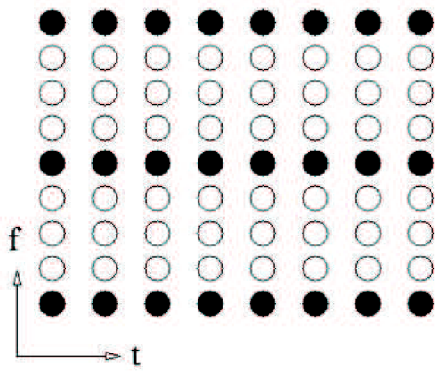
$$Y = \begin{bmatrix} y_0 \\ \vdots \\ y_{N-1} \end{bmatrix}, \qquad (2.1c)$$

ale trpí vysokou chybou MSE (Mean Square Error), což je rozdíl mezi odhadem kanálu a jeho reálnými parametry. Proto je používanější složitější algoritmus MMSE s různými modifikacemi, které mají za úkol zjednodušení algoritmu (např. OLR-MMSE – Optimal Low-Rank MMSE).

V blokovém typu pilotního signálu bývá rovněž použita interpolace s rozhodovací zpětnou vazbou. Ostatní přístupy používají stejný odhad po dobu celého bloku, zatímco pomocí rozhodovací zpětné vazby se odhad mění v průběhu bloku, čímž se opět zvyšuje přesnost odhadu.

2.2 Hřebenový typ pilotního signálu

Hřebenový typ pilotního signálu (viz obr. 2.2) umožňuje odhad kanálu pomocí algoritmu LS a LMS (Least Mean Squares). Základem zde je signál X_k, do kterého je vloženo N_p pilotů a poměr počtu nosných / pilotů L.



Obr. 2.2: Hřebenový typ pilotního signálu [10]

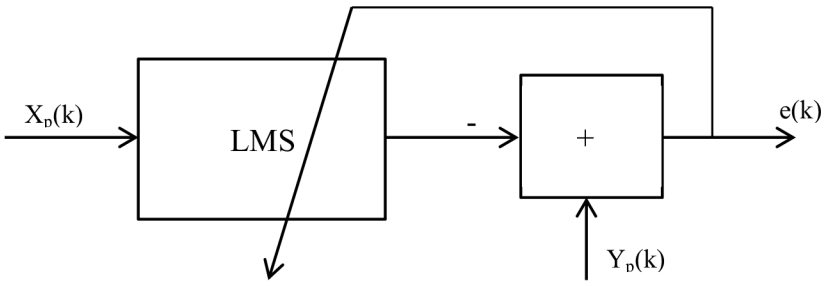
$$X(k) = X(mL + l) \qquad (2.2)\ [5]$$

$$= \begin{cases} x_p(m), & l = 0 \\ inf.data, & l = 1, \dots L-1 \end{cases}$$

Vstupem do LMS (viz obr. 2.3) je signál $X_p(k)$, kde k je pořadí pilotu vstupujícího do systému, $Y_p(k)$ je výstup téhož pilotu a $e(k)$ je chyba, která je znovu načtena do LMS a tímto vyloučena.

Algoritmus LS je zde opět jednoduchý:

$$H_p(k) = \frac{Y_P(k)}{X_P(k)} \qquad k = 0,1,\dots N_p - 1. \qquad (2.3)\ [5]$$



Obr. 2.3: Blokové znázornění algoritmu LMS pro hřebenový typ pilotního signálu [6]

Ve hřebenovém pilotním signálu lze využít některou z 5 druhů interpolací:

- Lineární interpolace
- Interpolace druhého řádu
- Interpolace v časové oblasti
- Interpolace dolní propustí (funkce *interp* v prostředí MATLAB)
- Splajnová interpolace (funkce *spline* v prostředí MATLAB)

Pro lineární interpolaci platí rovnice

$$H_e(k) = H_e(mL + l) = \left(H_P(m+1) - H_P(m)\right)\frac{l}{L} + H_p(m), \qquad (2.4)\ [5]$$

kde k je index nosné, H_e je celkový odhad kanálu, H_p je odhad kanálu pomocí LS a platí $0 \leq l < L$ a $mL < k < (m+1)L$.

Interpolace druhého řádu upravuje dle [5] rovnici (1.4) do tvaru

$$H_e(k) = c_1 H_p(m-1) + c_0 H_P(m) + c_{-1} H_P(m+1), \qquad (2.5a)$$

kde

$$c_1 = \frac{\alpha(\alpha-1)}{2} \qquad (2.5b)$$

$$c_0 = -(\alpha-1)(\alpha+1) \qquad (2.5c)$$

$$c_{-1} = \frac{\alpha(\alpha+1)}{2} \qquad (2.5d)$$

$$\alpha = \frac{l}{N} \qquad (2.5e)$$

Interpolace dolní propustí vkládá nuly do původní sekvence, výsledná sekvence je filtrována speciálně vygenerovanou dolní propustí, která odfiltruje nuly, nechává původní sekvenci nezměněnou a zároveň interpoluje vstupní signál tak, aby chyba MSE byla co nejnižší a výstupní signál měl stejný počet prvků jako vstupní signál.

Simulace v [5] ukazují, že nejvhodnější technikou odhadu kanálu je hřebenový typ pilotního signálu s interpolací dolní propustí. Hřebenový typ je totiž vhodnější než blokový pro sledování kanálů s rychlým únikem a interpolace dolní propustí minimalizuje chybu mezi interpolovanými body a jejich ideálními hodnotami nejvíce ze všech výše zmíněných druhů interpolací.

2.3 Odhad kanálu dle standardu IEEE 802.11

Pilotní signál dle standardu IEEE trvá 16 µs (platný pro rozdíl nosných kanálů 20MHz) a skládá se z 10 krátkých a 2 dlouhých tréninkových symbolů (t_1-t_{10} a T_1-T_2). Krátký tréninkový symbol je vysílán na 12 subnosných, které jsou modulovány prvky sekvence *S*:

$S_{-26,26} = \sqrt{(13/6)} \times \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0,0\}$ (2.6) [3]

Konstanta $\sqrt{(13/6)}$ je zavedena z důvodu normalizace průměrného výkonu OFDM. Signál bude generován dle rovnice

$$r_{SHORT}(t) = w_{SHORT}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} S_k \exp(j2\pi k \Delta_F t) \quad (2.7)\ [3]$$

Skutečnost, že pouze spektrální čáry $S_{-26:26}$ s indexy, které jsou násobky 4, mají nenulovou amplitudu, má za následek periodicitu $T_{FFT}/4$ = 0,8µs. Doba trvání krátkých tréninkových symbolů je tedy $T_{SHORT} = 10 \cdot 0{,}8 = 8$µs.

Dlouhý tréninkový symbol je vysílán na 53 subnosných, které jsou modulovány prvky sekvence *L*:

$L_{-26,26} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0, 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\}$ (2.8) [3]

a signál bude generován dle rovnice

$$r_{LONG}(t) = w_{TLONG}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} L_k \exp(j2\pi k \Delta_F (t - T_{G12})), \quad (2.9)\ [3]$$

kde $T_{G12} = 1{,}6\mu s$.

Pro zlepšený odhad kanálu jsou vysílány dvě periody dlouhých symbolů s celkovou dobou signálu $T_{LONG} = 1{,}6 + 2 \cdot 3{,}2 = 8$µs.

Hlavička PLCP (Physical Layer Convergence Procedure) je pak sloučením obou sekvencí:

$$r_{PREAMBLE}(t) = r_{SHORT}(t) + r_{LONG}(t - T_{SHORT}) \quad (2.10)\ [3]$$

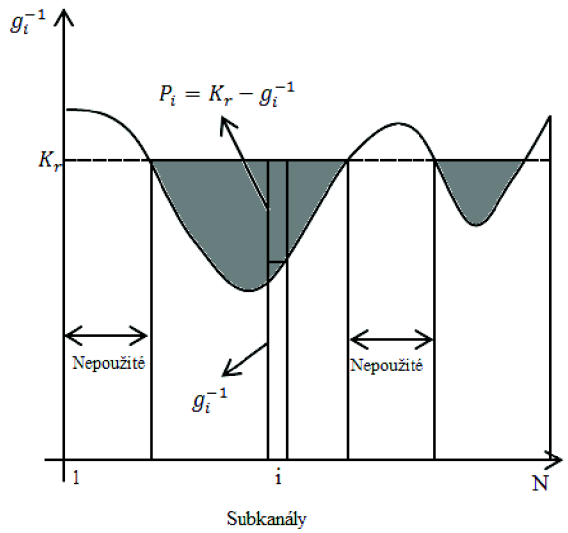
3 Adaptivní nastavování parametrů kanálů

Po odhadu všech kanálů a tedy zjištění poměru S/N pro každou subnosnou dochází k distribuci výkonu potřebného k přenosu dat, popř. bitů, a volbě modulace. Pro tyto parametry existuje více přístupů, tato studie se zaměřuje na nejpoužívanější metody.

3.1 Přidělování výkonu a bitů jednotlivým kanálům

Přidělování bitů a výkonu subkanálům je základním aspektem při tvorbě systémů s více nosnými. Algoritmy řešící toto přidělování mají za úkol vždy rozdělit celkový počet bitů a dostupný výkon mezi všechny subkanály co nejoptimálněji, aby byla zvýšena celková datová propustnost sítě při stejné kvalitě služeb.

Často používaným algoritmem pro přidělování výkonu je water-filling (viz obr. 3.1), který funguje na principu přidělování výkonu těm subnosným, které mají nejnižší útlum, popř. nejlepší poměr SNR.



Obr. 3.1: Princip algoritmu water-filling [7]

Uvažujme signál s více nosnými, kde pro každý subkanál *i* existuje přenosová funkce $R(P_i)$, která určuje, kolik bitů b_i může být vysláno za použití výkonu P_i. Tato funkce je závislá na maximální tolerovatelné pravděpodobnosti chyby a použitém modulačním schématu. Dále uvažujme inverzní funkci R^{-1} (výkonová funkce), která určuje, jaký je potřebný výkon P_i k přenosu b_i bitů. Při použití libovolné modulace QAM se dá funkce $R(P_i)$ vyjádřit jako

$$b_i = log_2(1 + P_i \cdot g_i), \qquad (3.1)\ [7]$$

kde $g_i = \frac{|H_i|^2}{N_i \cdot \Gamma}$ je poměr zisk / šum, H_i je přenosová funkce kanálu, N_i je výkon šumu a Γ je veličina SNR gap, která v dB vyjadřuje rozdíl mezi SNR nutným k dosažení dané chybovosti BER vybranou modulací a způsobem kódování a SNR daným Shannonovým vztahem.

SNR gap závisí na cílové chybovosti P_e, zisku užitého kódování γ_c a výkonovém odstupu γ_m, který vyjadřuje imunitu systému proti degradaci SNR. Čím vyšší je tento odstup, tím vyšší je výkon potřebný pro požadovanou chybovost. Pro QAM platí

$$\Gamma = \frac{1}{3} \cdot \left[Q^{-1} \frac{P_e}{4}\right]^2 \cdot \frac{\gamma_m}{\gamma_c}, \qquad (3.2)\ [7]$$

kde Q^{-1} je inverzní Q-funkce:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-y^2/2} dy. \qquad (3.3)\ [7]$$

Ze vzorce (3.1) vyplývá, že výkonová funkce je definovaná

$$P_i = \frac{2^{b_i - 1}}{g_i} \qquad (3.4)\ [7]$$

a celkový výkon P a přenosová rychlost B jsou dle [7]

$$P = \sum_{i=1}^{N} P_i \qquad (3.5a)$$

$$B = \sum_{i=1}^{N} b_i \qquad (3.5b)$$

3.1.1 Maximalizace přenosové rychlosti

Úkolem optimalizace je maximalizovat přenosovou funkci (3.1) nebo minimalizovat výkonovou funkci (3.4). Water-filling pro první možnost je vyjádřen

$$P_i = \begin{cases} K_r - g_i^{-1} & \text{pro všechny subkanály} \\ 0 & \text{jinde,} \end{cases} \qquad (3.6)\ [7]$$

kde K_r je konstanta závislá na dostupném výkonu P_{budget}.

Obr. 3.1 je vyjádřením rovnice (3.6). Spektrum může být chápáno jako nádoba a tvar dna této nádoby je určen hodnotami g_i^{-1}. Pokud bychom do této nádoby nalévali vodu (přidávali do systému výkon), stékala by do lokálních minim, tedy výkon by byl přidělen kanálům s nejmenšími ztrátami. K_r zde určuje výšku vodní hladiny; kanálům nacházejícím se pod touto hladinou je přidělován výkon, ostatní zůstávají neaktivní (viz vzorce (3.8 a,b)). Za předpokladu, že kanály jsou seřazeny sestupně, kdy $g_1 \geq g_i \geq g_N$ platí

$$K_r = \frac{P_{budget}}{M_r} + \frac{1}{M_r} \sum_{i=1}^{M_r} g_i^{-1}, \qquad (3.7)\ [7]$$

kde M_r je celkový počet použitých subkanálů dle kritérií popsaných v [7]

$$\frac{P_{budget}}{M_r} + \frac{1}{M_r} \sum_{i=1}^{M_r} g_i^{-1} \geq g_{M_r}^{-1} \qquad (3.8a)$$

$$\frac{P_{budget}}{M_r+1} + \frac{1}{M_r+1} \sum_{i=1}^{M_r+1} g_i^{-1} < g_{M_{r+1}}^{-1} \qquad (3.8b)$$

Úkolem algoritmu je tedy určit hodnotu vodní hladiny K_r a poslední z kanálů M_r, kterému bude přidělen výkon.

Optimální přidělování bitů se určí z rovnic (3.1) a (3.8) do tvaru

$$b_i = \begin{cases} \log_2(K_r g_i) & \forall i: 1 \leq i \leq M_r \\ 0 & \text{jinde,} \end{cases} \qquad (3.9)\ [7]$$

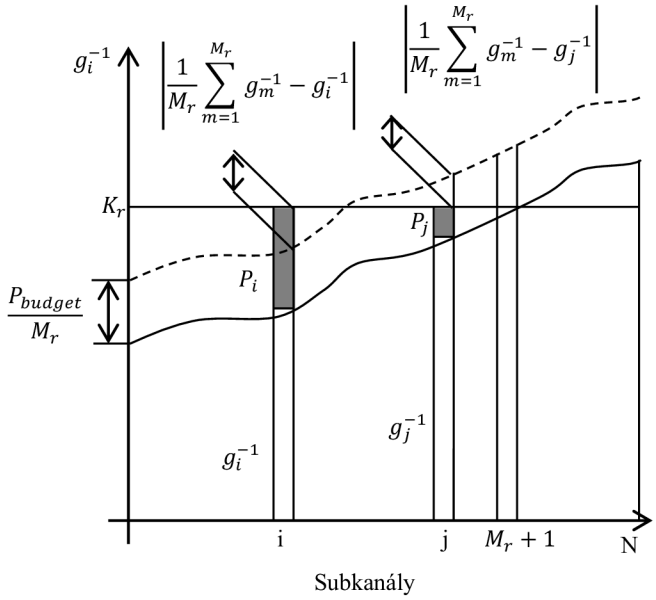
při celkové datové rychlosti

$$B = \sum_{i=1}^{N} M_r \log_2(K_r) - \sum_{i=1}^{M_r} \log_2(g_i^{-1}). \qquad (3.10)\ [7]$$

Vzorec pro optimální rozdělení výkonu je tedy

$$P_i = \frac{P_{budget}}{M_r} + \left(\frac{1}{M_r}\sum_{m=1}^{M_r} g_m^{-1} - g_i^{-1}\right), \qquad \forall i: 1 \leq i \leq M_r, \qquad (3.11)\ [7]$$

který lze vyjádřit pomocí obr. 3.2.



Obr. 3.2: Grafické vyjádření vzorce 3.11 [7]

3.1.2 Minimalizace potřebného výkonu

Analogicky k maximalizaci přenosové rychlosti lze určit minimální potřebný výkon pro přenos dat při dané bitové rychlosti. Water-filling pro tuto možnost je vyjádřen jako

$$b_i = \begin{cases} K_m - \log_2 g_i^{-1} & \text{pro všechny subkanály} \\ 0 & \text{jinde,} \end{cases} \qquad (3.12)\ [7]$$

kde K_m je konstanta závislá na cílové datové rychlosti B_{target}. Pokud jsou kanály seřazeny sestupně, lze K_m vyjádřit jako

$$K_m = \frac{B_{target}}{M_m} + \frac{1}{M_m}\sum_{i=1}^{M_m}\log_2(g_i^{-1}), \qquad (3.13)\ [7]$$

kde M_m je počet použitých subkanálů podle následujících kritérií popsaných v [7]

$$\frac{B_{target}}{M_m} + \frac{1}{M_m}\sum_{i=1}^{M_m}\log_2(g_i^{-1}) \geq \log_2\left(g_{M_m}^{-1}\right) \qquad (3.14a)$$

$$\frac{B_{target}}{M_m+1} + \frac{1}{M_m+1}\sum_{i=1}^{M_m+1}\log_2(g_i^{-1}) < \log_2\left(g_{M_m+1}^{-1}\right) \qquad (3.14b)$$

Výkon každého kanálu, kterému bylo přiděleno b_i bitů podle (3.12) je pak vyjádřen

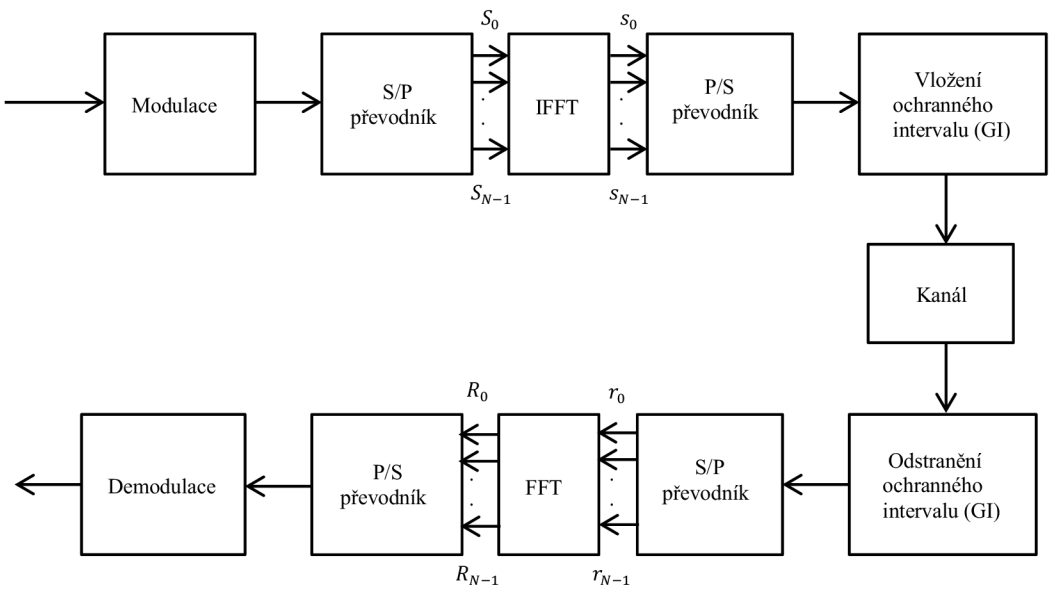
$$\begin{cases} 2^{K_m} - g_i^{-1} & \forall i: 1 \leq i\ M_m \\ 0 & \text{jinde} \end{cases} \qquad (3.15)\ [7]$$

a celkový výkon je tedy

$$P = \sum_{i=1}^{N} P_i = M_m 2^{K_m} - \sum_{i=1}^{M_m} g_i^{-1}. \qquad (3.16)\ [7]$$

3.2 Volba modulace v kanálech

Každá subnosná v OFDM může být modulována jiným modulačním schématem, přičemž záleží na konkrétních požadavcích na kanál, např. datovou rychlosti či BER. Nejčastěji jsou použity modulace BPSK, QPSK, DQPSK, 16QAM a 64QAM.



Obr. 3.3: Systémový model OFDM [8]

Z modelu systému OFDM (viz obr. 3.3) vyplývá, že modulovaný signál s N subnosnými je obohacen o cyklický prefix a je odvysílán kanálem s časově proměnnou impulzní odezvou $h(\tau,t)$ a aditivním šumem AWGN. V přijímači je odstraněn cyklický prefix a datové vzorky jsou převedeny pomocí FFT do časové oblasti, kde vystupují jako symboly R_n. Tato přijátá data R_n se dají vyjádřit jako

$$R_n = S_n \cdot H_n + n_n, \qquad (3.17)\ [8]$$

kde n_n je vzorek AWGN.

Protože je energie šumu v každé subnosné nezávislá na přenosové funkci H_n, lokální SNR v subnosné lze vyjádřit jako

$$\gamma_n = |H_n|^2 \cdot \gamma\,, \qquad (3.18)\ [8]$$

kde γ je celkový poměr SNR.

Cílem adaptivní modulace tedy je vybrat patřičnou metodu modulace pro vysílání v každé subnosné při daném SNR γ_n za účelem vhodného kompromisu mezi datovou rychlostí a celkovým BER. Přijatelná chybovost BER se liší v závislosti na ostatních parametrech systému jako kódovací poměr zabezpečení proti chybám a povaze poskytovaných služeb.

Výběr modulačního schématu probíhá tak, že se jednotlivým modulacím přiřadí proměnná M_m, kde m značí počet bitů potřebný k modulaci. Tedy pokud systém používá např. modulace BPSK, QPSK, 16-QAM a „nevysílat," pak $m \in (0,1,2,4)$. Aby byla zachována nízká složitost systému, modulační schémata se přiřazují ne každému subkanálu zvlášť, ale každému subpásmu složenému z několika sousedních subkanálů.

K výběru správného modulačního schématu slouží množství algoritmů, z nichž bude blíže rozebrán

- Algoritmus pevného práhu (Fixed threshold adaptation algorithm)
- Algoritmus odhadu BER subpásma (Subband BER estimator adaptation algorithm)
- Algoritmus konstatní datové rychlosti (Constant throughput adaptation algorithm)

3.2.1 Algoritmus pevného práhu

Pro tento algoritmus musí být kvalita kanálu konstantní pro všechny přenášené symboly. Tyto jsou pak všechny modulovány stejnou modulací vybrané v závislosti na SNR. Úrovně přepnutí na modulační schéma (viz tab. 3.1) s vyšším počtem BPS (Bits Per Symbol) byly určeny pro OFDM na základě výpočtů v [9].

Tab. 3.1: Úrovně SNR pro přepnutí na další typ modulace [8]

	I_0	I_1	I_2	I_4
Zvuk [dB]	-∞	3,31	6,48	11,61
Data [dB]	-∞	7,98	10,42	16,76

Pokud je tedy SNR subkanálu pro datový přenos mezi hodnotami I_0 a I_1, používá se modulace BPSK, pokud je ale překročena úroveň I_1 a je nižší než I_2, užije se modulace QPSK.

3.2.2 Algoritmus odhadu BER subpásma

Toto schéma bere v úvahu neustále se měnící hodnoty γ_j (SNR) na N_s subnosných v j-tém subpásmu. Pro jeho použití je potřeba vypočítat předpokládanou celkovou pravděpodobnost chyby pro všechny dostupné modulace M_n v každém subpásmu

$$\overline{p_e}(n) = \frac{1}{N_s}\sum_j p_e(\gamma_j, M_n). \qquad (3.19)\ [8]$$

V každém subpásmu je pak vybráno modulační schéma, které zajišťuje nejvyšší datovou rychlost za předpokladu, že jeho BER je nižší, než daný práh.

3.2.3 Algoritmus konstantní datové rychlosti

Toto modulační schéma využívá frekvenční selektivity kanálu a zaručuje konstantní datovou rychlost, což je velmi důležité při audio a video přenosech, kdy je vysílání velmi citlivé na zpoždění. Pro každé subpásmo je vypočítaná „cena“

$$c_{n,s} = \frac{e_{n,s+1} - e_{n,s}}{b_{n,s+1} - b_{n,s}} \qquad (3.20)\ [8]$$

pro všechny typy modulace označené s, kromě modulace s nejvyšší hodnotou BPS.

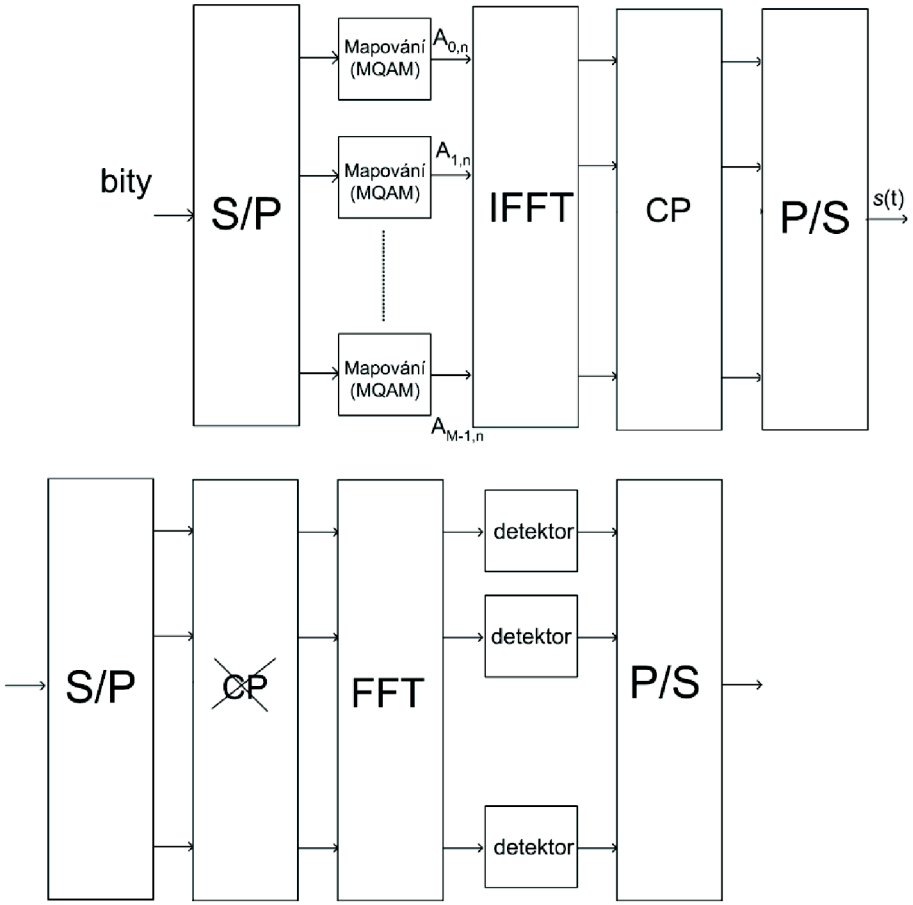
Přidělování subpásem funguje na principu použití modulačního schématu s nejnižší cenou přenosu $c_{n,s}$, která závisí na předpokládaném počtu bitových chyb $e_{n,s}$ a na bitové rychlosti $b_{n,s}$ v každém subpásmu n pro každou možnou modulaci s.

4 Simulace v prostředí MATLAB

Pro simulaci byl vytyčen základní cíl vytvořit funkční algoritmus pro alokaci bitů na jednotlivé subnosné v závislosti na jejich přenosu. Tyto bity jsou pak modulovány vnitřními modulacemi a mapovány na vstup IFFT. Použitá vnitřní modulační schémata jsou BPSK, QPSK, 16QAM a 64QAM. Pro simulaci bylo rovněž použito několik zjednodušení:

- Přenosový kanál a úrovně přenosu na jednotlivých subnosných jsou časově invariantní
- Omezení vysílání na 1 OFDM symbol na subnosnou kvůli lepší názornosti algoritmu
- Předpokládáme, že přijímač zná použitou vnitřní modulaci na každé subnosné bez přenosu doplňkových informací

Blokové schéma simulovaného modelu je shodné s blokovým schématem klasického OFDM:



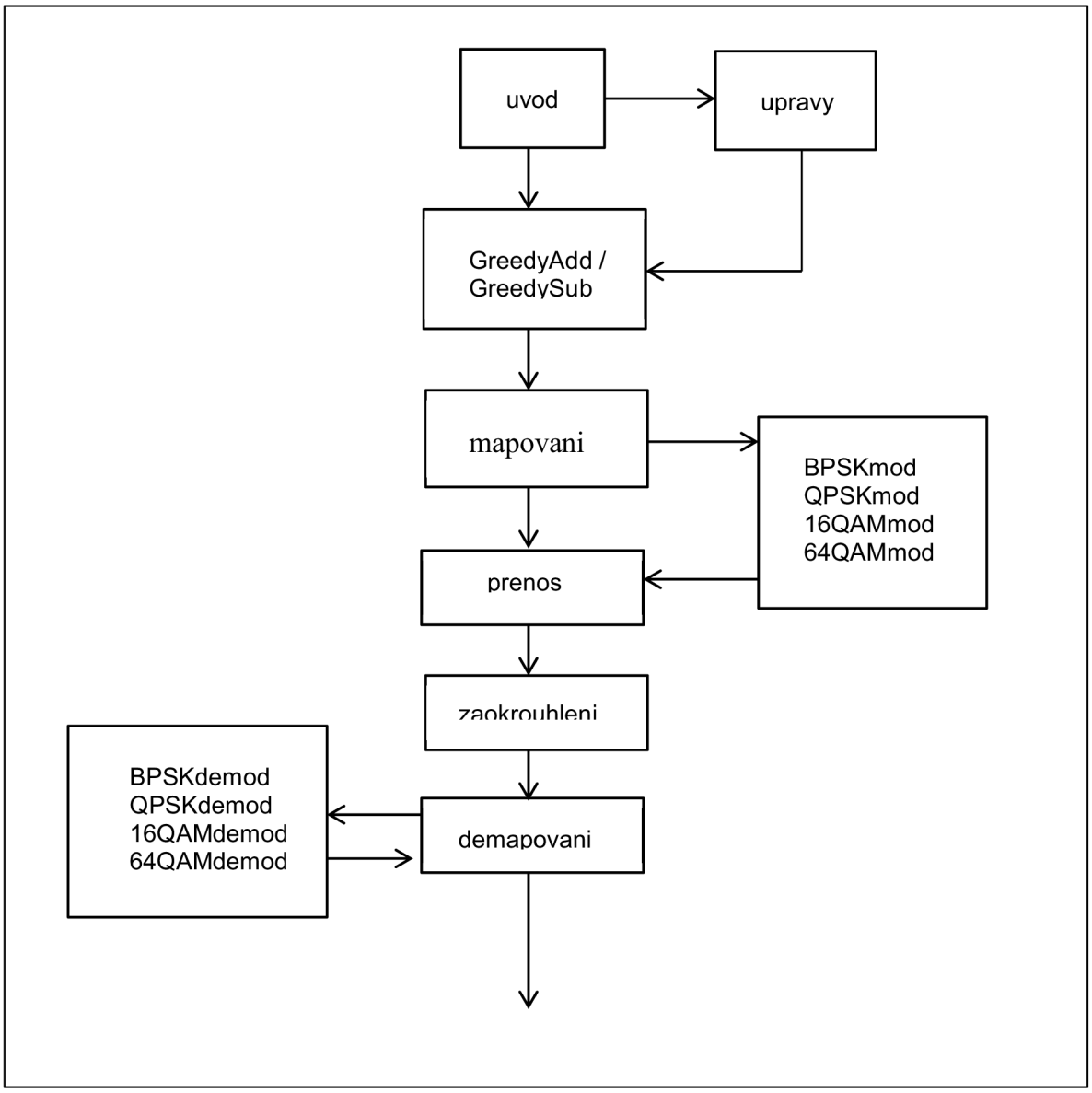
Obr. 4.1: Blokové schéma simulovaného modelu [10]

Pro simulaci systému v prostředí MATLAB byly použity parametry v tab. 1. Parametry vysílače byly přejaty ze standardu IEEE 802.11n[15] a parametry přenosového kanálu s Riceovým šířením ze standardu ITU-R M.1225[16]. K-faktor byl zvolen jako aritmetický průměr výsledků měření v [13]. Maximální Dopplerův posun byl zvolen 0 Hz, tedy při stacionárním vysílači i přijímači.

Tab. 4.1: Parametry systému WLAN použité v simulovaném modelu

Parametr	Hodnota
Vzorkovací kmitočet f_{vz}	20 MHz
Počet datových subnosných	52
Počet vložených nulových subnosných	11 + 1 DC
Délka cyklického prefixu (CP)	1/4
K-faktor Riceova kanálu	7,66
Maximální Dopplerův posun	0 Hz
Vektor útlumu pro zpoždění o [0; 50e-9; 110e-9] s	[0; -3; -10] dB

V simulaci je vytvořeno množství funkcí, proto pro větší přehlednost viz vývojový diagram na obr. 4.2. Pro stručný popis funkcí viz tab. 4.2.



Obr. 4.2: Vývojový diagram simulace v prostředí MATLAB

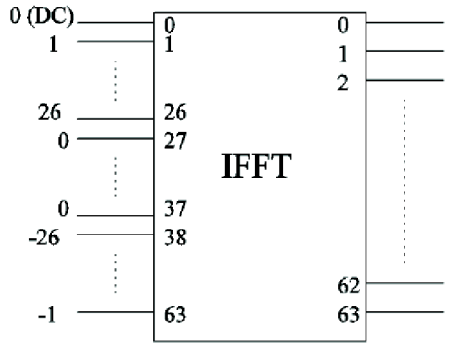
Tab. 4.2: Popis vytvořených funkcí pro simulaci v prostředí MATLAB

Název funkce	Popis	Důležité výstupní proměnné
uvod	generace pilotního signálu, odhad přenosu kanálu	*zisk*
upravy	odstranění CP, následná FFT a odstranění nul	*obnoveny, konecny*
GreedyAdd	Greedy algoritmus pro přičítání bitů	*pocitadlo, P*
GreedySub	Greedy algoritmus pro odečítání bitů	*pocitadlo, P*
mapovani	alokace bitů na subnosné, volba modulace	*ofdm_symbols*
BPSKmod	převod bitů na symboly dané modulace	-
QPSKmod		
QAMmod16		
QAMmod64		
BPSKdemod	převod symbolů na bity	-
QPSKdemod		
QAMdemod16		
QAMdemod64		
prenos	vysílání a příjem OFDM symbolů přenosovým kanálem	*received*
zaokrouhleni	zaokrouhlení reálné a imaginární složky přijatých symbolů	*rounded*
demapovani	demodulace, převod bitů z nosných na sériový datový tok	*mapped*

4.1 Úvodní odhad kanálu

K získání úvodního odhadu přenosu na jednotlivých subnosných je využit místo hřebenového typu pilotního signálu blokový typ (viz obrázky 2.1, 2.2). Odhad kanálu je tak získán z prvního symbolu na všech nosných a 4 nosné, které jsou v systému WLAN použity jako pilotní a tudíž nepřenášejí užitečná data, mohou být v tomto případě užity k přenosu užitečných dat. Funkce *uvod* respektuje tyto parametry a je napsaná dle blokového schématu na obr. 4.1. Pomocí této funkce je vygenerován 1 QPSK symbol na každou nosnou, tato data jsou paralelním datovým tokem o 52 nosných. Odhad pomocí QPSK symbolů je použit kvůli zamezení ovlivnění nulovými nosnými, které jsou přidány dále.

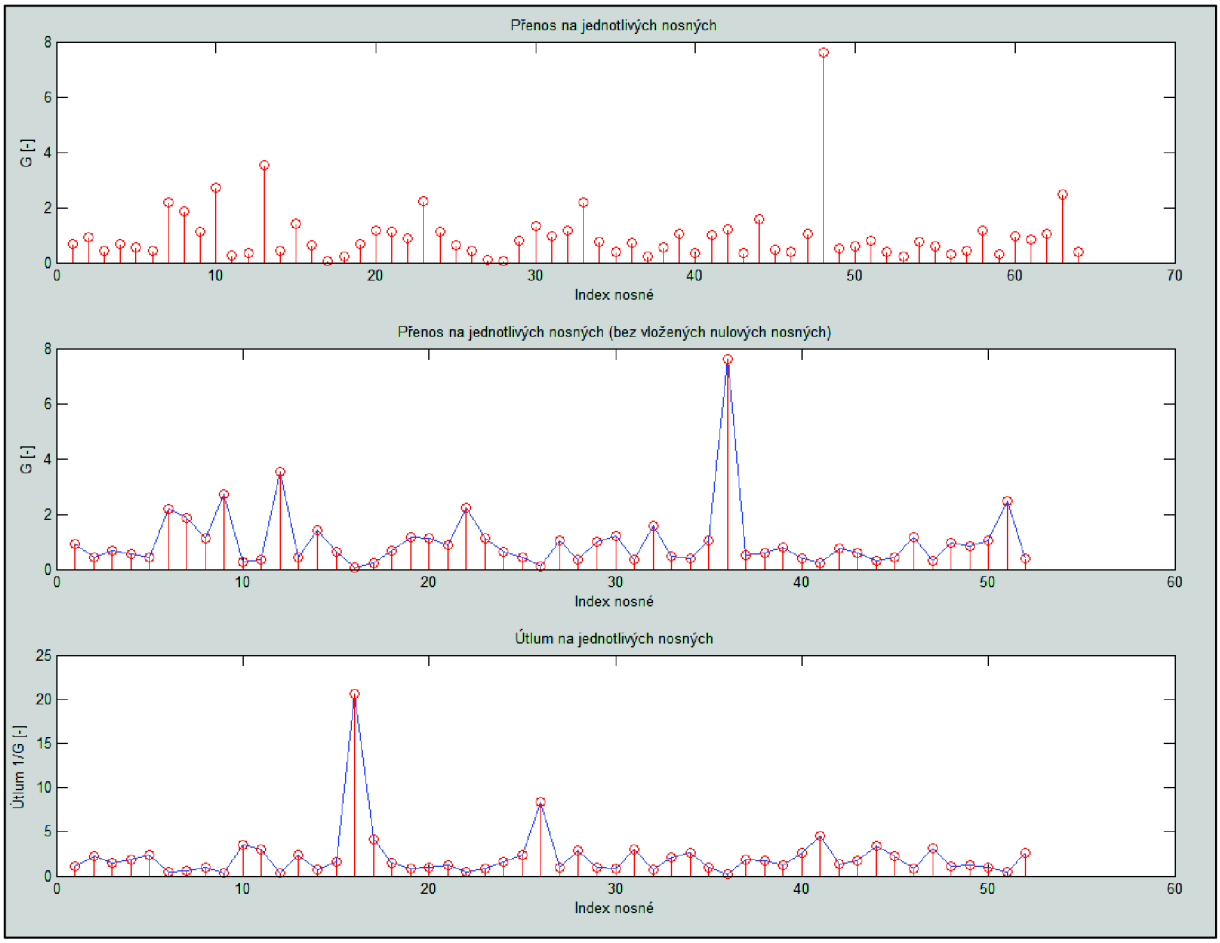
K nosným nesoucím užitečná data je před vstupem do IFFT přidáno celkem 12 nulových nosných - nulová DC složka (nosná 0 na obr. 4.3) a 11 nosných v okolí $f_{vz}/2$ (nosné 27-37 na obr. 4.3). Tento mechanismus je zde kvůli oddělení spektra od sousedních kanálů.



Obr. 4.3: Nosné vstupující do IFFT [10]

Po průchodu IFFT je k signálu přidán cyklický prefix o délce 1/4 a vzniklý datový tok je převeden z paralelního uspořádání do sériového. Protože předpokládáme Riceův přenosový kanál, je CP nutný kvůli snížení vlivu vícecestného šíření. Navíc je k signálu přidán AWGN se SNR 20dB. Parametry Riceova kanálu byly zvoleny dle tabulky 4.1.

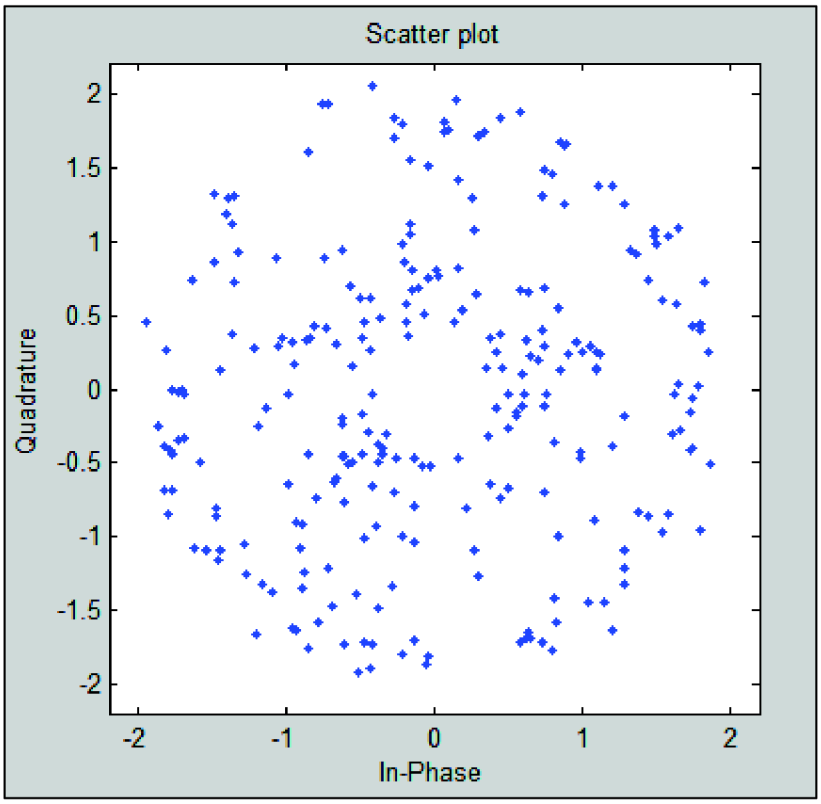
V přijímači je pomocí funkce *upravy* nejdříve odstraněn CP, poté je signál převeden zpět na paralelní datový tok. Po FFT jsou odstraněny i nulové nosné a získaný paralelní signál je poté použit k výpočtu přenosu na jednotlivých subnosných. Pro výpočet se používá všech 52 přijatých symbolů, tedy pilotní signál. Přenos kanálu je spočten vydělením vzorek po vzorku (tzv. element-wise dělením) výstupní matice získané v přijímači z FFT vstupní maticí vstupující do IFFT.



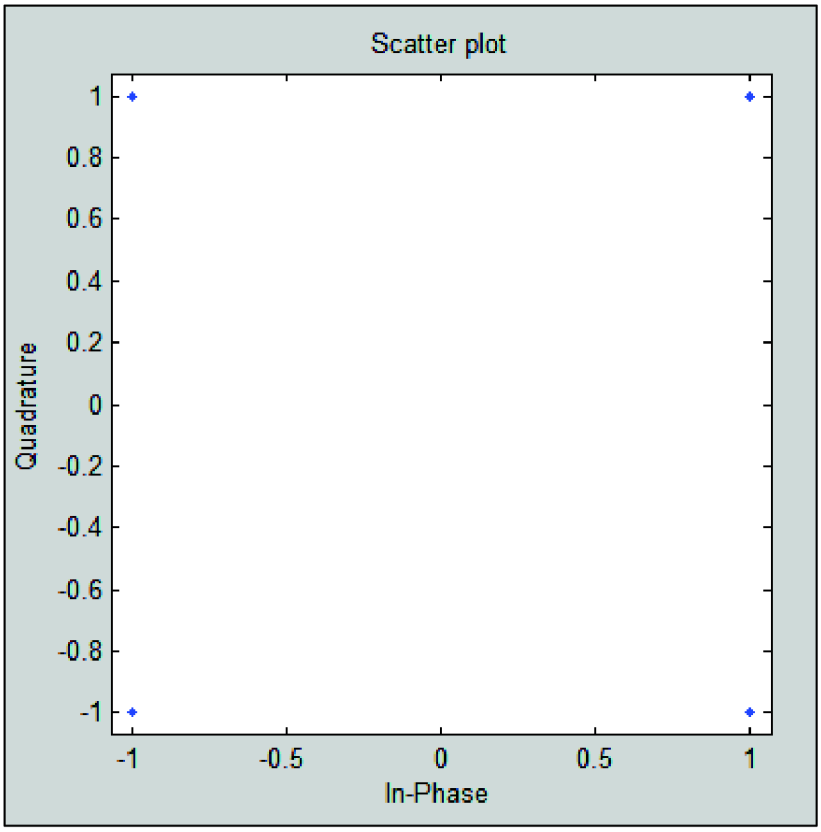
Obr. 4.4: Úvodní odhad přenosu a útlumu na jednotlivých subnosných

Útlum na jednotlivých subnosných je vypočítán inverzí přenosu (G^{-1}). Na nosné s nejvyšším přenosem, tedy minimálním útlumem, pak bude alokováno nejvíce bitů a bude možno použít modulace s vyšším počtem stavů.

Při znalosti přenosu kanálu je možno zavést ekvalizaci v podobě zero-forcing ekvalizéru, jenž je rovný inverzi přenosu *1/G* a napravuje amplitudové i fázové odchylky přijatého signálu způsobené přenosovým kanálem. Ekvalizační maticí vynásobíme vzorek po vzorku každý přijatý symbol, díky čemuž získáme původní pilotní signál v nezměněné podobě. Tato metoda ekvalizace je použita i pro další příjem. Vliv ekvalizace na přijaté symboly lze porovnat z obrázků 4.5 a 4.6.



Obr. 4.5: Konstelační diagram přijatého signálu bez ekvalizace



Obr. 4.6: Konstelační diagram přijatého signálu po použití zero-forcing ekvalizéru

4.2 Algoritmus Greedy

Po úvodním odhadu kanálu je nutné zajistit, aby na subnosné s nejvyšším přenosem bylo alokováno nejvíce bitů. Toto řeší algoritmus Greedy, který využívá principu water-filling popisovaného v kap. 3.1.

Základem tohoto algoritmu je určení změny výkonu nutné k vyslání dalšího bitu na dané nosné. Tento výkon je možno určit dvěma způsoby, podle druhu Greedy algoritmu. Prvním způsobem je přičítání výkonu, tedy za každý bit vyslaný na nosné je k celkovému výkonu na nosné přičtena hodnota změny výkonu. Druhým přístupem je odečítání výkonu tak, že se určí maximální možný počet bitů na všech nosných (např. při nejvyšší modulaci 64QAM je to 6 bitů na nosnou) a poté je vypočtena úspora výkonu při odebrání 1 bitu na každé nosné.

Bity jsou postupně přiřazovány na nosné s nejnižší změnou výkonu, resp. Odebírány z nosných s nejvyšší úsporou výkonu. Po přidání, resp. odebrání bitu je pro danou nosnou spočtena nová změna výkonu pro přidání, resp. odebrání dalšího bitu. Protože v modelu není určen výkon na jednotlivých nosných v jednotkách W, je změna výkonu jen vyjádřením míry změny v poměru k ostatním nosným.

Změna výkonu při metodě přičítání (4.1a) a odečítání (4.1b) je určena podle vzorců ze [7]

$$\Delta P_i^+(b_i) = \frac{2^{b_i}}{g_i}, \tag{4.1a}$$

$$\Delta P_i^-(b_i) = \frac{2^{(b_i-1)}}{g_i}, \tag{4.1b}$$

kde b_i je počet bitů již přiřazených dané nosné a g_i je přenos na dané nosné.

Model umožňuje simulovat pouze metodu přičítání, nebo pouze metodu odečítání, nebo srovnání obou metod.

4.2.1 Metoda přičítání (funkce GreedyAdd)

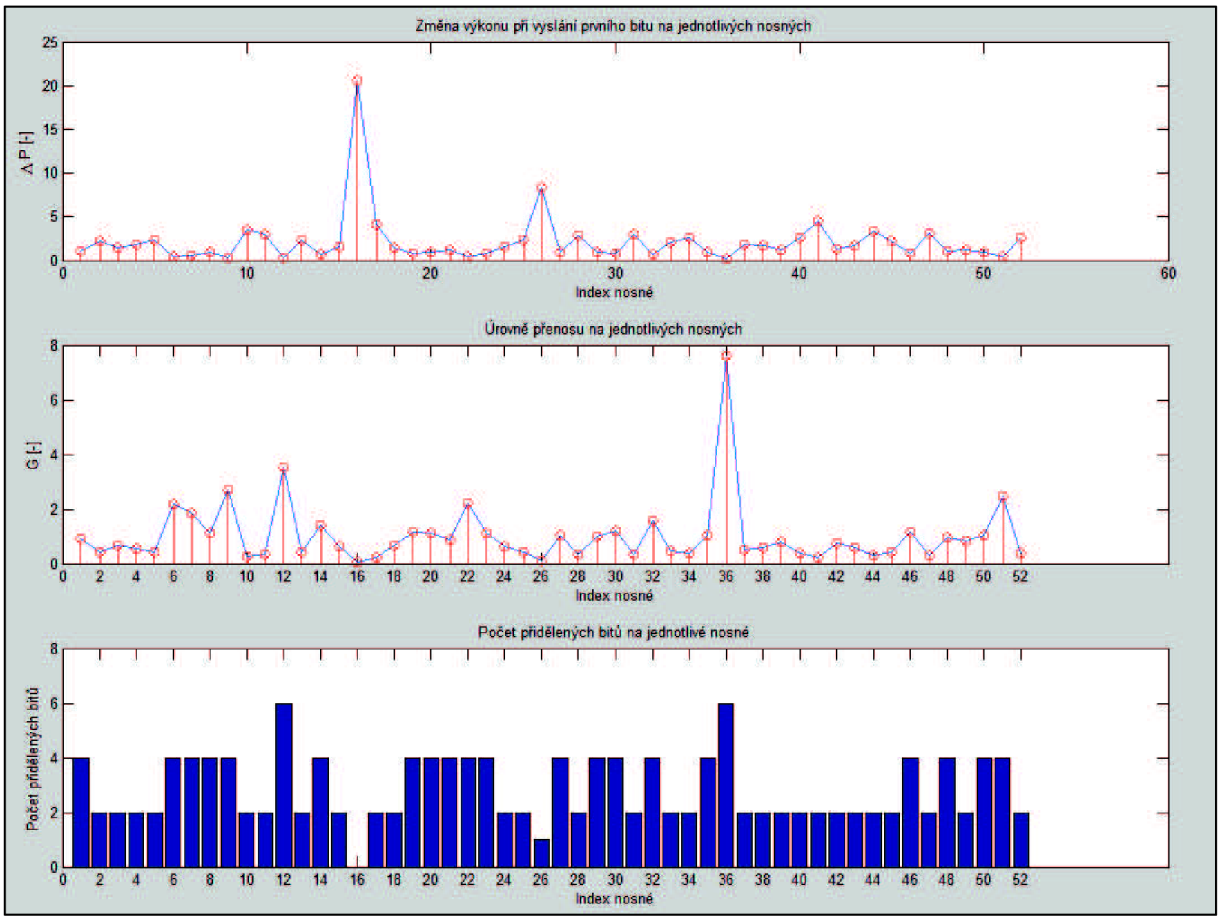
Podle rovnice 4.1a je určena změna výkonu na všech nosných při 0 vysílaných bitech. Tato úvodní změna výkonu je rovna převrácené hodnotě přenosu. Je vygenerován nový datový tok o maximální délce *počet_nosných*maximálně_bitů_na_nosnou*. V případě 52 nosných je tedy maximální délka datového toku 312 bitů.

Pomocí cyklu *while* trvajícího tolikrát, kolik bitů bude později alokováno, a funkce *find* je vyhledána nosná s minimální změnou výkonu a vektorová proměnná *pocitadlo* se pro index dané nosné zvýší o 1. Do algoritmu je zakomponováno i omezení pro maximální počet 6 bitů na subnosnou, které je řešeno nastavením nenumerické hodnoty změny výkonu (NaN – Not a Number) na dané nosné a vyhledáním nové nosné s nejnižší změnou výkonu.

Druhé omezení se týká počtu bitů na subnosné, pro které není implementována vnitřní modulace, tedy počty bitů 3 a 5. V modelu jsou totiž použity jen modulace BPSK, QPSK,

16QAM a 64QAM, tedy pro počty bitů 1, 2, 4 a 6. Pokud by se tedy systém pokusil přiřadit na subnosnou počet bitů 3 nebo 5, přiřadí 4, resp. 6. V případě, že je hodnota proměnné *pocitadlo* pro subnosnou 2 nebo 4, je další změna výkonu vypočítána pro přidání dvou bitů.

Zobrazení konečného počtu bitů k alokaci by mělo přibližně kopírovat tvar určený zobrazením hodnot přenosu kanálu. Obr. 4.7 zobrazuje, že k tomuto jevu opravdu dochází, ačkoliv je viditelné ovlivnění nepoužitím modulací 8PSK a 32QAM.

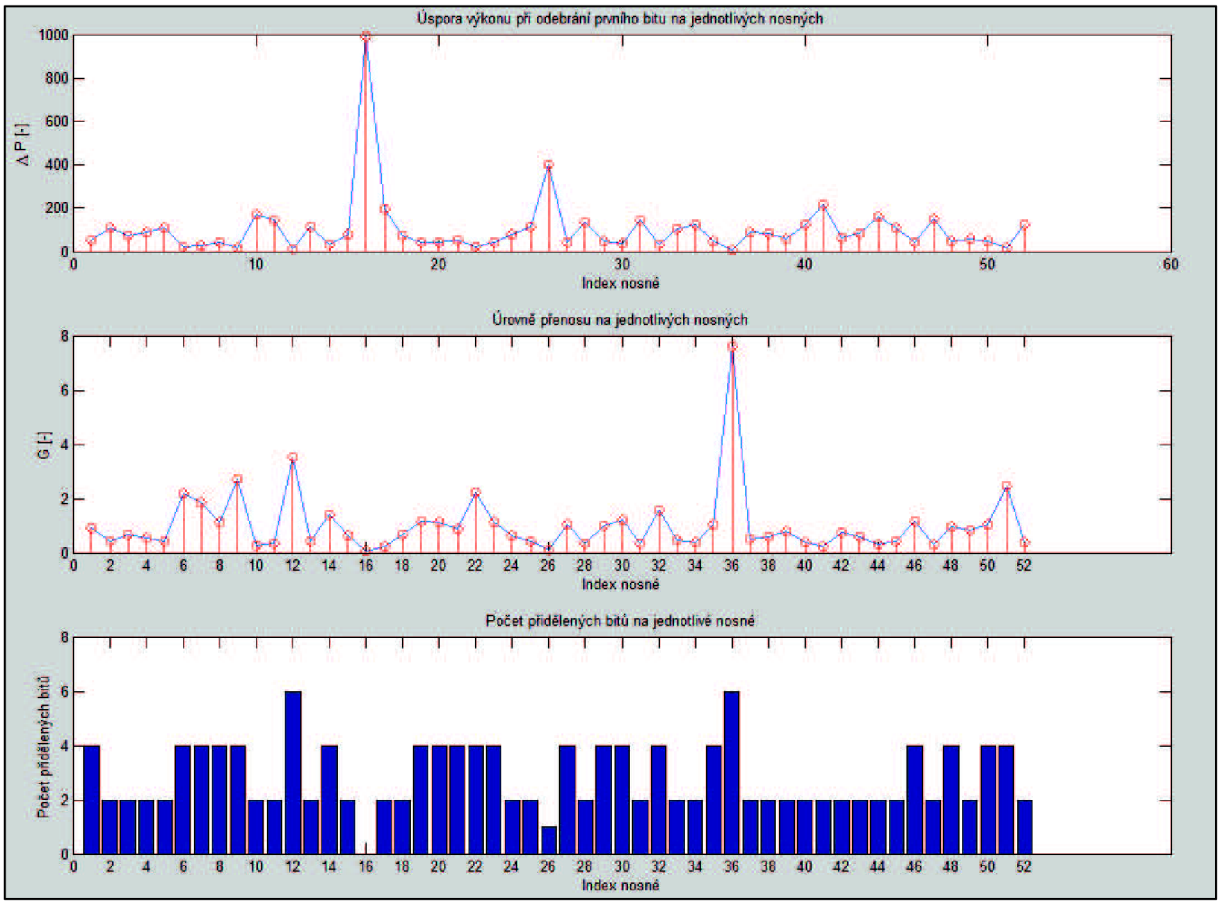


Obr. 4.7: Rozdělení bitů na subnosné při metodě přičítání a počtu vysílaných bitů 150

4.2.2 Metoda odečítání (funkce GreedySub)

Metoda odečítání pracuje na podobném principu jako metoda přičítání, s tím rozdílem, že úvodní změna výkonu je vyjádřením úspory výkonu (dle vzorce 4.1b) pro maximální počet bitů na subnosné, tedy 6, a ubráním prvního bitu z nosné s maximální úsporou výkonu. Proměnná *pocitadlo* má tedy implicitně na všech nosných nastavenu hodnotu 6. Protože ale model nepodporuje modulaci 32QAM, nosné s prvním odebraným bitem je odňat i druhý bit. Rovněž při odebrání bitu z počtu 4 na 3 jsou odebrány 2 bity, protože model nepodporuje modulaci 8PSK a dojde tedy ke změně počtu bitů ze 4 na 2.

Analogicky k omezení maximálního počtu bitů na subnosnou u metody přičítání je u metody odečítání zavedeno omezení minimálního počtu bitů na 0. Mechanismus je stejný, jako u metody přičítání, tedy při zjištění nulového počtu bitů na nosné se změna výkonu nastaví na nenumerickou hodnotu a vyhledá se další nosná s nejvyšší úsporou výkonu.



Obr. 4.8: Rozdělení bitů na subnosné při metodě odečítání a počtu vysílaných bitů 150

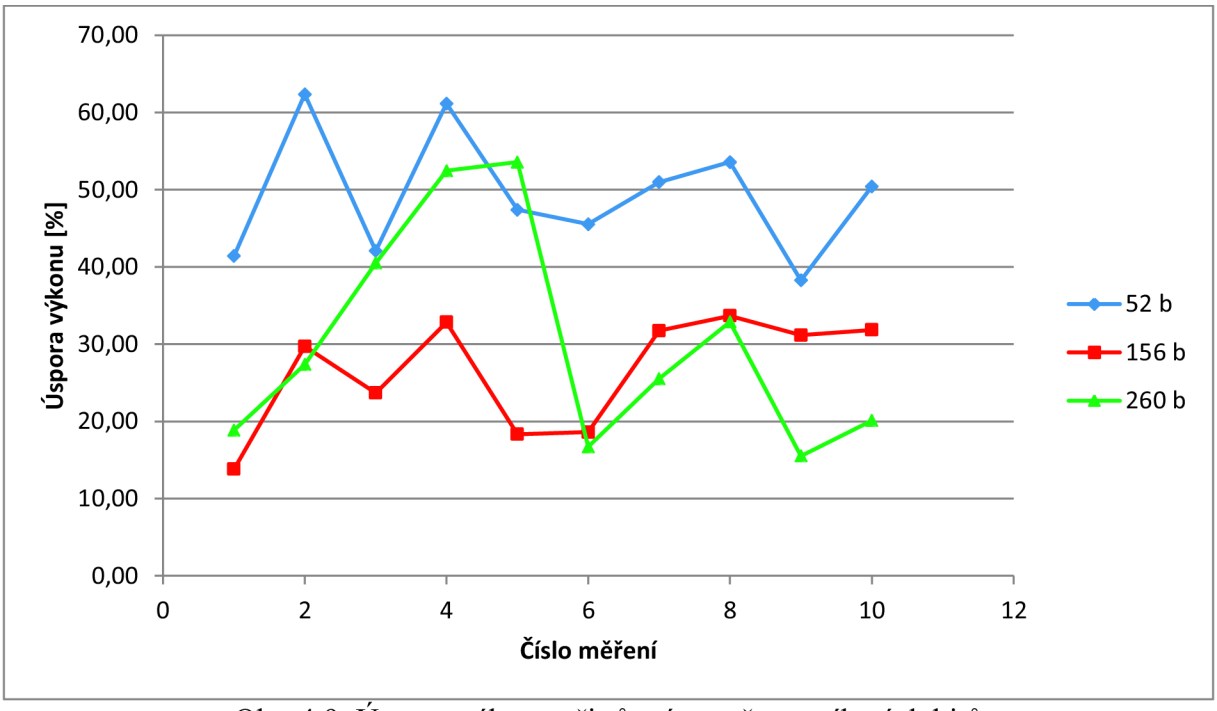
Z obrázků 4.7 a 4.8 lze vidět, že počet přidaných bitů na jednotlivé nosné je u obou metod shodný. O žádné z metod nelze říci, že by byla přesnější než druhá, ale díky lehce jednoduššímu algoritmu u metody přičítání je tato metoda vhodnější k implementaci do reálných rádií.

4.3 Alokace výkonu pro metodu přičítání

Během výpočtů změny výkonu při přidání každého bitu na subnosnou je změna ΔP připočtena ke stávajícímu potřebnému výkonu na subnosné. Po přidělení všech bitů je pomocí příkazu *sum* sečten potřebný výkon na všech subnosných a tímto součtem je vydělen přidělený výkon, který je už závislý na typu použitého vysílače, do kterého by byl tento algoritmus implementován. Zpětným vynásobením takto vzniklé konstanty jednotlivými výkony na subnosných je možno získat bezezbytkové rozdělení nastaveného výkonu, které je zobrazeno v proměnné *pridelenyvykon*.

Úsporu výkonu při využití adaptivního OFDM můžeme vypočítat na základě srovnání sumy výkonu pro jednotlivé subnosné po použití Greedy algoritmu se sumou stejně vypočítaného potřebného výkonu (podle rovnice 4.1a), ale pro stejný počet bitů na všech nosných. Na obrázku 4.9 lze vidět srovnání pro 52, 156 a 260 vysílaných bitů, což by bez použití Greedy algoritmu znamenalo 1, 3 a 5 bitů na každé nosné. Při nižším počtu vysílaných bitů šetří Greedy algoritmus jednoznačně více výkonu, kdy se minimální úspora pohybuje kolem 40%. I při vyšším počtu bitů je ale úspora pokaždé vyšší než 10%, mnohdy velmi

výrazně. Každé měření bylo zaznamenáno po vygenerování nových hodnot přenosu kanálu, následném použití algoritmu a výpočtu úspory výkonu.

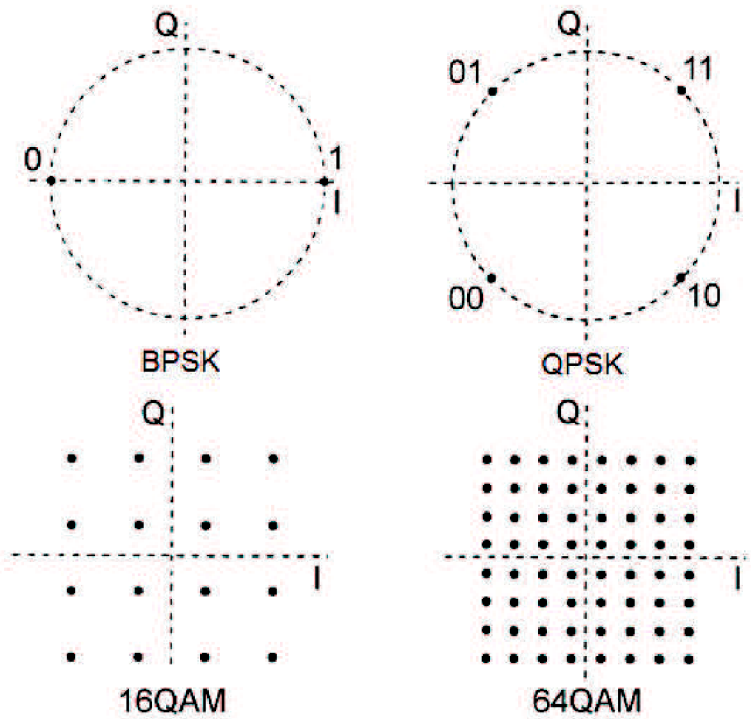


Obr. 4.9: Úspora výkonu při různém počtu vysílaných bitů

4.4 Alokace bitů a vnitřní modulace

Po přidělení počtu bitů na jednotlivé nosné (tuto informaci nese proměnná *pocitadlo*) je nutné alokovat bity datového toku. Přidělování bitů je řešeno funkcí *mapovani* tak, že každé bitové slovo o délce 1/2/4/6 bitů je modulováno příslušnou modulací vybranou pomocí příkazu *switch* a na každé nosné je vytvořen 1 OFDM symbol. Příkaz *Switch* vybírá mezi čtveřicí modulačních funkcí *BPSKmod*, *QPSKmod*, *QAMmod16* a *QAMmod64*, které fungují na principu několikanásobného použití rozhodovacího členu *If*. U modulací 16QAM a 64QAM je použito vyhledávání symbolu pomocí rozdělení bitového slova na poloviny a prohledávání konstelačních diagramů po řádcích (pro první polovinu bitového slova) a po sloupcích (pro druhou polovinu bitového slova).

Např. pro modulaci 16QAM a bitové slovo 0111 je první polovina 01 a druhá polovina 11. Pro první polovinu je vyhledán sloupec, ve kterém je shodná první polovina bitového slova, v tomto případě hodnota -1, a pro druhou polovinu je vyhledán řádek se shodnou druhou polovinou, v tomto případě hodnota +j. OFDM symbol pro bitové slovo 0111 při použití modulace 16QAM je tedy -1+j. Konstelační diagramy vnitřních modulací jsou zobrazeny na obr. 4.10.



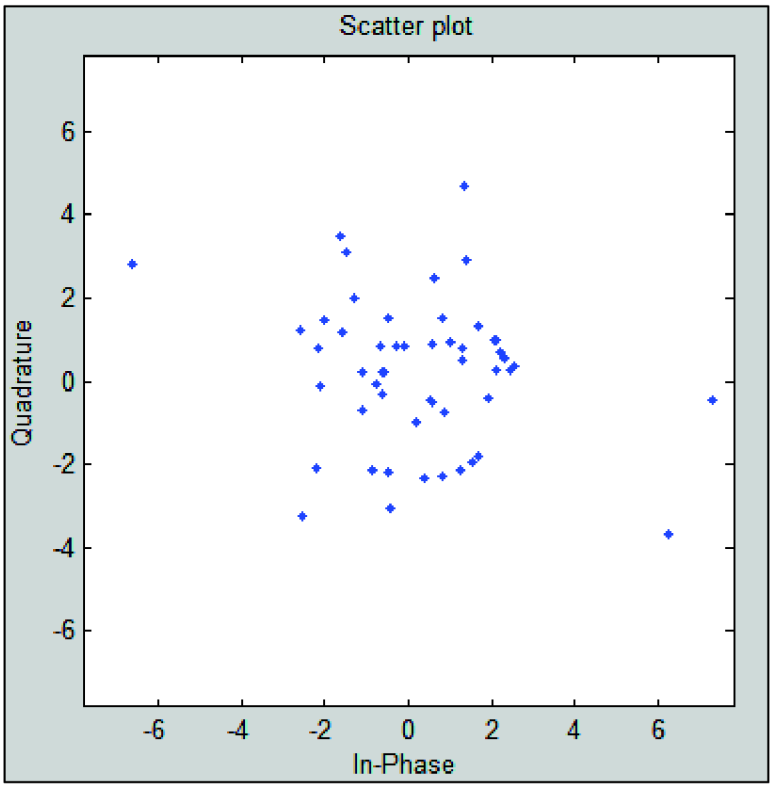
Obr. 4.10: Konstelační diagramy modulací použitých v modelu [18]

4.5 Vysílání a příjem OFDM symbolů, průchod kanálem

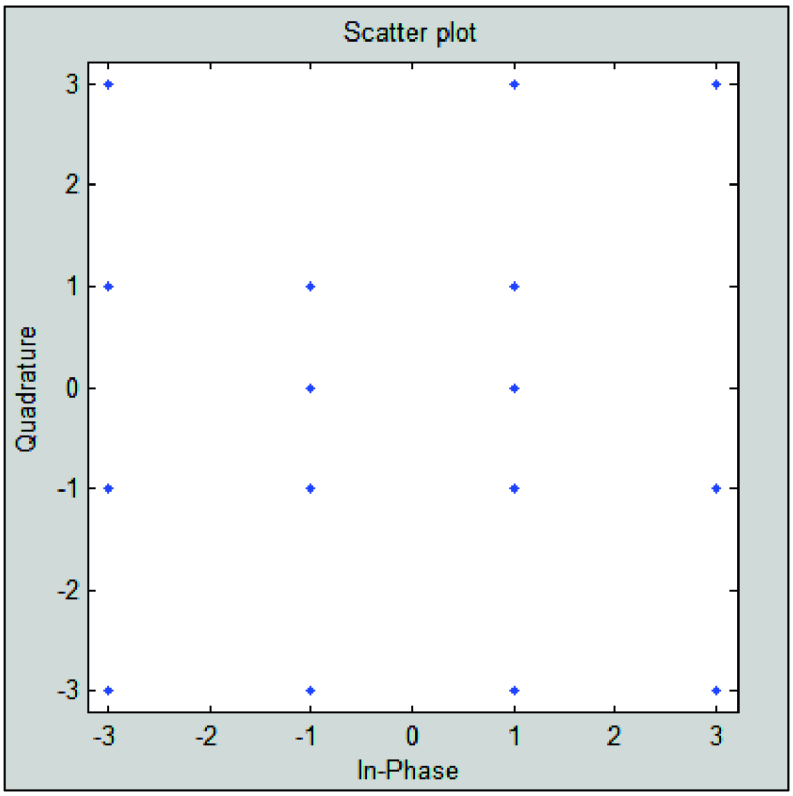
Úpravy signálu před vysíláním ve funkci *prenos* jsou totožné, jako při úvodním odhadu přenosu (viz kap. 4.1). Jediným rozdílem je nepoužití pilotního signálu, protože je předpokládán časově neměnný přenos na subnosných a tudíž není potřeba ho znovu odhadovat. Přenosový kanál je totožný jako při úvodním odhadu přenosu – v kanále s Riceovým šířením je k signálu přidán AWGN s poměrem SNR 20dB. Úpravy signálu v přijímači (funkce *upravy*) zahrnují opět převod sériového datového toku na paralelní, použití Fourierovy transformace a odstranění CP a následně i nulových nosných. Znovu je použit zero-forcing ekvalizér.

Získané OFDM symboly na jednotlivých nosných jsou upraveny funkcí *zaokrouhleni*, která nejdříve rozdělí všechny konstelační diagramy úsečkami určenými aritmetickými průměry mezi každými dvěma body určujícími stav dané modulace a pak zaokrouhlí získaný symbol na hodnotu ve středu čtverce ohraničeného úsečkami. Krajní hodnoty konstelačních diagramů pak zahrnují všechny symboly, které mají hodnotu větší (resp. menší) než je maximum (resp. minimum) na osách.

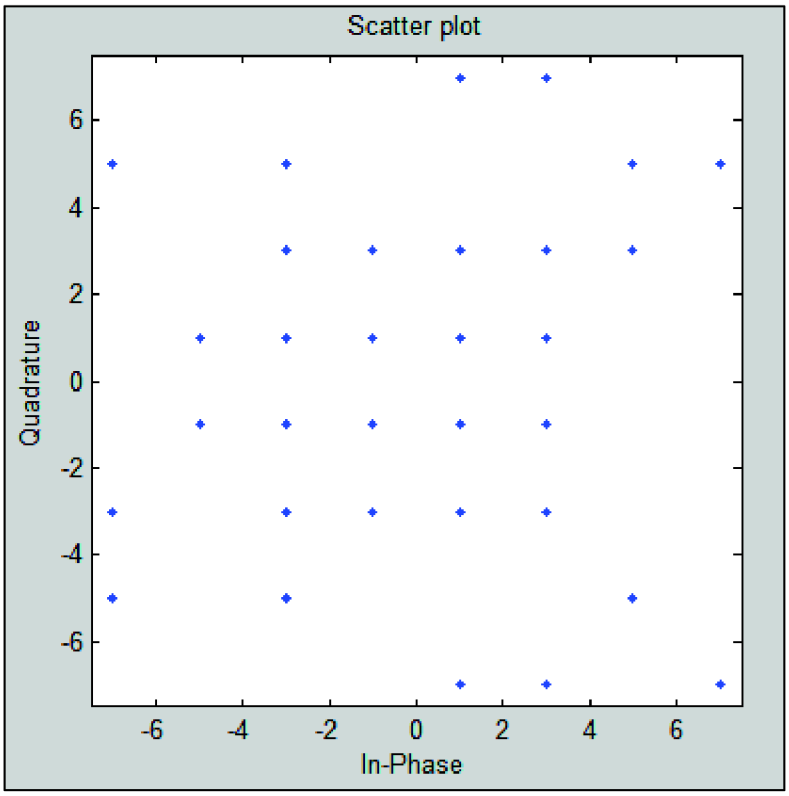
OFDM symboly jsou nakonec demodulovány zpět na bitová slova funkcí *demapovani* – symbol je rozdělen na reálnou a imaginární část a k nim jsou vyhledávány příslušné bity (pro reálnou část symbolu první polovina a pro imaginární část druhá polovina bitové sekvence). Pokud nebyl na nosné přenášen žádný bit, je nosné přidělena hodnota NaN.



Obr. 4.11: Konstelační diagram přijatého signálu bez ekvalizace



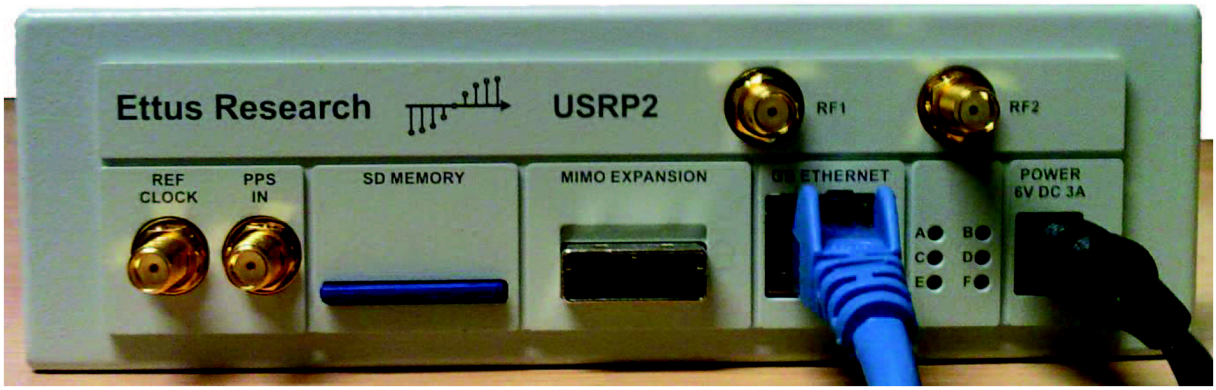
Obr. 4.12: Konstelační diagram přijatého signálu po ekvalizaci (pro 156 vysílaných bitů)



Obr. 4.13: Konstelační diagram přijatého signálu po ekvalizaci (pro 260 vysílaných bitů)

5 Implementace do USRP2 v prostředí Simulink

K implementaci výše popsaného kódu bylo využito univerzální softwarové rádio USRP2 (Universal Software Radio Peripheral) společnosti Ettus Research. Toto rádio obsahuje základní desku - motherboard, ke které lze připojit dceřinou desku – daughterboard zajišťující vysílání a příjem v požadovaném kmitočtovém pásmu. Pro účely této práce byl zvolen daughterboard RFX2400, který umožňuje vysílat i přijímat signál v pásmu ISM. Od verze prostředí MATLAB 2010b navíc Simulink obsahuje blockset pro práci s tímto hardwarem. Tento blockset automaticky programuje FPGA v USRP2 a tím umožňuje jednoduchou práci s tímto rádiem.



Obr. 5.1: USRP2

Tab. 5.1: Základní parametry motherboardu USRP2 [12]

Parametr	Hodnota
Počet A/D převodníků	2
Počet bitů A/D převodníků	14
Vzorkovací frekvence A/D převodníků	100 MS/s
Počet D/A převodníků	2
Počet bitů D/A převodníků	16
Vzorkovací frekvence D/A převodníků	400 MS/s
Maximální šířka pásma vysílaného/přijímaného signálu	50 MHz

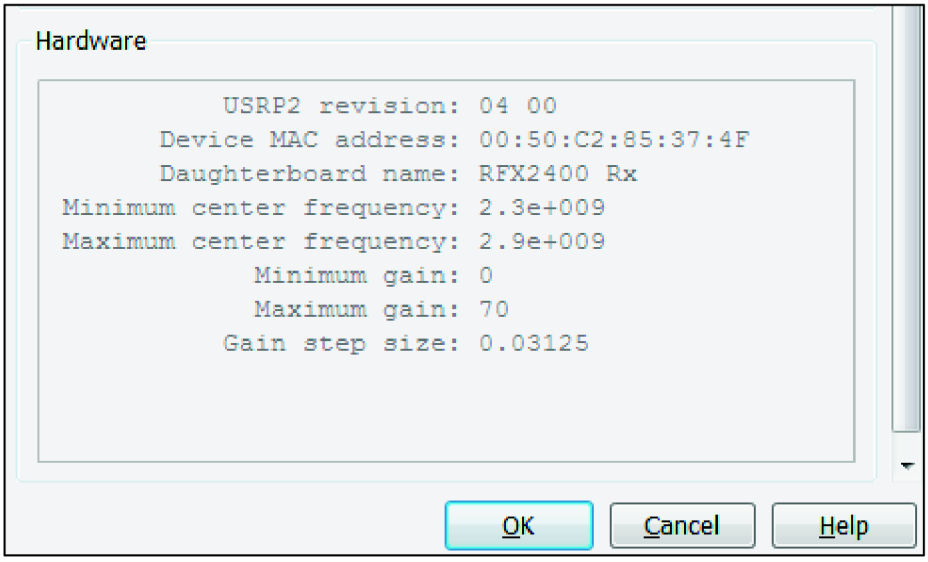
Tab. 5.2: Základní parametry daughterboardu RFX2400 [12]

Parametr	Hodnota
Frekvenční rozsah	2,3 - 2,9 GHz
Typický vysílací výkon	50 mW
Volitelný zisk přijímače	0 - 70 dB

Pro správný chod USRP2 v prostředí Simulink je nutno:

- nahrát na paměťovou SD kartu USRP2 firmware pro zvolený daughterboard
- připojit USRP2 síťovým kabelem ke gigabitové ethernetové lince
- nastavit IP adresu síťové karty v počítači na 192.168.10.1 a masku na 255.255.255.0
- nastavit IP adresu bloku USRP2 vysílače/přijímače v prostředí Simulink na 192.168.10.255
- v případě použití dvou USRP2 je potřeba nastavit druhou síťovou kartu na IP 192.168.20.1, masku na 255.255.255.0 a IP druhého rádia v prostředí Simulink na 192.168.20.255

Po těchto procedurách by ve spodní části parametrů modelu vysílače/přijímače USRP2 v prostředí Simulink měly být zobrazeny parametry připojeného rádia. Návod pro nastavení počítače i pro nahrání firmwaru lze nalézt v [19] a [20].



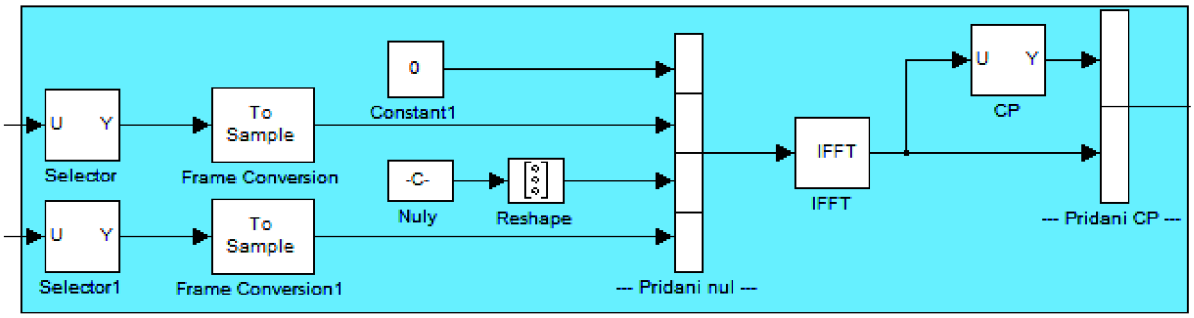
Obr. 5.2: Parametry připojeného rádia zobrazující se v prostředí Simulink

5.1 Převedení kódu na model v prostředí Simulink

Algoritmus Greedy pro přičítání popsaný v kap. 4 byl převeden na model v prostředí Simulink, jehož grafické vyjádření je srozumitelnější pro pochopení celé operace. Celý algoritmus je rozčleněn do 6 modelů – kroků.

5.1.1 Pilotní signál a jeho vysílání

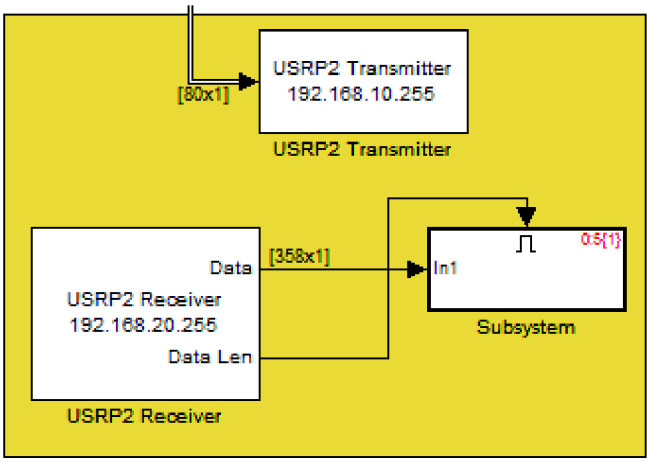
První krok se zabývá tvorbou pilotního signálu a jeho vysláním. Nejprve je vytvořena konstanta, což je 52 pevně určených QPSK symbolů, z nichž každý je na jednom řádku (tedy jedné nosné). Jak je vidno na obrázku 5.3, tento vektor je rozdělen bloky *Selector* na poloviny, mezi něž je vloženo 11 nulových nosných. Na počátek je navíc přidána nulová DC nosná. Pomocí bloku *Vector Concatenate* jsou tyto složky signálu spojeny do jednoho vektoru, který je vyslán do bloku *IFFT*, k jehož výstupu je, opět pomocí bloků *Selector* a *Vector Concatenate*, přidán cyklický prefix o délce 1/4, tedy 16 vzorků ze 64.



Obr. 5.3: Krok 1 - přidání nul a CP

Výsledný vektor 80 vzorků je vložen do rámce a poslán do bloku vysílače USRP2, který má nastavenu velikost interpolace 5. Vysílačem je datový tok vložen do UDP rámců o velikosti 8000 vzorků a převeden na analogový signál na zvolené frekvenci (f = 2,451234 GHz).

Přijímač je nastaven pomocí decimace na vzorkovací frekvenci 20 MS/s, což odpovídá vzorkovací frekvenci WLAN 20MHz. Výstupní port *Data Len* zapíná zápis přijatého vzorku do workspace pokaždé, když je přijat nenulový vzorek.

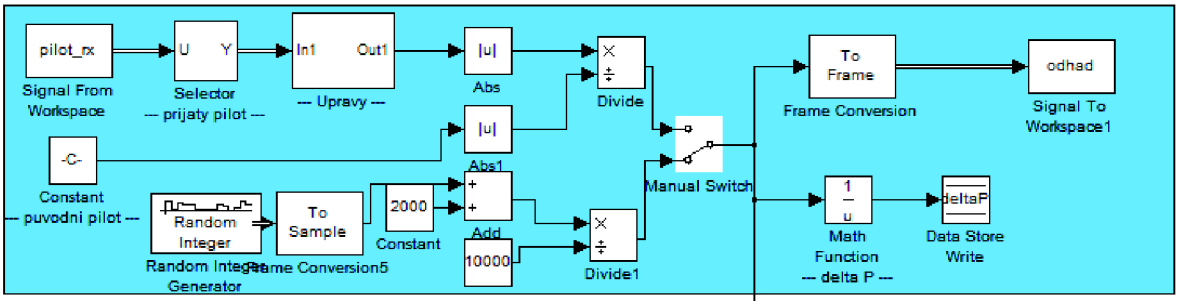


Obr. 5.4: Krok 1 – vysílač a přijímač USRP2

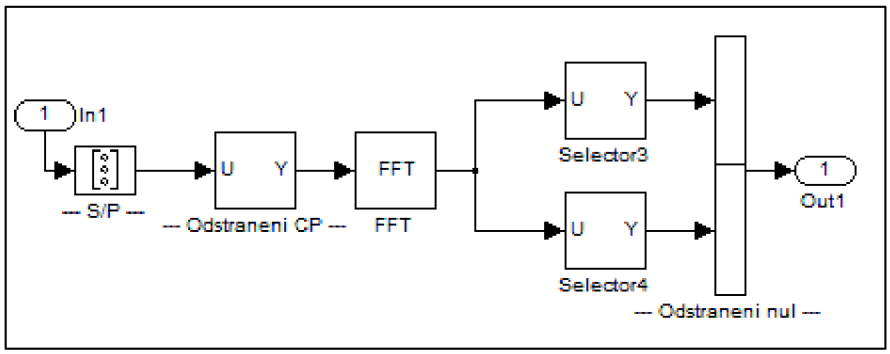
5.1.2 Odhad přenosu, Greedy a potřebný výkon

Z důvodů inicializace přijímače je nutno pilotní signál vyslat opakovaně a z přijímaných vzorků vybrat vzorky 6117-6196. Je nutno vybrat právě vzorky s těmito indexy, protože zahrnují první celou přijatou sekvenci vysílaného signálu reprezentujícího pilotní signál. Předešlé přijaté vzorky jsou šumem, popř. jen částí sekvence vysílaného signálu. Tento výběr je zprostředkován blokem *Selector*, jehož výstup je přiveden do masky *Upravy* (viz obr. 5.5 a 5.6), ve které jsou z přijatého vektoru odstraněny vzorky odpovídající cyklickému prefixu, na signál je v bloku *FFT* aplikována rychlá Fourierova transformace a z vektoru jsou odstraněny vzorky odpovídající nulám vloženým ve vysílači. Tím je zrekonstruováno 52 symbolů, jejichž absolutní hodnoty jsou vzorek po vzorku vyděleny absolutní hodnotou původního pilotního signálu. Získaná reálná čísla jsou velikostí přenosu na jednotlivých 52 subnosných.

Druhou možností získání hodnot přenosu je automatické generování těchto hodnot v rozmezí 0,2-1,2. Toto rozmezí lze upravit přenastavením implicitně přičítané konstanty 2000. Přepínačem mezi oběma možnostmi je blok *Manual Switch*.

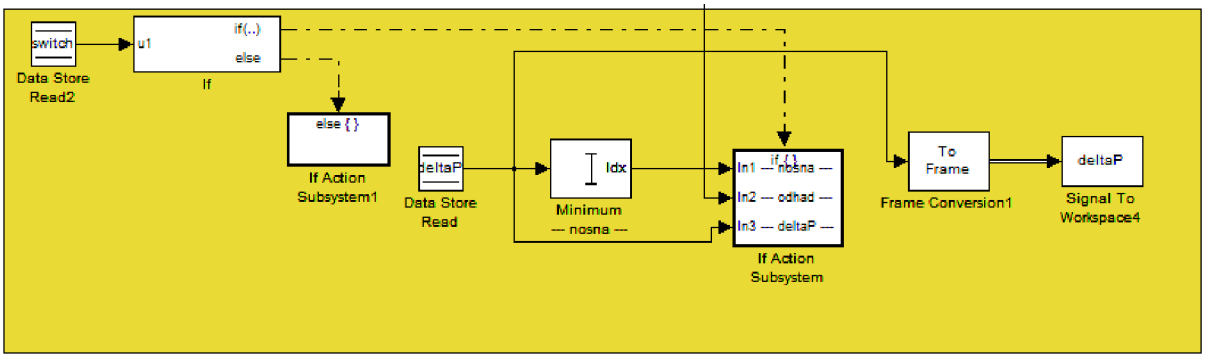


Obr. 5.5: Krok 2 – odhad přenosu



Obr. 5.6: Krok 2 - detail bloku Upravy

Se znalostí přenosu kanálu lze přistoupit k určení počtu bitů k přenosu na jednotlivých subnosných pomocí Greedy algoritmu na obrázcích 5.7, 5.8 a 5.13. Protože je předpokládána vzorkovací frekvence vysílaného signálu 20MHz, je nutno v modelu nastavit dobu simulace rovnu N-násobku vzorkovací periody (1/20e6), kde N je požadovaný počet bitů.

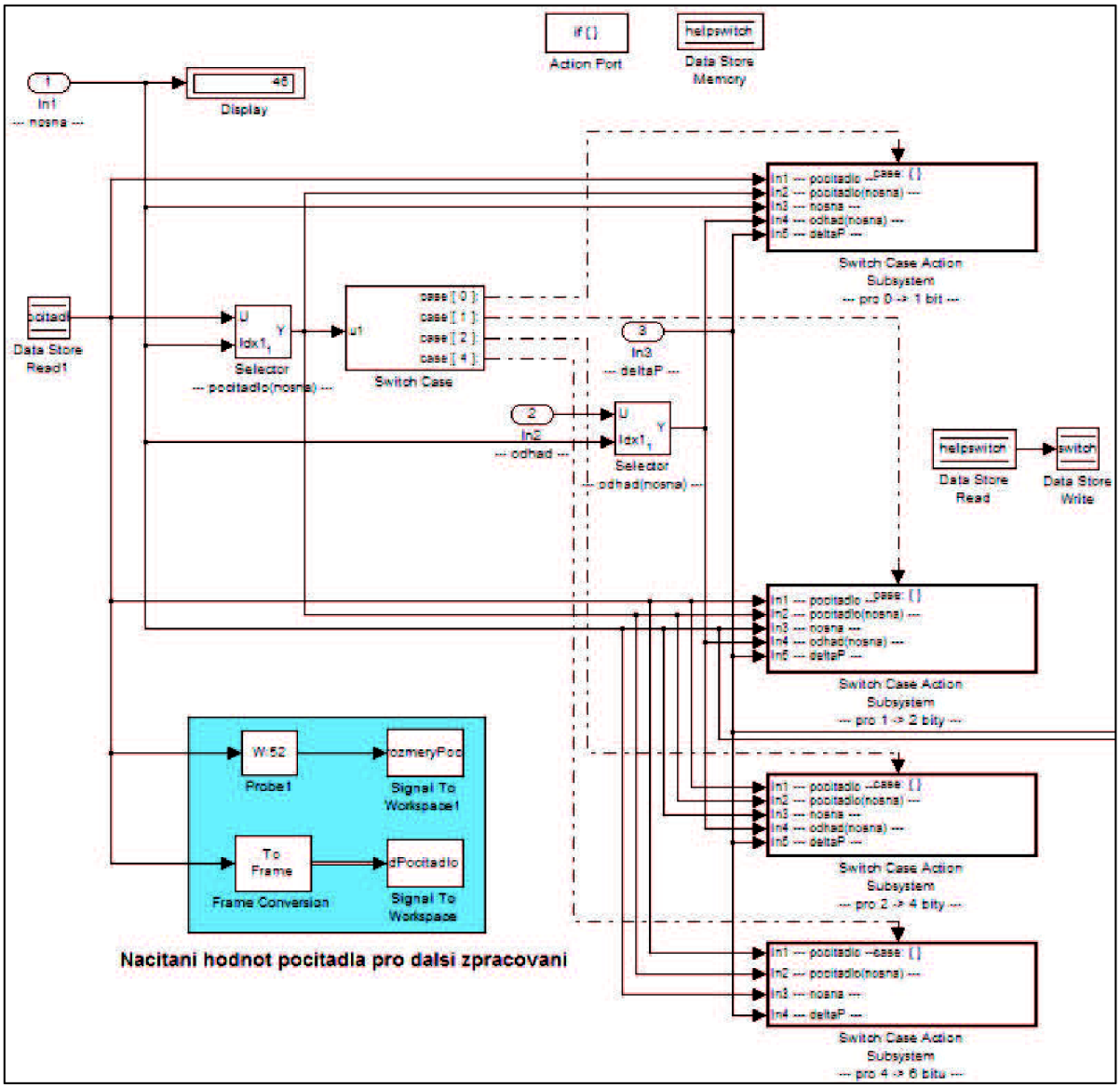


Obr. 5.7: Krok 2 – Greedy algoritmus (1. část)

Základem algoritmu je blok *If*, pomocí kterého je určeno, zda byly minulou periodu simulace některé nosné přiřazeny dva bity. Pokud ano, simulace přejde k další periodě, pokud ne, pokračuje dále do bloku *If Action Subsystem*. Tento blok je zobrazen na obrázku 5.8 a vstupují do ní hodnoty:

- pořadí nosné s nejmenším nárůstem výkonu při přidání dalšího 1 nebo 2 bitů
- vektor velikosti přenosu na jednotlivých nosných
- vektor nárůstu výkonu při přidání dalšího 1 nebo 2 bitů na každé nosné

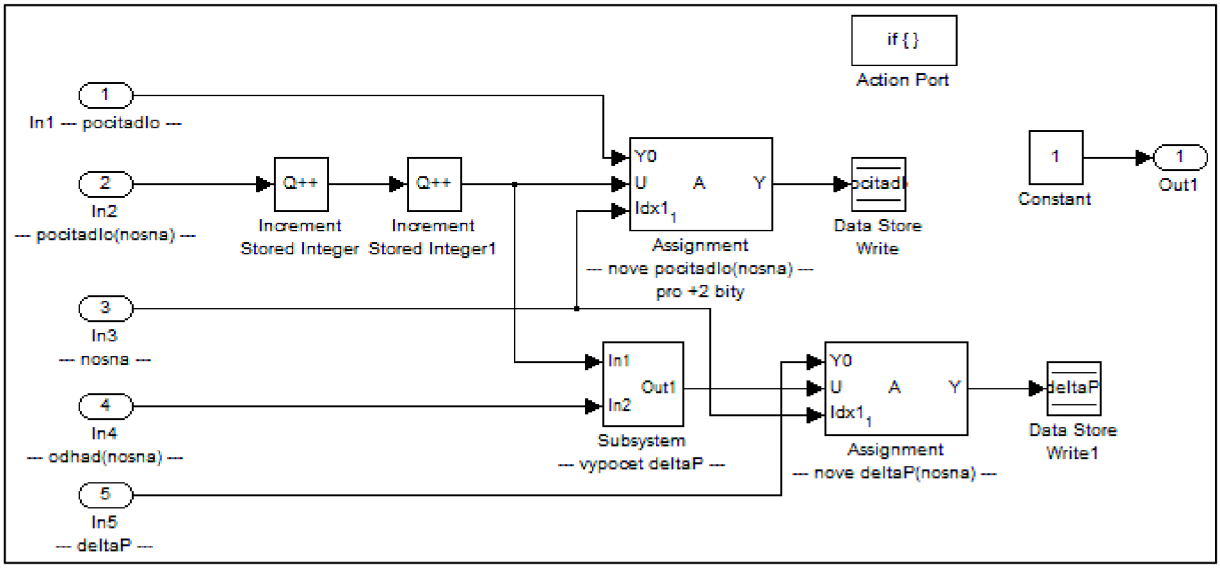
Uvnitř bloku *If Action Subsystem* (viz obr. 5.8) je umístěn blok *Selector*, který vytvoří proměnnou *pocitadlo(nosna)* shodnou se stejně nazvanou proměnnou v kap. 4. Blok *Switch* na základě velikosti této proměnné rozhodne, zda budou na nosnou přidány 2 bity nebo jen 1 a jaká bude nově vypočtená hodnota ΔP. Pro všechny možnosti jsou připojeny bloky *Switch Case Action*.



Obr. 5.8: Krok 2 - detail bloku If Action Subsystem (úroveň 1)

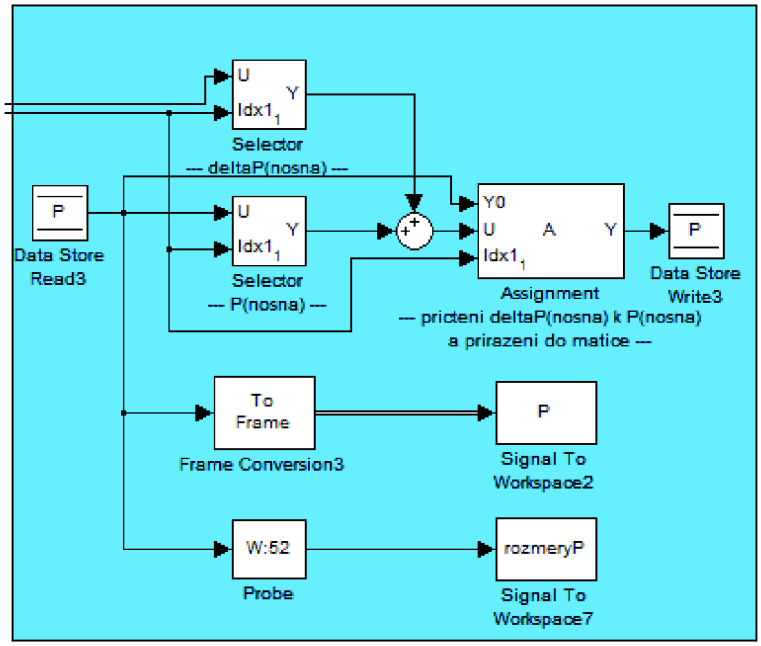
Tyto bloky zvýší proměnnou *pocitadlo(nosna)* o 1 nebo 2 bity (pokud jen 1, z obr. 5.9 by ubyl jeden z bloků *Increment Stored Integer*). Takto upravenou hodnotu poté pomocí bloku *Assignment* přiřadí do vektoru *pocitadlo* na správný index dané proměnnou *nosna*. Zároveň zvýšená hodnota *pocitadla* slouží pro výpočet nové hodnoty ΔP (podle rovnice 4.1a), která je rovněž přiřazena do vektoru všech změn výkonu na danou nosnou.

V případě přidání dvou bitů na nosnou je výstupem bloku *Switch Case Action* hodnota 1, v případě přidání jednoho bitu je to 0. Tyto výstupy slouží k identifikaci počtu přiřazených bitů a možnému přeskočení další periody simulace (viz první odstavec pod obr. 5.7).



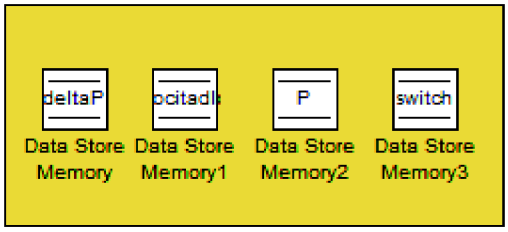
Obr. 5.9: Krok 2 - detail bloku Switch Case Action (úroveň 2)

Každá použitá hodnota ΔP je přičítána k hodnotě potřebného výkonu na dané nosné a pomocí bloků *Assignment* a *Data Store* je uložena k dalšímu zpracování. Sumou konečných hodnot na každé nosné je získán potřebný výkon napříč celým systémem a lze ho srovnat s teoreticky vypočítanou hodnotou stejně jako v kap. 4.3.



Obr. 5.10: Krok 2 – výpočet celkového potřebného výkonu pro každou nosnou

Paměťové bloky *Data Store Memory*, společně s bloky na zápis a čtení dat z této paměti, slouží jako skalární i vektorové proměnné, díky kterým je možno jednoduše přenášet hodnoty mezi různými místy simulace. V tomto kroku byly paměti použity celkem 4 – viz obr. 5.11. Navíc je v bloku *If Action Subsystem* umístěn jeden pomocný paměťový blok (*helpswitch*).



Obr. 5.11: Krok 2 – použité paměťové bloky

Poslední částí kroku 2 je generace samotných dat k přenosu. Blokem binárního generátoru o vzorkovací frekvenci 20 MHz je během simulace vytvořeno implicitně nastavených 156 bitů (toto číslo lze změnit úpravou doby simulace dle vzorce 5.1, ale počet bitů musí být v rozmezí 1-312), které jsou uloženy do workspace do proměnné *data*. Počet bitů je vhodné volit jako násobek 52, aby bylo možno v dalším kroku vypočítat úsporu výkonu při použití Greedy algoritmu (viz tabulka 5.2).

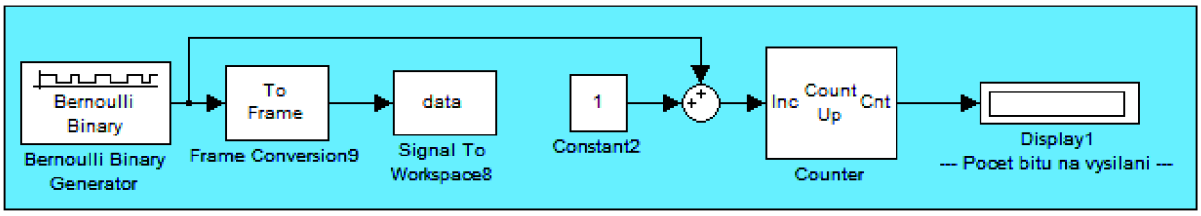
$$t_{SIM} = \frac{1}{2 \cdot 10^7} \cdot N, \tag{5.1}$$

kde t_{SIM} je doba simulace a N je požadovaný počet vysílaných bitů.

Tab. 5.3: Nastavení doby simulace pro různé počty bitů

Počet bitů	Doba simulace [s]
52	0,0000025999
104	0,0000051999
156	0,0000077999
208	0,0000103999
260	0,0000129999
312	0,0000155999

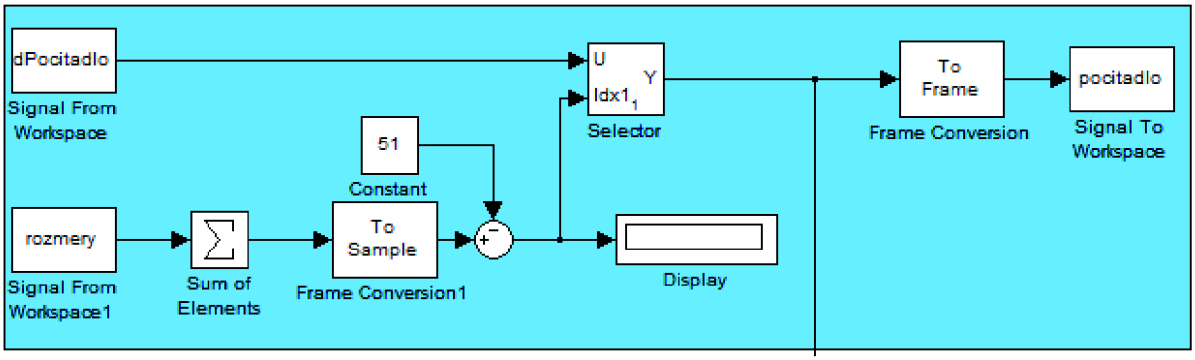
Blok *Display* po doběhnutí kroku 2 zobrazuje počet vygenerovaných bitů, díky tomu je zpětná vazba okamžitá a při nesprávně nastavené době simulace lze upravit počet generovaných bitů ihned, bez nutnosti dokončit simulaci celého algoritmu.



Obr. 5.12: Krok 2 – generace dat k vysílání

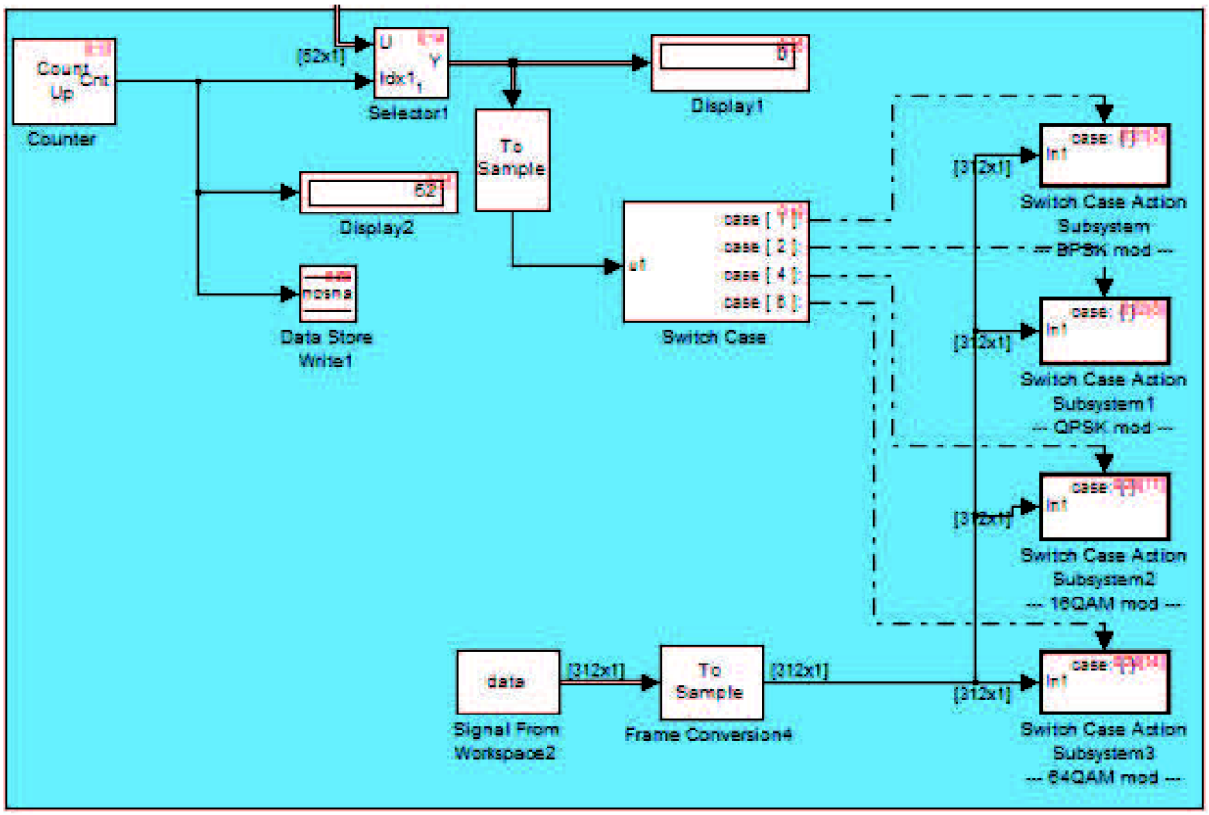
5.1.3 Alokace bitů a modulace

Protože je do proměnné *dPocitadlo* uložen vždy celý vektor pro 52 nosných, je nutné na začátku dalšího kroku vybrat jen posledních 52 vzorků. Toho je docíleno pomocí proměnné *rozmery* získané v předchozím kroku, která určí celkovou velikost proměnné *dPocitadlo* a od ní odečte 51. Od vzorku tohoto pořadí dále se nacházejí požadované hodnoty, které jsou vybrány pomocí bloku *Selector*, uloženy do workspace proměnné *pocitadlo* a zároveň poslány dále v rámci modelu.



Obr. 5.13: Krok 3 – Greedy algoritmus (2. část)

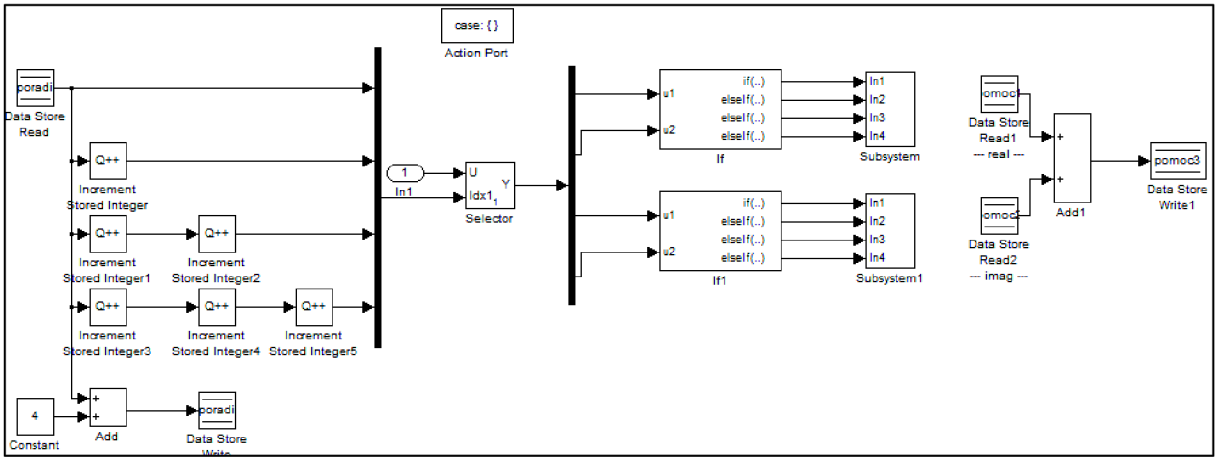
Pro každou nosnou danou blokem *Counter* je určena velikost proměnné *pocitadlo* s tímto indexem. Blok *Switch Case* pak podle hodnoty proměnné *pocitadlo* zvolí správnou modulaci, pomocí které je zakódován správný počet bitů z proměnné *data* do OFDM symbolu. Stavba vnitřní modulace bude vysvětlena na příkladu modulace 16QAM. Doba simulace tohoto kroku je nastavena na $52*T_s$, kde T_s je vzorkovací perioda signálu. Všechny operace tak proběhnou pro každou nosnou a zároveň je udržována vzorkovací frekvence 20 MHz.



Obr. 5.14: Krok 3 – Alokace bitů a modulace

Základem používání modulací je sledování indexu alokovaných bitů. Index prvního bitu určeného k alokaci je průběžně ukládán do proměnné *poradi*. Každá modulace vybere bity s příslušnými indexy (např. pro index prvního bitu určeného k alokaci 15 a modulaci 16QAM budou alokovány bity s indexy 15-18) a index dalšího dosud nealokovaného bitu (v tomto případě 19) je zapsán do proměnné *poradi*, odkud může být načten jinou modulací.

Zvolené sekvence bitů jsou z datového toku vyňaty pomocí bloku *Selector* a rozděleny na poloviny (při QPSK na jednotlivé bity, při 16QAM na dibity a při 64QAM na trojice bitů). Tyto poloviny pak charakterizují reálnou a imaginární složku OFDM symbolu, která je zvolena pomocí bloku *If*. Na základě posloupnosti bitů jsou do pamětí *pomoc1* a *pomoc2* uloženy obě složky, jejichž součet, tedy OFDM symbol, je uložen do paměti *pomoc3*.



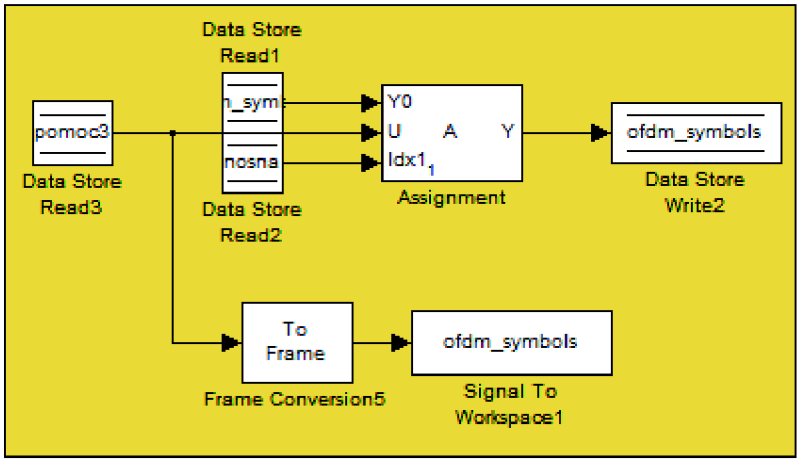
Obr. 5.15: Krok 3 – detail bloku Switch Case Action pro modulaci 16QAM

Tab. 5.4: Přehled přiřazování částí symbolů dle posloupností bitů

QPSK	bity	symbol	64QAM	bity	symbol
1. polovina	0	i	**1. polovina**	[0 0 0]	-7
	1	-i		[0 0 1]	-5
2. polovina	0	1		[0 1 1]	-3
	1	-1		[0 1 0]	-1
				[1 1 0]	1
16QAM	**bity**	**symbol**		[1 1 1]	3
1. polovina	[0 0]	-3		[1 0 1]	5
	[0 1]	-1		[1 0 0]	7
	[1 1]	1	**2. polovina**	[0 0 0]	-7i
	[1 0]	3		[0 0 1]	-5i
2. polovina	[0 0]	-3i		[0 1 1]	-3i
	[0 1]	-i		[0 1 0]	-i
	[1 1]	i		[1 1 0]	i
	[1 0]	3i		[1 1 1]	3i
				[1 0 1]	5i
				[1 0 0]	7i

Takto vytvořené symboly jsou postupně načítány do workspace proměnné *ofdm_symbols*, která je použita v dalším kroku. Paměťové bloky použité v tomto kroku jsou:

- *poradi*
- *ofdm_symbols*
- *nosna*
- *pomoc 1, 2, 3*



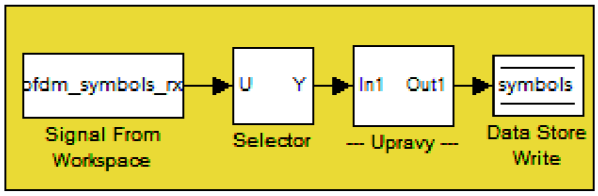
Obr. 5.16: Načítání symbolů

5.1.4 Přenos OFDM symbolů

Symboly vytvořené v předchozím modelu jsou upraveny a přeneseny stejným způsobem, jako pilotní signál v kroku 1. Tedy jsou přidány nuly, na výsledný datový tok je použita IFFT a nakonec je přidán cyklický prefix. Takto upravená data jsou vysílána bloky USRP2 (viz obr. 5.3 a 5.4).

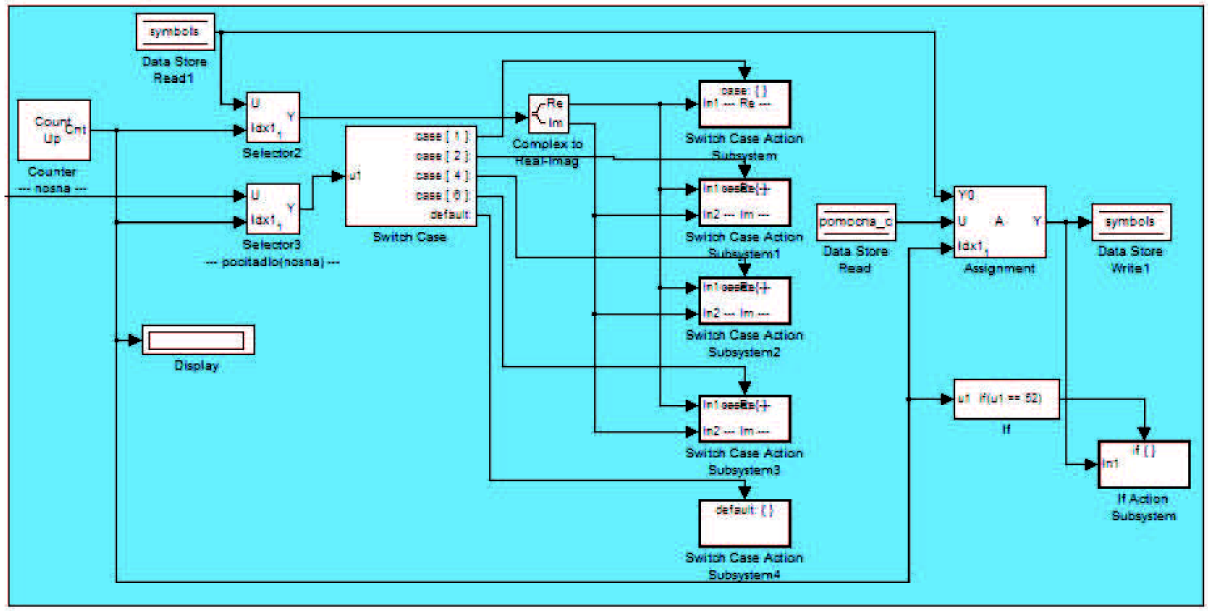
5.1.5 Zaokrouhlování

Přenos OFDM symbolů je stejně jako v kap. 5.1.2 nutno opakovat kvůli inicializaci přijímače a z přijatých symbolů je nutné vybrat ty s indexy 6133 - 6212. Je nutno vybrat právě vzorky s těmito indexy, protože zahrnují první celou přijatou sekvenci vysílaného signálu reprezentujícího OFDM signál. Předešlé i další vzorky je možno zanedbat, jsou to periodicky opakované posloupnosti vysílaných dat, popř. šum.



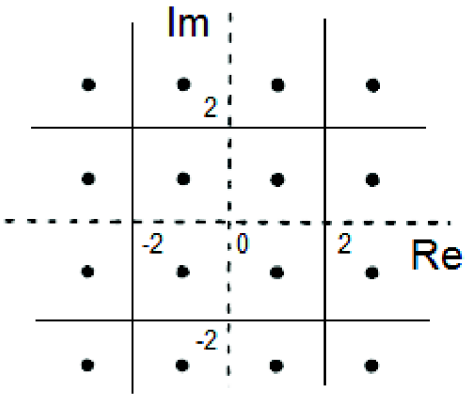
Obr. 5.17: Výběr symbolů, odstranění CP a nul

Zaokrouhlování přijatých symbolů je nutné pro následující demodulaci, aby byly zajištěny symboly určující některý stav modulace. Není možné použít žádnou ze zaokrouhlovacích funkcí prostředí Simulink, protože některé symboly mohou nabýt hodnoty i o 2 vyšší než je maximum pro danou modulaci a byly by ztraceny. Proto je tento problém řešen konstrukcí *Switch Case* se zaokrouhlením pro každou modulaci. Blok *Switch Case* je řízen hodnotou proměnné *pocitadlo* na dané nosné a symbol vstupující do bloku Switch Case Action, vykonávaného při splnění jeho podmínek, je rozdělen na reálnou a imaginární část.



Obr. 5.18: Zaoukrouhlování symbolů

Proces zaoukrohlování v jednotlivých subsystémech se dá popsat tak, že konstelační diagram dané modulace je rozdělen mřížkou (viz obr. 4.10 a 5.19), která protíná osy vždy uprostřed mezi dvěma stavy modulace. Reálná a imaginární složka jsou zaokrouhleny k nejbližší možné hodnotě a následně uloženy do pamětí *pomocna_r* a *pomocna_i*. Tyto jsou pak sečteny a uloženy do paměti *pomocna_c*, která je pak pomocí bloku Assignment přiřazena na danou nosnou a celý vektor symbolů je uložen do paměti *symbols*. Až blok *Counter* dosáhne hodnoty 52 (je zaokrouhlen poslední symbol), uloží se celý vektor symbolů do workspace proměnné *ofdm_zaokrouhlene*, která je posléze použita v dalším kroku.



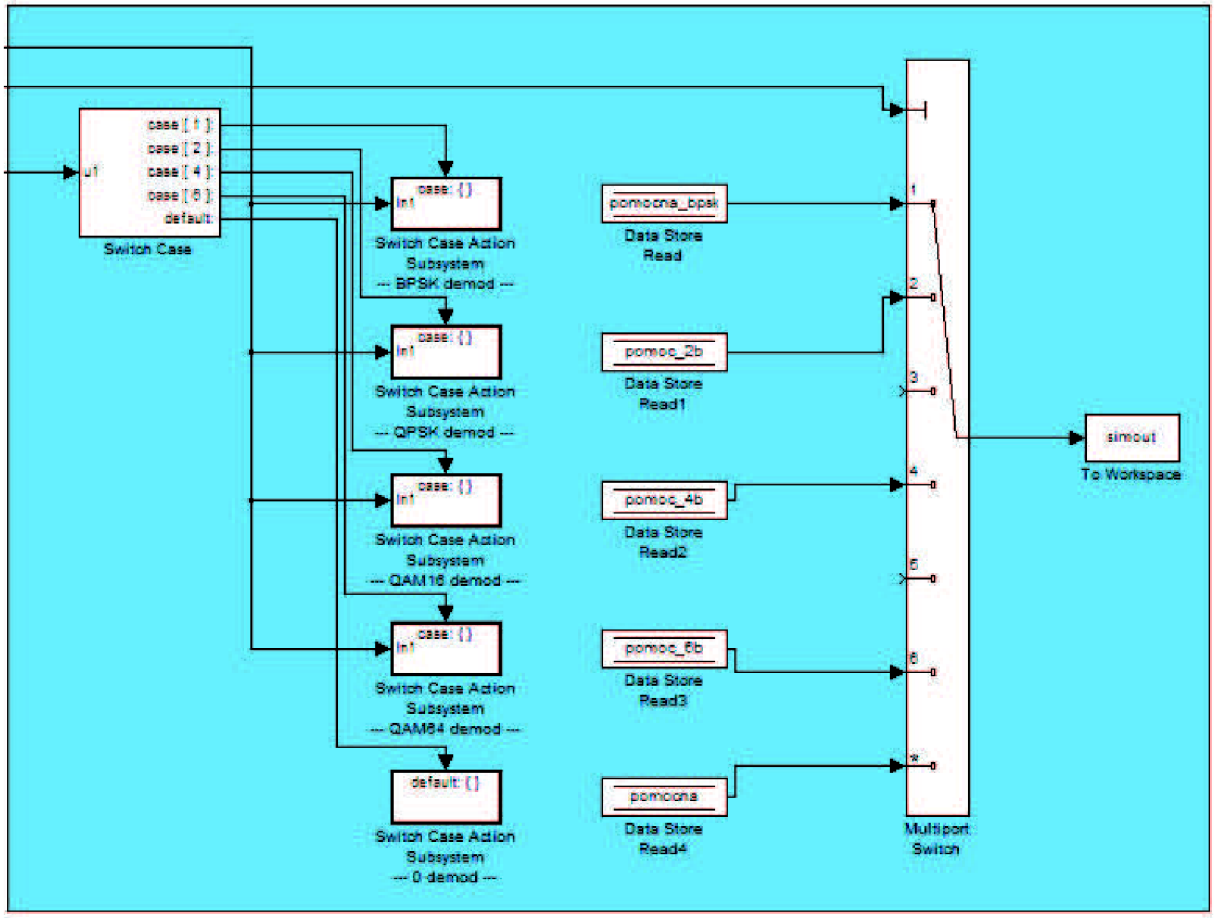
Obr. 5.19: Zaokrouhlovací mřížka pro modulaci 16QAM [17]

Paměťove bloky použité v tomto kroku jsou:

- *symbols*
- *pomocna_r*
- *pomocna_i*
- *pomocna_c*

5.1.6 Demodulace

Demodulace je řešena opět pomocí rozhodovacího bloku *Switch Case*, který spustí demodulaci na základě počtu bitů na proměnné *pocitadlo* na dané nosné. Demodulace je inverzní operací k modulaci, proto jednotlivé demodulátory využívají opačnou operaci, než jaká byla popsána v kap. 5.1.3. Převod symbolů na bity je možno rovněž provést pomocí tabulky 5.3.



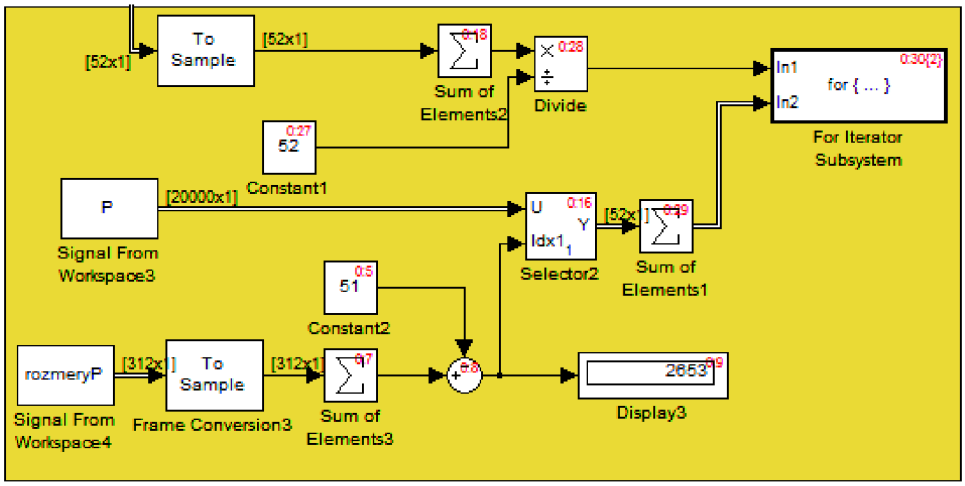
Obr. 5.20: Demodulace

Demodulované bity jsou posílány do bloku *Multiport Switch* s více vstupními datovými porty a jedním ovládacím portem, který je ovládán, stejně jako blok *Switch Case*, hodnotou proměnné *pocitadlo(nosna)* a na výstup přiřazuje vždy to bitové slovo, které je aktuální. Toto opatření je nutné kvůli tomu, že při každém kroku jsou vysílány ze všech pomocných pamětí v nich uložená bitová slova, proto nelze použít ani součet, ani spojení vektorů pomocí bloku *Vector Concatenate*. Výstupní data jsou pak kvůli své nestejné délce uložena do struktury *simout*, která obsahuje jak hodnoty počítadla, tak výstupní bity i časové kroky, ve kterých bylo do struktury zapisováno. Po manuálním srovnání vyslaného a přijatého datového toku lze dojít ke zjištění, že jde o bezchybný přenos.

Paměťové bloky použité v tomto kroku jsou 1-bitový (3x), 2-bitový, 4-bitový a 6-bitový.

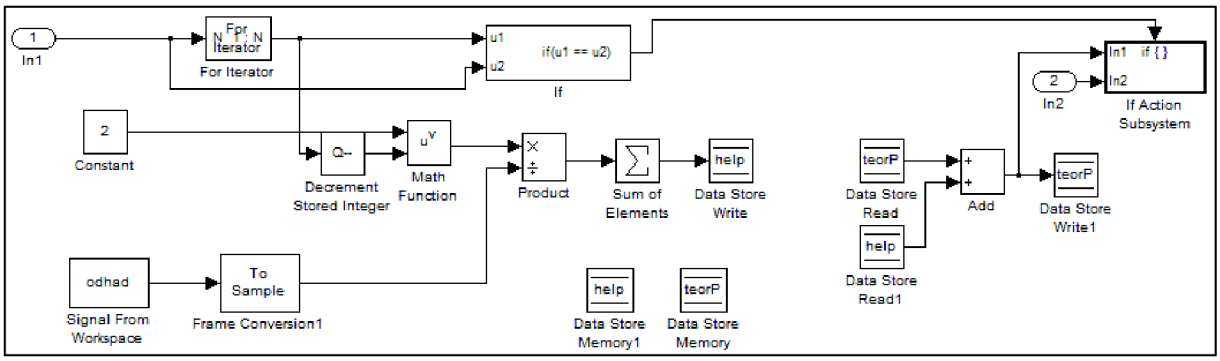
5.2 Úspora výkonu při použití USRP2

Úspora výkonu je vypočítána v kroku 3. Jejím základem je vydělení počtu vyslaných bitů počtem nosných (tedy 52). Získané číslo je počtem iterací následujícího bloku *For*.



Obr. 5.21: Výpočet úspory výkonu (úroveň 1)

V cyklu *For* je aktuální iterace snížena o 1 a je vypočten potřebný výkon pro vyslání 1 bitu na každé nosné, který je uložen do pomocné paměti *help*. Tento výkon je připočten k celkové hodnotě *teorP*. Tento proces proběhne Nkrát, kde *N* je vstup do bloku *For* a poslední hodnota ukládaná do paměti *teorP* je rovněž uložena do stejnojmenné workspace proměnné.

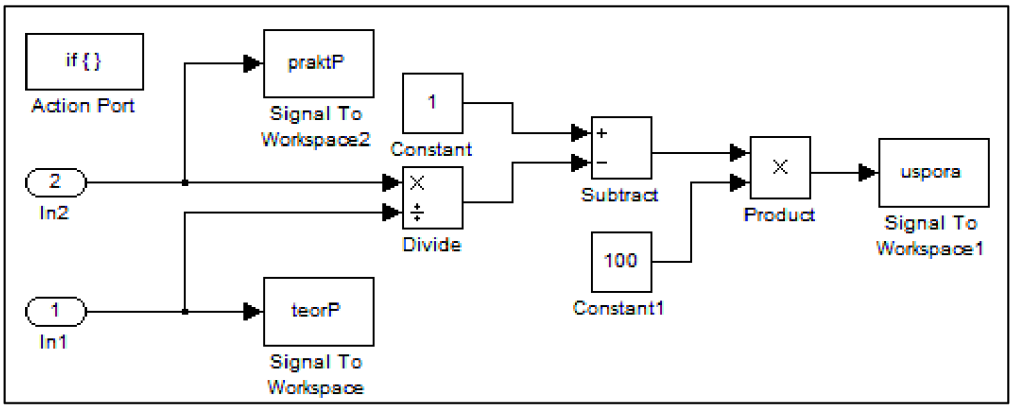


Obr. 5.22: Výpočet úspory výkonu (úroveň 2)

Zároveň je spočtena úspora výkonu, která je rovna:

$$P_{ÚSP} = \left(1 - \frac{P_{MĚŘ}}{P_{TEOR}}\right) \cdot 100, \tag{5.2}$$

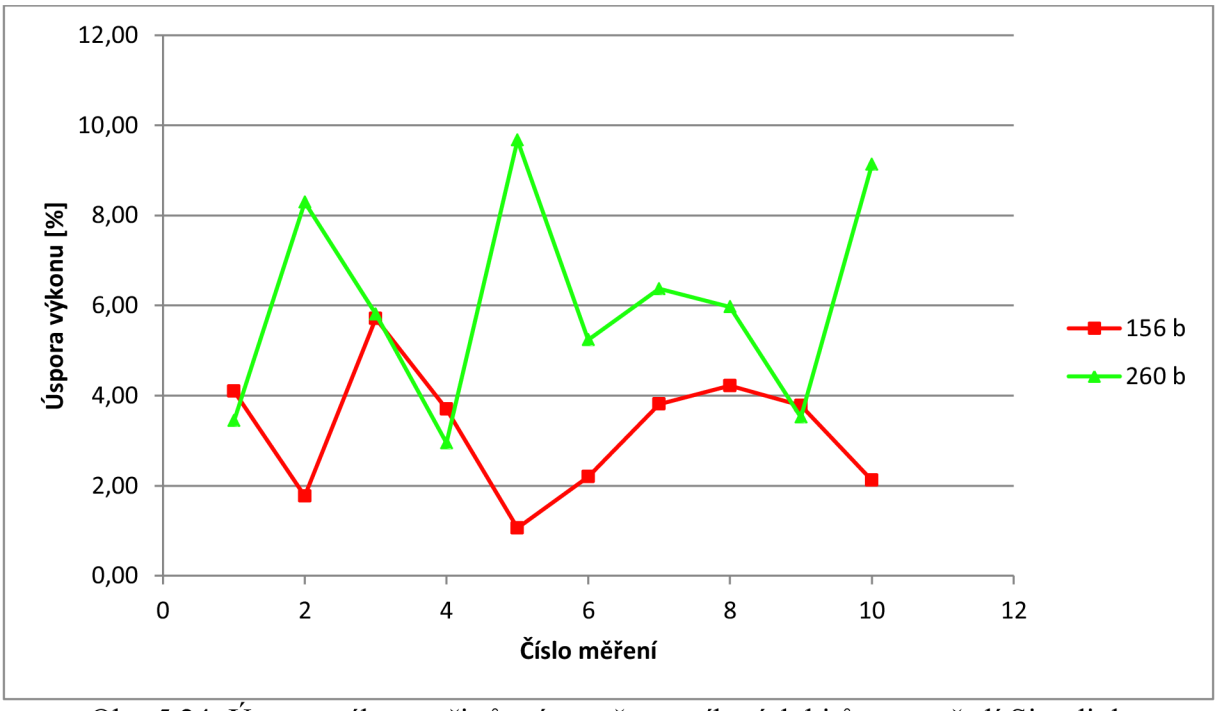
kde $P_{MĚŘ}$ je změřený výkon uložený ve workspace proměnné *praktP* a P_{TEOR} je teoreticky spočtený výkon uložený ve workspace proměnné *teorP*.



Obr. 5.23: Výpočet úspory výkonu (úroveň 3)

Protože se odhad kanálu při vysílání pomocí dvou rádií USRP2 vzdálených od sebe 2 metry pohybuje pouze v rozmezí 0,9999-1,0001, byla do modelu kvůli zjištění správné funkce algoritmu zařazena možnost náhodné generace hodnot odhadu přenosu pro jednotlivé subnosné v rozmezí 0,2-1,2. Pro další měření v reálných podmínkách postačí přepnout blok *Manual switch*, který pak bude opět načítat přenos z reálných naměřených hodnot.

Při použití algoritmu převedeného do prostředí Simulink a s manuálním zadáním přenosu jednotlivých subnosných se úspora výkonu pohybuje nejčastěji mezi 2% a 8%. Lze předpokládat, že při měření v reálném prostředí při vyšší vzdálenosti vysílače a přijímače se bude hodnota úspory výkonu pohybovat v podobných mezích.



Obr. 5.24: Úspora výkonu při různém počtu vysílaných bitů v prostředí Simulink

6 Shrnutí

Cílem diplomové práce bylo prostudovat možnosti adaptivního nastavení systému OFDM a implementovat je do modelu vytvořeného v prostředích MATLAB a Simulink. Základem pro toto nastavování je zjištění stavu kanálu pomocí pilotních signálů. Takto nabytá data o přenosu subkanálů jsou pak použita při přidělování prostředků, tedy dat a modulačního schématu a k následnému výpočtu potřebného výkonu.

Rozšířeným principem pro přidělení výkonu a bitů je water-filling, který alokuje bity na subkanály s nejvyšším přenosem. Při použití tohoto principu v podobě Greedy algoritmu je možno volit mezi možnostmi maximalizovat datovou rychlost nebo minimalizovat potřebný výkon. Tato volba závisí na konkrétním použití každého systému OFDM.

V diplomové práci byly popsány tři způsoby volby vnitřní modulace subnosných modulace OFDM. Nejjednodušším typem je algoritmus pevného práhu, který přejde na typ modulace s vyšším počtem bitů na symbol pokaždé, když SNR subpásma stoupne nad pevně danou hodnotu. Složitějším typem je algoritmus odhadu BER subpásma, kdy se kalkuluje s pravděpodobností chyby na subkanálech při dané modulaci a je zvoleno takové schéma, které zajišťuje nejvyšší rychlost přenosu při zachování jisté úrovně chybovosti. Posledním popisovaným způsobem volby modulace je algoritmus konstantní datové rychlosti, který zavádí cenu přenosu závislou na poměru bitových chyb a přenosové rychlosti a volí nastavení modulace s nejnižší cenou. Algoritmus pevného práhu, konkrétně časově neměnná modulace všech symbolů na dané subnosné, se stal základem pro vytvoření algoritmu použitého v simulaci v prostředí MATLAB, který minimalizuje potřebný výkon k přenosu určeného datového toku.

V prostředí MATLAB byl vytvořen model OFDM s vybranými parametry systému WLAN, který vysláním blokového pilotního signálu zjistí výši přenosu na všech subnosných. Poté je vygenerována nová datová sekvence k přenosu a funkce, která přiřadí nejvyšší počet bitů k přenosu na nosnou s nejvyšším přenosem. Kvůli názornosti je na každé nosné přenášen jen 1 OFDM symbol. Na základě předešle získaných dat jsou bity datové sekvence alokovány na jednotlivé nosné a modulovány příslušnou vnitřní modulací dle počtu bitů na nosné. Model využívá vnitřní modulace BPSK, QPSK, 16QAM a 64QAM. Po přenosu kanálem jsou OFDM symboly demodulovány a převedeny zpět na bitovou sekvenci.

Po určení počtu bitů k přidělení na jednotlivé subnosné je spočtena úspora výkonu při použití Greedy algoritmu, tedy různém počtu bitů na subnosných, oproti případu se stejným počtem bitů na všech nosných. Tato úspora se v závislosti na počtu vyslaných bitů pohybovala mezi 20% a 70%, přičemž při nižším počtu vysílaných bitů byla úspora vyšší.

Na základě této simulace byl vytvořen model v prostředí Simulink, ve kterém byly s pomocí softwarového rádia USRP2 zjištěny stejné veličiny jako v simulaci v prostředí MATLAB a s jejich pomocí byla opět spočtena úspora výkonu. Kvůli příliš malému rozmezí přenosu kanálu na jednotlivých subnosných byly pro další operace zvoleny náhodné hodnoty přenosu kanálu v rozmezí 0,2-1,2. Úspora výkonu pak byla nižší než teoretická, její aritmetický průměr činil 3,25% pro 156 vysílaných bitů a 6,04% pro 260 vysílaných bitů. Lze tedy sledovat, že i přes nepoužití modulací pro 3 a 5 bitů je Greedy algoritmus s ohledem na úsporu výkonu výhodný.

Mezi doporučené možnosti, jak navázat na tuto práci, patří zakomponování modelů v prostředí Simulink do jednoho celistvého modelu, aby všechny operace byly spustitelné sekvenčně. Pro další zpracování signálu by bylo vhodné doplnit model o další krok, ve kterém budou vyjmuty demodulované bity ze struktury a zapsány do workspace proměnné, se kterou se pracuje jednodušeji. Dalším krokem by mělo být nastavení požadovaného počtu vysílaných bitů v dialogovém okně, podle kterého by se doba simulace nastavovala automaticky. Po těchto úpravách by bylo vhodné zrušit omezení 1 OFDM symbolu na nosnou, aby mohl být použitý algoritmus prozkoumán při vyšším počtu vysílaných bitů, popř. upraven do podoby maximalizace přenosové rychlosti při konstantním přiděleném výkonu.

Seznam použité literatury

[1] BUMBÁLEK, Zdeněk. *Modulační techniky v moderních bezdrátových sítích* [online]. 8.2.2010 [cit. 2010-04-12]. Dostupný z WWW: <http://access.feld.cvut.cz/view.php?cisloclanku=2010020004>.

[2] WU, Zhibin.; LIU, Yan., JING, Xiangpeng. *Channel Estimation in OFDM Systems* [online]. 2003 [cit 2010-04-13]. Prezentace. Dostupné z WWW: <http://www.winlab.rutgers.edu/~spasojev/courses/projects/CE_OFDM.ppt>.

[3] IEEE 802.11. *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York : IEEE, 12.6.2007 [cit. 2010-04-13]. 1184 s. Dostupné z WWW: <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>.

[4] SHEN, Yushi; MARTINEZ, Ed. *Channel Estimation in OFDM Systems* [online]. 2006. Dostupný z WWW: <http://code.ucsd.edu/~yushen/publications_files/Yushi%20Shen-Jan06-Freescale-OFDM.pdf>.

[5] COLERI, Sinem, et al. Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems. In *IEEE Transactions On Broadcasting,* [online]. [s.l.] : IEEE, 2002, Vol. 48, No. 3 [cit. 2010-04-17]. Dostupné z WWW: <http://wow.eecs.berkeley.edu/ergen/docs/ofdm.pdf>.

[6] COLERI, Sinem; ERGEN, Mustafa. *A Survey on Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems* [online]. 2002 [cit. 2010-04-17]. Dostupné z WWW: <http://wow.eecs.berkeley.edu/ergen/docs/OFDM_IPCN02.ppt>.

[7] PAPANDREOU, Nikolaos, ANTONAKOPOULOS, Theodore. Bit and Power Allocation in Constrained Multicarrier Systems: The Single-User Case. In *EURASIP Journal on Advances in Signal Processing* [online]. New York : Hindawi Publishing Corporation, 2008 [cit. 2010-04-18]. Dostupné z WWW: <http://downloads.hindawi.com/journals/asp/2008/643081.pdf>.

[8] KELLER, Thomas; HANZO, Lajos. Adaptive Modulation Techniques for Duplex OFDM Transmission. In *IEEE Transactions On Vehicular Technology* [online]. [s.l.] : IEEE, 2000, Vol. 49, No. 5 [cit. 2010-04-21]. Dostupné z WWW: <http://eprints.ecs.soton.ac.uk/4458/1/49vt05-keller.pdf>.

[9] TORRANCE, Jeffery M.; HANZO, Lajos. Upper bound performance of adaptive modulation in a slow Rayleigh fading channel. In *Electronics Letters* [online]. [s.l.] : [s.n.], 11.4.1996, Vol. 32, No. 8 [cit. 2010-04-25]. Dostupné z WWW: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00491048>.

[10] MARŠÁLEK, Roman. *Teorie rádiové komunikace - přednáška č. 8: OFDM*. Brno, 2010 [cit. 2010-12-23]. 44 s. Prezentace. VUT v Brně, FEKT.

[11] ALMEIDA, Nuno T.; ABRANTES, Sílvio A.. *A novel approach to ARQ error control mechanisms for wireless LANs communications* [online]. 2000 [cit. 2010-12-23]. Dostupné z WWW: <http://ieeexplore.ieee.org/iel5/7152/19253/00891003.pdf>.

[12] ETTUS RESEARCH. *USRP2 Datasheet* [online]. 2010 [cit. 2011-05-12]. Dostupné z WWW: <http://www.ettus.com/downloads/ettus_ds_usrp2_v5.pdf>.

[13] WYSOCKI, Tadeusz A.; ZEPERNICK, Hans-Jürgen. Characterization of the indoor radio propagation channel at 2.4 GHz. In *Journal Of Telecommunications And Information Technology* [online]. [s.l.] : [s.n.], 2000 [cit. 2011-05-11]. Dostupné z WWW: <http://www.elec.uow.edu.au/staff/wysocki/publications/J4.pdf>.

[14] GUÉRIN, Sebastian; GUO, Yingjie J.; BARTON, Stephen K. Indoor propagation measurements at 5GHz for HiperLAN. In *10th International Conference on Antennas and Propagation Conference Publication* [online]. [s.l.] : [s.n.], 1997 [cit. 2010-12-29]. Dostupné z WWW: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=606993>.

[15] IEEE 802.11n. *IEEE Standard for Information technology - Telecommunications and Information exchange between systems - Local and metropolitan area network - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 5: Enhancements for Higher Throughput* [online]. New York : IEEE, 29.10.2009 [cit. 2011-05-11]. 502 s. Dostupné z WWW: <http://standards.ieee.org/getieee802/download/802.11n-2009.pdf>.

[16] ITU-R M.1225. *Guidelines Fof Evaluation Of Radio Transmission Technologie For IMT-2000*. [s.l.] : [s.n.], 1997 [cit. 2011-05-11]. 65 s. Dostupné z WWW: <http://my.com.nthu.edu.tw/~jmwu/LAB/ITU-R-Channel-Model.pdf>.

[17] PROKOPEC, Jan. *Systémy mobilních komunikací – přednáška č. 3: WLAN*. 29.11.2010 [cit. 2011-18-05]. 49 s. Prezentace. VUT v Brně, FEKT.

[18] HANUS, Stanislav. *Rádiové a mobilní komunikace – přednáška č. 2: Digitální modulace*. 2008 [cit. 2011-05-18]. 19 s. Prezentace. VUT v Brně, FEKT.

[19] *MathWorks* [online]. 25.4.2011 [cit. 2011-05-19]. Where can I obtain firmware for my USRP2. Dostupné z WWW: <http://www.mathworks.com/support/solutions/en/data/1-CUN7JZ/index.html?product=ML&solution=1-CUN7JZ>.

[20] *MathWorks* [online]. 23.8.2010 [cit. 2011-05-19]. How do I configure my computer to connect with USRP2 hardware. Dostupné z WWW: <http://www.mathworks.com/support/solutions/en/data/1-D2KBPZ/index.html?solution=1-D2KBPZ>.

Seznam zkratek

AWGN	Additive White Gaussioan Noise
BER	Bit Error Ratio
BPS	Bits Per Symbol
CP	Cyklický Prefix
DC	Direct Current
LMS	Least Mean Squares
LS	Least Square
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	OFDM Multiple Access
PLCP	Physical Layer Convergence Procedure
SNR	Signal-to-Noise Ratio