



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

**Fakulta elektrotechnická
Katedra radiotechniky**

UWB signály pro měření vzdáleností a určování polohy

The UWB signals for range measurement and position determination

Diplomová práce

Studijní program: Komunikace, multimédia a elektronika
Studijní obor: Bezdrátové komunikace

Vedoucí práce: prof. Ing. František Vejražka, CSc.

Bc. Zdeněk Fiala

Praha 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra elektromagnetického pole

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Zdeněk Fiala**

Studijní program: Komunikace, multimédia a elektronika
Obor: Bezdrátové komunikace

Název tématu: **UWB signály pro měření vzdáleností a určování polohy**

Pokyny pro vypracování:

Charakterizujte UWB signály a popište jejich aplikace se zaměřením na určování vzdálenosti a polohy.

Seznamte se s produktem DecaWave pro tyto účely. Vytvořte algoritmus určení polohy tzv. "tagu" v prostoru pokrytém signály 4 "kotev".

Navrhněte způsoby zpracování signálů pro různé aplikace uvažované k realizaci v projektu TE01020186.

Seznam odborné literatury:

- [1] Ultra-Wideband Positioning Systems.Theoretical Limits, Ranging Algorithms, and Protocols. ISBN-13 978-0-521-87309-3
- [2] IEEE Standard for Local and Metropolitan Area Networks.Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs).ISBN 978-0-7381-6684-1 STDPD97126

Vedoucí: prof. František Vejražka Ing., CSc.

Platnost zadání: LS 2016/2017



Prof. Ing. Pavel Pechač, Ph.D.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 4. 2. 2016

Čestné prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne:

Podpis:

Poděkování

Touto cestou bych rád poděkoval vedoucímu své diplomové práce prof. Ing. Františku Vejražkovi, CSc. za cenné rady jak charakteru technického, tak formálního. Dále bych rád poděkoval své rodině, bez které by práce mohla vzniknout jen stěží, za podporu psychickou i materiální.

Anotace

Tato práce si klade za cíl charakterizovat UWB signály a popsat jejich aplikace se zaměřením na určování vzdálenosti a polohy. V rámci teoretické části práce byla provedena analýza technologie UWB, která zahrnuje zejména popis UWB standardu, dále pak používané impulsy a vyzářovací limity. Teoretická část dále poukazuje na chybu určení vzdálenosti v případě průchodu signálu např. zdí. Pro účely praktické části práce bylo nutné se seznámit s produktem DecaWave (vývojovou sadou) EVK1000, který je navržen v souladu s doporučením IEEE 802.15.4a. Zdrojový kód k produktu EVK1000 byl dále upravován tak, aby bylo dosaženo co nejlepších výsledků s ohledem na určování polohy pohyblivého bodu, tzv. „tagu“. Další část práce je poté zaměřena na testování systému DecaWave a rozbor získaných výsledků. Práce dále zahrnuje popis aplikace pro zobrazování pozice „tagu“ v reálném čase. V závěru práce jsou diskutovány získané výsledky.

Klíčová slova

UWB, „tag“, DecaWave

Annotation

The thesis focuses to feature UWB signals and describe their applications regarding determination of distance and position. Within the theoretical part of the thesis I implemented an analysis of the UWB technology which includes in particular a description of the UWB standard as well as the impulses used and emission limits. The theoretical part of my thesis also points out the error in determining distance in the event of the signal passing through brickwork, etc. In order to implement the practical part of the thesis it was necessary to become acquainted with the DecaWave product (development kit) EVK1000 which is designed in accordance with the IEEE 802.15.4a recommendation. The source code for the product EVK1000 was further modified in order to achieve the best results as regards determining the position of the floating point, the so called “*tag*”. Another part of the thesis focuses on testing the system DecaWave and it also analyzes the results obtained. The thesis includes the description of an application for displaying the position of “the *tag*” in real time. At the end of the thesis the obtained results are discussed.

Keywords

UWB, “*tag*”, DecaWave

Obsah

| | | |
|--------|--|----|
| 1 | UWB technologie | 9 |
| 1.1 | Standard IEEE 802.15.4a..... | 9 |
| 1.1.1 | UWB PHY | 10 |
| 1.1.2 | UWB generátor | 13 |
| 2 | Používané impulsy UWB technologie | 16 |
| 2.1 | Gaussovské impulsy..... | 16 |
| 2.2 | Hermitovské polynomy..... | 18 |
| 2.3 | Ortogonální modifikované hermitovské impulsy | 18 |
| 3 | Analýza kanálové kapacity | 19 |
| 4 | FCC limity | 20 |
| 5 | Odhad polohy objektu pomocí UWB signálů..... | 21 |
| 5.1 | Úroveň přijatého signálu..... | 21 |
| 5.2 | Doba šíření signálu..... | 23 |
| 5.3 | Hyperbolická navigace..... | 24 |
| 5.4 | Určování polohy pomocí úhlů (AoA) | 24 |
| 6 | DecaWave EVK1000..... | 28 |
| 6.1 | Určování vzdálenosti pomocí metody „Two Way Ranging“ | 29 |
| 7 | Penetrace UWB signálu skrz překážku | 30 |
| 7.1 | Analýza relativní odchylky určení vzdálenosti..... | 30 |
| 7.2 | Odhad relativní permitivity vloženého materiálu | 33 |
| 8 | Výpočet odhadu pozice | 34 |
| 8.1 | Levenberg-Marquartův iterační algoritmus | 34 |
| 9 | Testování systému DecaWave..... | 35 |
| 9.1 | Časování | 36 |
| 9.2 | Komunikace s PC..... | 36 |
| 9.2.1 | Komunikace pomocí programu RealTerm | 36 |
| 9.2.2 | Komunikace pomocí vlastní C# aplikace | 37 |
| 9.3 | Princip fungování skriptu v programu Matlab R2014 b | 38 |
| 9.4 | Měření ve venkovním prostředí..... | 39 |
| 9.4.1 | Blízké okolí kotev | 40 |
| 9.4.2 | Vzdálenější okolí kotev | 42 |
| 9.4.3 | Trajektorie Technickou ulicí (Dejvice)..... | 43 |
| 9.5 | Měření ve vnitřním prostředí | 45 |
| 9.5.1 | Trajektorie 1 | 45 |
| 9.5.2 | Trajektorie 2 | 47 |
| 9.5.3 | Statický bod..... | 48 |
| 10 | C# aplikace..... | 53 |
| 10.1 | Kontextové menu „File“..... | 53 |
| 10.2 | Tlačítka aplikace | 54 |
| 10.2.1 | Tlačítko start..... | 55 |
| 10.2.2 | Tlačítko stop..... | 55 |
| 10.2.3 | Tlačítko „Turn Distance Extrapolation On/Off“ | 55 |
| 10.2.4 | Tlačítko „Turn Position Filtration On/Off“ | 56 |
| 10.3 | Textové pole..... | 57 |
| 10.4 | Graf..... | 57 |
| 10.5 | Příjem dat od sériové linky..... | 58 |
| 10.6 | CSML knihovna | 58 |
| 10.7 | Výpočet parametrů DOP..... | 59 |

| | | |
|----|--------------------------|----|
| 11 | Programové vybavení..... | 59 |
| 12 | Seznam příloh | 60 |
| 13 | Závěr..... | 63 |
| 14 | Seznam zkratek | 64 |
| 15 | Citovaná literatura..... | 64 |

1 UWB technologie

UWB signály jsou charakteristické mnohem větší šířkou pásma ve srovnání s konvenčními úzkopásmovými technologiemi. K definování UWB signálu je nejprve třeba uvést základní definici absolutní šířky pásma BW_{-10dB} signálu pro pokles o 10 dB:

$$BW_{-10dB} = f_{MAX} - f_{MIN} \quad (1)$$

Schematicky lze absolutní šířku pásma pozorovat na obr. 1. Kromě absolutní šířky pásma lze také definovat tzv. relativní šířku pásma, která je dána následujícím vztahem.

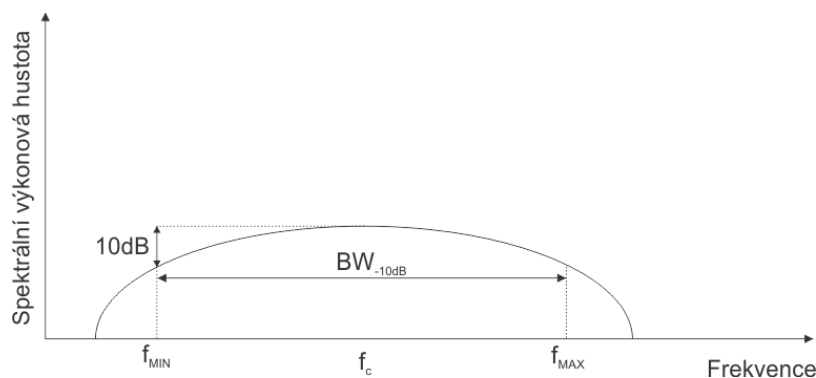
$$B_{rel} = \frac{BW_{-10dB}}{f_c} \quad (2)$$

Kde f_c má význam střední frekvence a lze ji určit pomocí vztahu:

$$f_c = \frac{f_{MAX} + f_{MIN}}{2} \quad (3)$$

Z uvedených vztahů poté vyplývá, že relativní šířka pásma může být vyjádřena pouze pomocí krajních frekvencí pro pokles o 10 dB, viz vztah:

$$B_{rel} = \frac{2(f_{MAX} - f_{MIN})}{f_{MAX} + f_{MIN}} \quad (4)$$

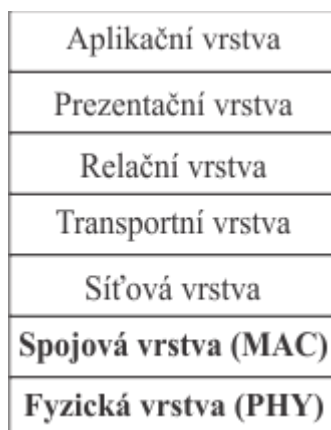


Obr. 1 - Znárodnění šířky pásma UWB signálu

Pokud je střední frekvence f_c větší než 2,5 GHz, spadá do kategorie UWB jakýkoli signál, který má absolutní šířku pásma pro pokles o 10 dB BW_{-10dB} větší než 500 MHz. Pokud je střední frekvence f_c menší než 2,5 GHz, musí mít signál relativní šířku pásma větší než 0,2, aby spadl do UWB kategorie, viz [1].

1.1 Standard IEEE 802.15.4a

Standard IEEE 802.15.4 definuje fyzickou (PHY) a MAC vrstvu sítí WPAN pro pevná a pohyblivá zařízení. Na obr. 2 lze pozorovat referenční model ISO/OSI s vyznačenými vrstvami (tučně), kterými se standard IEEE 802.15.4 zabývá.



Obr. 2 - Referenční model ISO/OSI

V dokumentu IEEE 802.15.4 [1] je fyzická vrstva definována pro různé druhy zařízení:

- Zařízení pracující v bezlicenčních pásmech (868–868,6 MHz, 902–928 MHz a 2400–2483,5 MHz)
- Zařízení s přesným určováním polohy, zvětšeným dosahem a s větší robustností a pohyblivostí (do této kategorie patří i rádiový modul DW1000 společnosti DecaWave, který je součástí produktu EVK1000)
- Zařízení pracující v souladu s čínskými regulacemi
- Zařízení pracující v japonských pásmech 950 – 956 MHz

1.1.1 UWB PHY

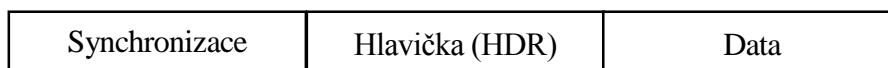
Norma IEEE 802.15.4a dále definuje různé druhy fyzické vrstvy. Jedním z typů je technologie nesoucí název UWB PHY, ve které mimo jiné pracuje systém DecaWave. Tato technologie kombinuje modulace BPM a BPSK. UWB PHY lze použít buď pro sub-gigahertzová pásma nebo pro pásma v rozsahu 3 až 10 GHz, viz [2].

| | Sub-gigahertzové pásmo | „Low band“ pásmo | „High band“ pásmo |
|--------------------------|-------------------------------|-------------------------|--------------------------|
| Frekvenční rozsah | 249,6 MHz až 749,6 MHz | 3,1 GHz až 4,8 GHz | 6 GHz až 10,6 GHz |
| Počet kanálů | 1 | 4 | 11 |

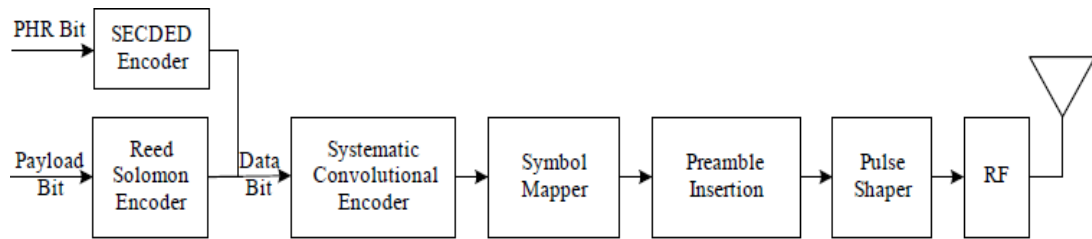
Tab. 1 - Frekvenční rozsahy a počty kanálů

1.1.1.1 Formát datové jednotky

Obr. 4 popisuje obecné schéma UWB vysílače. Z tohoto schématu vyplývá formát datových zpráv, které se skládají z preamble (SHR), hlavičky (HDR) a dat.



Obr. 3 - Formát datového rámce



Obr. 4 - Blokové schéma UWB vysílače [2], SECDED Encoder = SECDEC kódér, Reed Solomon Encoder = Reed Solomonův kódér, Systematic Convolutional Encoder = Systematický konvoluční kódér, Symbol Mapper = Mapování symbolů, Preamble Insertion = Vložení preamble, Pulse Shaper = Tvarovač impulsů

Formát datového rámce je znázorněn na obr. 3. Jako první je přenášena synchronizační posloupnost, pak hlavička a posléze zakódovaná data. Pokud některé z polí obsahuje více bajtů, je pro přenos použit systém „little endian“ (nejméně významný bajt – LSB – je přenášěn jako první). Hlavička je přenášena přenosovou rychlostí 850 kb/s, pokud jsou data přenášena rychlostí větší nebo rovnou 850 kb/s. V ostatních případech je hlavička přenášena rychlostí 110 kb/s. Data jsou modulována rychlostí, která je specifikována v hlavičce.

1.1.1.2 Preamble

Vzhledem k tomu, že délka kódu preamble a hodnoty PRF se mohou lišit, existuje také více přípustných hodnot doby trvání symbolu preamble. Tyto hodnoty jsou shrnuty v následující tabulce – čerpáno z [2].

| Délka kódu | PRF _{MAX} [MHz] | PRF _{AVG} [MHz] | Čipů/symbol | T _s [ns] | Rychlost [Msymbolů/s] |
|------------|--------------------------|--------------------------|-------------|---------------------|-----------------------|
| 31 | 31,2 | 16,10 | 496 | 993,59 | 1,01 |
| 127 | 124,8 | 62,89 | 508 | 1017,63 | 0,98 |

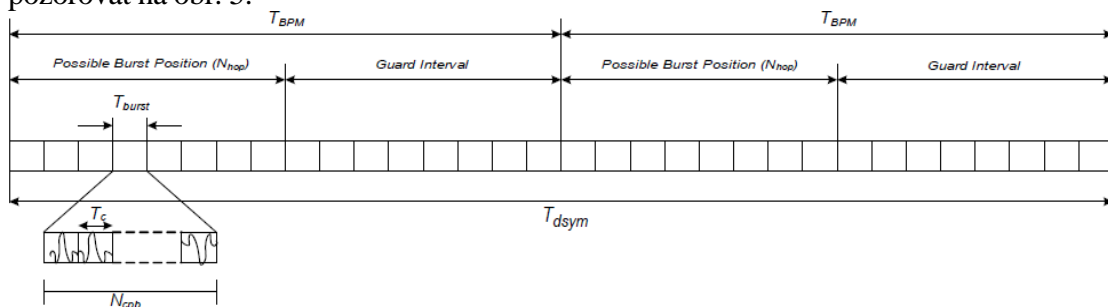
Tab. 2 - Vlastnosti pro různé délky kódu preamble

Střední hodnotu PRF lze získat pomocí vztahu, viz obr. 5:

$$PRF_{mean} = N_{cpb} / T_{dsym} \quad (5)$$

1.1.1.3 Struktura symbolu

Jak dále popisuje zdroj [2], použitá modulace UWB technologie BPM-BPSK umožňuje přenést dvoubitovou informaci na jedno slovo. Strukturu datového symbolu UWB lze pozorovat na obr. 5.



Obr. 5 - Struktura datového symbolu, převzato z [2], Possible Burst Position = Možné pozice „burst“ impulsu, Guard Interval = Ochranný interval

Každý symbol sestává z několika čipových pozic N_c . Každá čipová pozice má dobu trvání T_c . Doba trvání symbolu je poté dána jako $T_{dsym} = N_c T_c$. Každý symbol je rozdělen do 2 stejných intervalů $T_{BPM} = T_{dsym} / 2$, což umožňuje použít BPM modulaci. Impuls „burst“ se skládá z N_{cpb} čipových intervalů o délce trvání T_c . Doba trvání „burst“ impulsu je tím pádem dána rovnicí $T_{burst} = N_{cpb} T_c$. „Burst“ impuls může být umístěn buď v první, nebo v druhé polovině symbolu. Pomocí umístění „burst“ impulsu je přenášena jednobitová informace. Pomocí fáze „burst“ impulsu $\in \{1, -1\}$ je přenášen druhý bit. V každém datovém symbolu by měl být přenesen jeden „burst“ impuls. Díky skutečnosti, že platí nerovnice $T_{dsym} \gg T_{burst}$, lze potlačit interference v případě mnohonásobného přístupu pomocí časového skákání. Ve snaze zamezit mezisymbolové interferenci je v symbolu obsažen také ochranný interval. Detailnější informace jsou obsaženy v [2].

1.1.1.4 Modulace technologie UWB

Radiová vlna přenášená éterem může být vyjádřena pomocí vztahu (viz [2]):

$$x^{(k)}(t) = (1 - 2g_1^{(k)}) \sum_{n=1}^{N_{cpb}} (1 - 2s_{n+kN_{cpb}}) \times p(t - g_0^{(k)}T_{BPM} - h^{(k)}T_{burst} - nT_c) \quad (6)$$

Symbol na k -té pozici nese dvoubitovou informaci $g_0^{(k)}$ a $g_1^{(k)} \in \{0, 1\}$. Bit $g_0^{(k)}$ ovlivňuje pozici impulsu, bit $g_1^{(k)}$ mění jeho polaritu. Sekvence $s_{n+kN_{cpb}} \in \{0, 1\}$, $n = 0, 1, \dots, N_{cpb} - 1$ je scamblovací kód použitý pro k -tý symbolový interval. Tato sekvence slouží k vylepšení spektrálních vlastností přenášené rádiové vlny. $h^{(k)} \in \{0, 1 - N_{hop} - 1\}$ je tzv. „burst hopping“ pozice. Sekvence $h^{(k)}$ umožňuje potlačit interference v případě většího počtu uživatelů, $p(t)$ je výsledný přenášený tvar impulsu.

1.1.1.5 Používaná frekvenční pásma

Technologie UWB definuje celkem 4 kanály „low band“ pásma, 11 kanálů v „high band“ pásmu a 1 sub-gigahertzový kanál. V každé skupině kanálů je definován jeden povinný kanál – v tabulce jest vyznačen tučně. Pro vytvoření tabulky byl použit zdroj [2].

| Skupina [-] | Číslo kanálu [-] | Nosná frekvence [MHz] | Šířka pásma [MHz] |
|-------------|------------------|-----------------------|-------------------|
| 0 | 0 | 499,2 | 499,20 |
| 1 | 1 | 3494,4 | 499,20 |
| | 2 | 3993,6 | 499,20 |
| | 3 | 4492,8 | 499,20 |
| | 4 | 3993,6 | 1331,20 |
| 2 | 5 | 6489,6 | 499,20 |
| | 6 | 6988,8 | 499,20 |
| | 7 | 6489,6 | 1081,60 |
| | 8 | 7488,0 | 499,20 |
| | 9 | 7987,2 | 499,20 |
| | 10 | 8486,4 | 499,20 |

| | | | |
|--|----|--------|---------|
| | 11 | 7987,2 | 1331,20 |
| | 12 | 8985,6 | 499,20 |
| | 13 | 9484,8 | 499,20 |
| | 14 | 9984,0 | 499,20 |
| | 15 | 9484,8 | 1354,97 |

Tab. 3 - Rozdělení kanálů

1.1.2 UWB generátor

V programu Matlab R2014b byl vytvořen skript, který generuje UWB signál v souladu se standardem IEEE 802.15.4a dle obr. 4 a vztahu (6).

1.1.2.1 Reed-Solomonův kodér

Vstupní bitová posloupnost je kódována pomocí RS kodéru s Galoisovým polem $GF(2^6)$, jehož primitivní polynom má tvar $1 + x + x^6$. Kódovací proces funguje následujícím způsobem (viz [2]):

1. Přidání nulových bitů – vstupní posloupnost je doplněna na začátku nulami tak, aby celková délka byla 330 bitů. Posloupnost bude mít tvar $\{d_0, d_1, \dots, d_{329}\}$, kde d_0 je první bit v čase.
2. Konverze bitové posloupnosti na symbolovou – posloupnost 330 bitů je převedena na 55 RS symbolů $\{D_0, D_1, \dots, D_{54}\}$ pomocí vztahu:

$$D_k = 2^5 d_{6k+5} + 2^4 d_{6k+4} + 2^3 d_{6k+3} + 2^2 d_{6k+2} + 2d_{6k+1} + d_{6k}, \quad (7)$$

kde index k se mění v rozsahu od 0 do 54.

3. Kódování – takto vytvořené symboly jsou kódovány systematickým kodérem $RS_6(63,55)$ (v programu Matlab R2014b řešeno pomocí funkce *rsenc*). Výstup kodéru má tvar:

$$U_k = \begin{cases} D_k & (k = 0, 1, \dots, 54) \\ P_k & (k = 55, 56, \dots, 62) \end{cases}, \quad (8)$$

kde P_k má význam parity.

4. Zpětná konverze do bitové posloupnosti – výstupní symboly $\{U_0, U_1, \dots, U_{62}\}$ jsou zpětně převedeny do bitové posloupnosti $\{u_0, u_1, \dots, u_{337}\}$.
5. Odstranění nulových bitů – nulové bity přidány na začátek posloupnosti jsou odstraněny.

1.1.2.2 Vložení hlavičky PHR

V dalším kroku je nutné provést vložení hlavičky, která se skládá celkem z 19 bitů.

| | | | | | | | | | | | | | | | | | | |
|-----|-----|----|----|----|----|----|----|----|-----|-----|-----|-----|----|----|----|----|----|----|
| Rd1 | Rd0 | L6 | L5 | L4 | L3 | L2 | L1 | L0 | Rng | Ext | Pt1 | Pt0 | C5 | C4 | C3 | C2 | C1 | C0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

Obr. 6 - Hlavička (PHR), převzato z [2]

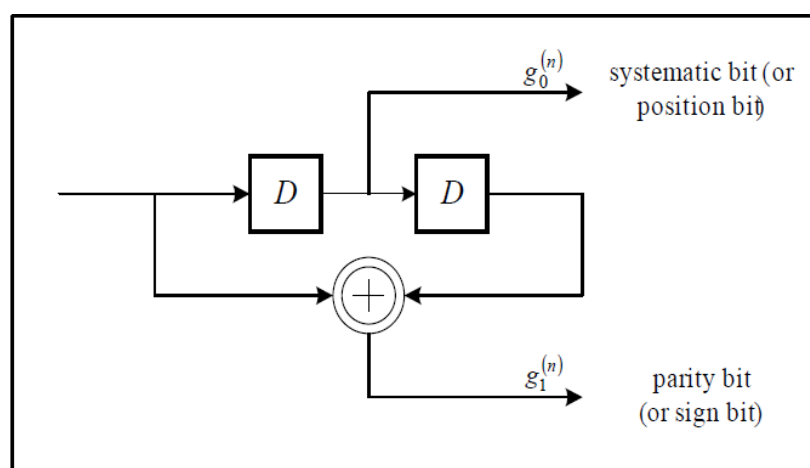
Bity $Rt1$ a $Rt0$ mají význam přenosové rychlosti. Bity L mají význam délky rámce. Rng bit určuje, zda se jedná o R-frame (*Ranging-frame*). V případě použití metody „Two Way Ranging“ je tento bit nastaven na hodnotu 1. Bit Ext určuje, zda se jedná o rozšířenou hlavičku. Bity P určují délku synchronizační posloupnosti a bity C jsou tzv. SECDED bity – jedná se o kontrolní posloupnost.

1.1.2.3 Konvoluční kodér

Hlavička je vložena na začátek výstupu RS kodéru a na konec posloupnosti jsou vloženy 2 nulové tzv. „tail“ bity (pro vyčištění kodéru). Tato posloupnost je dále kódována pomocí konvolučního kodéru viz obr. 7.

Lze pozorovat, že $R = 1/2$. Výstupy kodéru lze matematicky popsat následujícím způsobem, viz [2]:

$$\begin{aligned} g_0 &= [010]_2 \\ g_1 &= [101]_2 \end{aligned} \quad (9)$$



Obr. 7 - Konvoluční kodér (převzato z [2])

1.1.2.4 UWB scrambler

Bitové posloupnosti g_0 a g_1 poté vstupují do scrambleru, který generuje posloupnost $s_{n+kN_{cpb}}$ a $h^{(k)}$. Polynomiální reprezentace scrambleru je dána následujícím vztahem:

$$g(D) = 1 + D^{14} + D^{15} \quad (10)$$

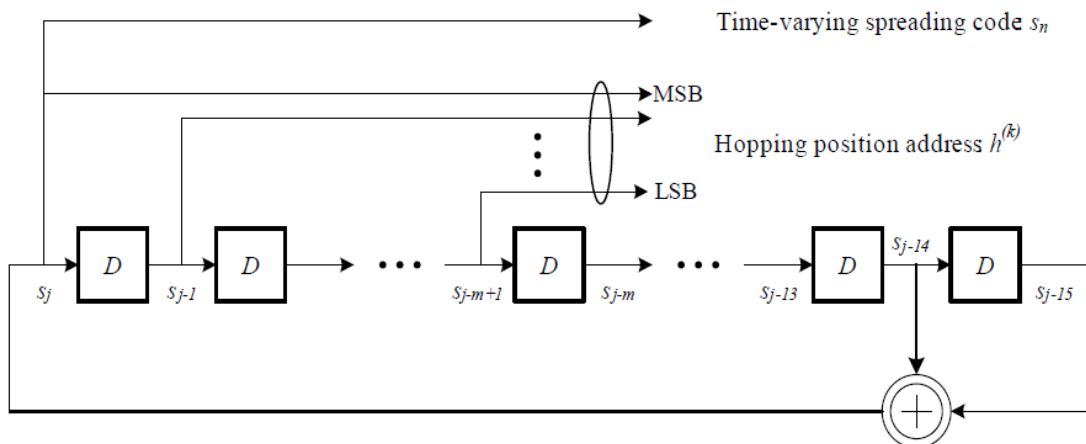
Tomu odpovídá výstupní rovnice

$$s_n = s_{n-14} \oplus s_{n-15}, \quad (11)$$

kde operátor \oplus má význam operace sčítání modulo 2. Tento scrambler může být realizován například pomocí LFSR registru. Po průchodu PHR scramblerem není registr restartován, viz [2]. LFSR registr je zobrazen na obr. 8.

Inicializace LFSR registru se provádí pomocí kódu preamble. Kód preamble může sestávat buď z 31, nebo 127 členů. Kód preamble je sekvence z ternární abecedy $\{-1, 0, 1\}$. Celkem existuje 17 různých kódů preamble. Z kódu preamble je inicializační stav LFSR registru získán tak, že jsou odstraněny všechny 0 a všechny -1 jsou nahrazeny 0. Prvních 15 bitů takto vzniklé (již binární) posloupnosti je poté nahráno do LFSR registru, viz [2].

Během k symbolového intervalu je LSFR taktován $N_{cpb} \times$.



Obr. 8 - LSFR registr, převzato z [2]

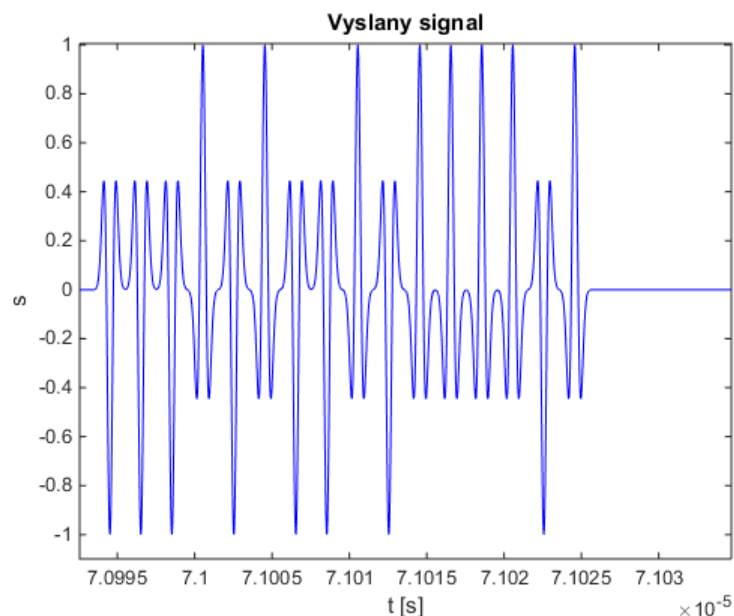
Pozice posloupnosti impulsů v časové oblasti označovaná jako „burst hopping position“ $h^{(k)}$ je poté určena pomocí vztahu

$$h^{(k)} = 2^0 s_{kN_{cpb}} + 2^1 s_{1+kN_{cpb}} + \dots + 2^{m-1} s_{m-1+kN_{cpb}}, \quad (12)$$

kde $m = \log_2 N_{hop}$. Tím jsou určeny všechny proměnné vztahu (6).

1.1.2.5 Vyslaný signál

Část vyslaného signálu vygenerovaného programem Matlab R2014b je zobrazena na obr. 9.



Obr. 9 - Příklad UWB signálu

Jako UWB impuls byla použita druhá derivace Gaussova impulsu (bude popsáno dále).

2 Používané impulsy UWB technologie

Mezi nepoužívanější typy vysílaných impulsů UWB technologie patří gaussovské či ortogonální modifikované hermitovské impulsy. Některé jejich vlastnosti včetně matematického popisu budou ozřejměny v následujících podkapitolách.

2.1 Gaussovské impulsy

Jedním ze základních typů používaných UWB impulsů jsou impulsy Gaussovy. Pro přenos informace se používají různé derivace Gaussova impulsu, jehož definice se mírně liší napříč prameny. Lze ho definovat např. podle následujícího vztahu, viz [3].

$$g(t) = \frac{A}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}} \quad (13)$$

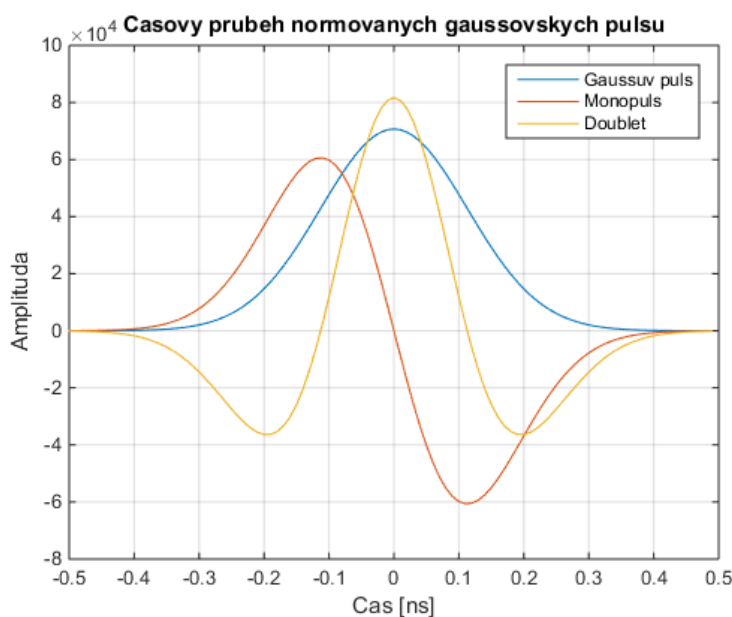
První derivace Gaussova impulsu se poté označuje jako „monopuls“. Tento je dán vztahem níže.

$$g^{(1)}(t) = -\frac{At}{\sqrt{2\pi}\sigma^3} e^{-\frac{t^2}{2\sigma^2}} \quad (14)$$

Druhá derivace Gaussova impulsu je označována jako „doublet“ a je dána rovnicí:

$$g^{(2)}(t) = A\left(\frac{t^2}{\sqrt{2\pi}\sigma^5} - \frac{1}{\sqrt{2\pi}\sigma^3}\right)e^{-\frac{t^2}{2\sigma^2}} \quad (15)$$

Jednotlivé průběhy gaussovských impulsů lze pozorovat na obr. 10. Impulsy byly normovány na jednotkovou energii, případně byl otočen jejich smysl tak, aby nejvyšší část impulsu byla kladná. Šířka „doubletu“ byla zvolena 1 ns. K vygenerování impulsů byl použit program Matlab R2014b.

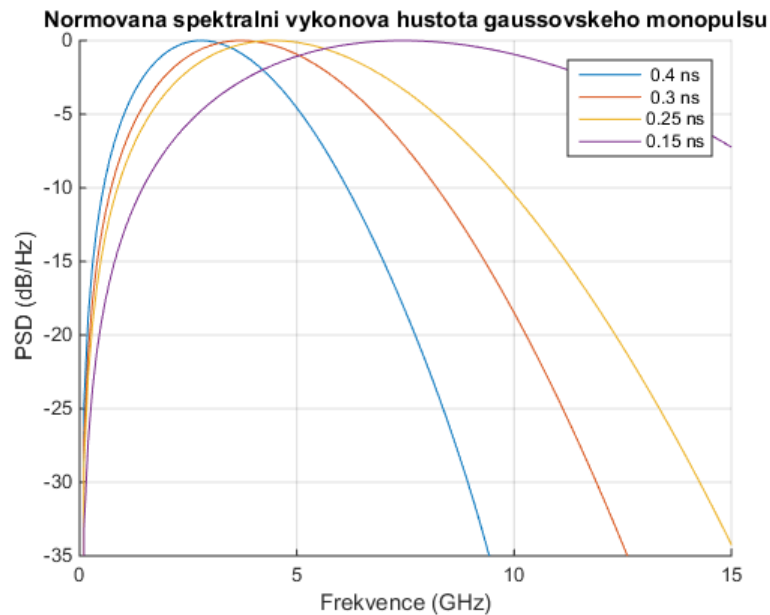


Obr. 10 – Normované gaussovské impulsy

Na obr. 11 lze poté pozorovat spektrální výkonovou hustotu monocyklů pro různé šířky impulsů. Šířka Gaussova monocyklu byla určena podle přibližného vztahu, viz [3]:

$$Tp \approx 7\sigma \quad (16)$$

Jelikož je „monocykl“ ve skutečnosti nekonečně široký, byla šířka definována tak, aby v jejím intervalu leželo 99,99% energie impulsu, viz [3]. Normování bylo provedeno na každý impuls zvlášť.



Obr. 11 – PSD „monocyklů“ různých šířek

Lze pozorovat efekt zúžení frekvenčního spektra pro širší impulsy v časové oblasti. To vyplývá z vlastností Fourierovy transformace. Fourierův obraz Gaussova impulsu je dán následujícím vztahem, viz [4] a [3]:

$$\text{FT}\{g(t)\} = Ae^{-\frac{1}{2}(2\pi\sigma f)^2} \quad (17)$$

Fourierovým obrazem Gaussova impulsu je tedy opět Gaussův impuls. K analytickému vyjádření obrazů vyšších derivací Gaussových impulsů lze využít větu o obrazu derivace signálu, která nám říká, že:

$$\text{FT}\left\{\frac{d}{dt}s(t)\right\} = j\omega S(\omega) \quad (18)$$

Fourierův obraz Gaussových impulsů vyšších řádů lze tedy získat ze vztahu, viz [3]:

$$\text{FT}\left\{\frac{d^n}{dt^n}g(t)\right\} = A(j2\pi f)^n e^{-\frac{1}{2}(2\pi\sigma f)^2} \quad (19)$$

Kde n je řád derivace. Použití vyšších derivací Gaussova impulsu je vhodné, jelikož tyto pulsy lépe respektují FCC limity.

2.2 Hermitovské polynomy

Hermitovské polynomy vynalezl francouzský matematik, který se jmenoval Charles Hermite (1822 až 1901). Polynom je konečná suma příspěvků typu $\alpha_k x^k$, kde k je nezáporné celé číslo. Hermitovský polynom je pak generován s pomocí vztahu níže, viz [5]

$$h_{e_n}(t) = (-\tau)^n e^{\frac{t^2}{2\tau^2}} \frac{d^n}{dt^n} \left(e^{-\frac{t^2}{2\tau^2}} \right) \quad (20)$$

kde n je nezáporné celé číslo a t leží v prostoru reálných čísel. V [5] je poté obsaženo prvních 9 hermitovských polynomů. Pro tvorbu UWB signálů může být však podstatná ortogonalita systému, z tohoto důvodu se používají tzv. ortogonální modifikované hermitovské impulsy, viz [5].

2.3 Ortogonální modifikované hermitovské impulsy

Definice ortogonalit říká, že na intervalu $a \leq t \leq b$ jsou funkce g_m a g_n ortogonální, pokud platí následující vztah.

$$\int_a^b g_m(t) g_n^*(t) dt = 0, \text{ pro } m \neq n \quad (21)$$

Hermitovské polynomy ortogonální systém netvoří. Pro dosažení ortogonalit je však lze modifikovat podle následujícího vztahu, viz [5].

$$h_n(t) = k_n e^{-t^2/4\tau^2} h_{e_n}(t) \quad (22)$$

V [5] jsou pak dále uvedeny ortogonální modifikované hermitovské impulsy pro $0 \leq n \leq 8$. Ortogonální systém prvních 9 modifikovaných hermitovských impulsů má tedy následující tvar.

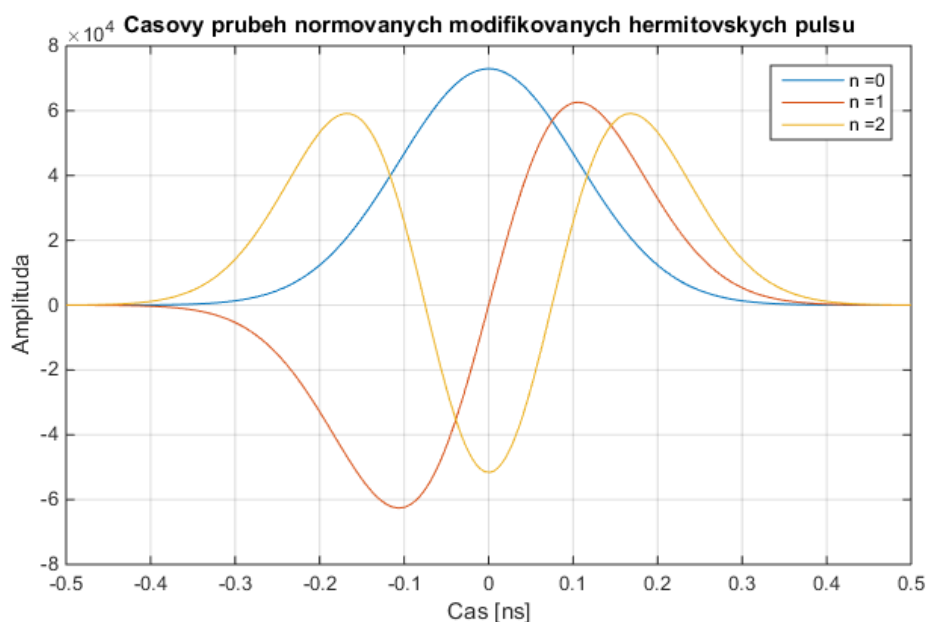
$$\begin{aligned} h_0(t) &= k_0 e^{-t^2/4\tau^2} \\ h_1(t) &= k_1 \frac{t}{\tau} e^{-t^2/4\tau^2} \\ h_2(t) &= k_2 \left[\frac{t^2}{\tau^2} - 1 \right] e^{-t^2/4\tau^2} \\ h_3(t) &= k_3 \left[\frac{t^3}{\tau^3} - 3 \frac{t}{\tau} \right] e^{-t^2/4\tau^2} \\ h_4(t) &= k_4 \left[\frac{t^4}{\tau^4} - 6 \frac{t^2}{\tau^2} + 3 \right] e^{-t^2/4\tau^2} \\ h_5(t) &= k_5 \left[\frac{t^5}{\tau^5} - 10 \frac{t^3}{\tau^3} + 15 \frac{t}{\tau} \right] e^{-t^2/4\tau^2} \end{aligned} \quad (23)$$

$$h_6(t) = k_6 \left[\frac{t^6}{\tau^6} - 15 \frac{t^4}{\tau^4} + 45 \frac{t^2}{\tau^2} - 15 \right] e^{-t^2/4\tau^2}$$

$$h_7(t) = k_7 \left[\frac{t^7}{\tau^7} - 21 \frac{t^5}{\tau^5} + 105 \frac{t^3}{\tau^3} - 105 \frac{t}{\tau} \right] e^{-t^2/4\tau^2}$$

$$h_8(t) = k_8 \left[\frac{t^8}{\tau^8} - 28 \frac{t^6}{\tau^6} + 210 \frac{t^4}{\tau^4} - 420 \frac{t^2}{\tau^2} + 105 \right] e^{-t^2/4\tau^2}$$

Konstanty k_n určují energii impulsu. V programu Matlab bylo však provedeno normování na jednotkovou energii stejně jako u gaussovských impulsů. Časový průběh impulsů prvních tří řádů lze pozorovat na obr. 12.



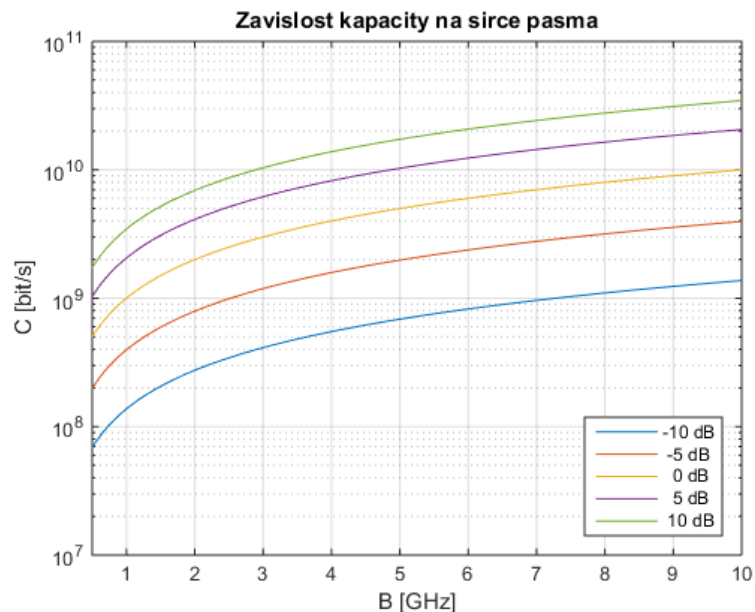
Obr. 12 - Hermitovské impulsy

3 Analýza kanálové kapacity

Vzhledem k tomu, že UWB technologie využívá poměrně velkou šířku pásma, lze očekávat i poměrně velké kapacitní možnosti celého systému. Za předpokladu, že se jedná o AWGN kanál a že kódová kniha na straně vysílače je gaussovská, platí vztah [6]:

$$C = B \log_2(1 + SNR) \quad (24)$$

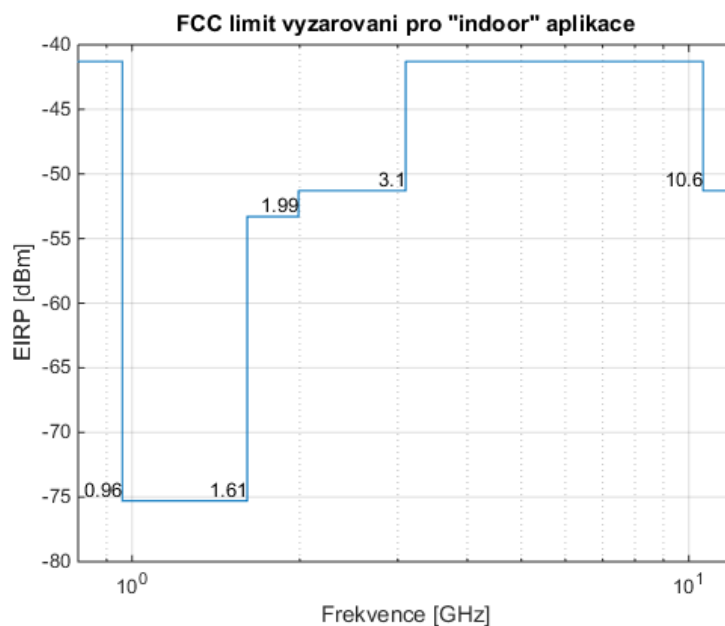
Kde B je šířka pásma v Hz a SNR je odstup signálu od šumu. Graf závislosti kanálové kapacity na šířce pásma je zobrazen na obr. 13. Jednotlivé křivky se od sebe liší různou hodnotou SNR , viz legenda vpravo dole.



Obr. 13 - Zavislost kapacity na širce pásma

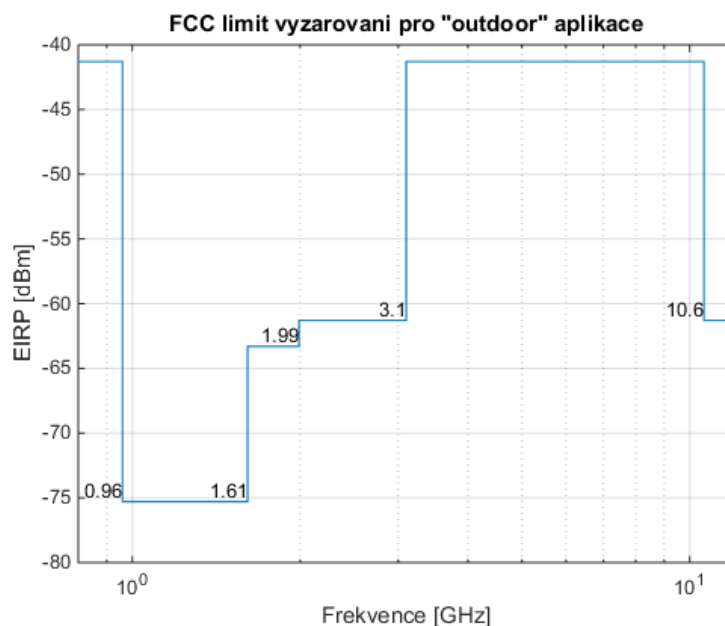
4 FCC limity

Vzhledem ke skutečnosti, že UWB se vyznačuje svou širokopásmovostí, dochází ve frekvenční oblasti k interferenci s ostatními technologiemi. Je proto nutné stanovit limity vyzařování, které jsou definovány FCC regulacemi.



Obr. 14 – „Indoor“ limit

V rámci těchto regulací jsou definovány různé limity vyzařování pro „indoor (obr. 14)“ a „outdoor“ (viz [1]) aplikace. „Outdoor“ limit má v některých pásmech o 10 dB přísnější mez než limit pro „indoor“ aplikace.



Obr. 15 – „Outdoor“ limit

5 Odhad polohy objektu pomocí UWB signálů

K odhadu polohy lze využít různých parametrů přijatého signálu. V dalších sekcích budou analyzovány metody založené na měření přijatého výkonu, časových parametrů signálu nebo směru příchodu signálu.

5.1 Úroveň přijatého signálu

Mezi základní techniky určování vzdálenosti patří metoda založená na úrovni přijatého signálu. Využívá se zde skutečnosti, že úroveň signálu při zvětšování vzdálenosti mezi přijímačem a vysílačem klesá. Za předpokladu, že přijímači je znám vysílací výkon, může být proveden odhad vzdálenosti (přijímač – vysílač).

Nárůst ztrát elektromagnetické vlny lze vyjádřit n mocninou vzdálenosti, čemuž odpovídá tzv. empirický model, viz [7].

$$P_p \approx \frac{1}{d^n} \quad (25)$$

Kde P_p je přijímaný výkon a d je vzdálenost mezi přijímačem a vysílačem. Použitelnost tohoto modelu je limitována tím, že se v rozsahu analyzovaných vzdáleností nesmí nacházet Fresnelův zlom, viz [7]. Pokud je tento předpoklad splněn, platí vztah, viz [8]:

$$\bar{P}(d) = P_0(d_0) + 10n \log \frac{d}{d_0} \quad (26)$$

Kde \bar{P} je přijímaný výkon signálu ve vzdálenosti d , P_0 je přijímaný výkon v referenční vzdálenosti d_0 a n je tzv. spádový koeficient, viz [7] a [8].

Velikost ztrát šířením ovlivňují zejména dva jevy. Prvním z nich je vícecestné šíření. Tento jev lze značně eliminovat průměrováním přes dostatečně dlouhou dobu T , viz [8].

$$\bar{P}(d) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |s(t)|^2 dt \quad (27)$$

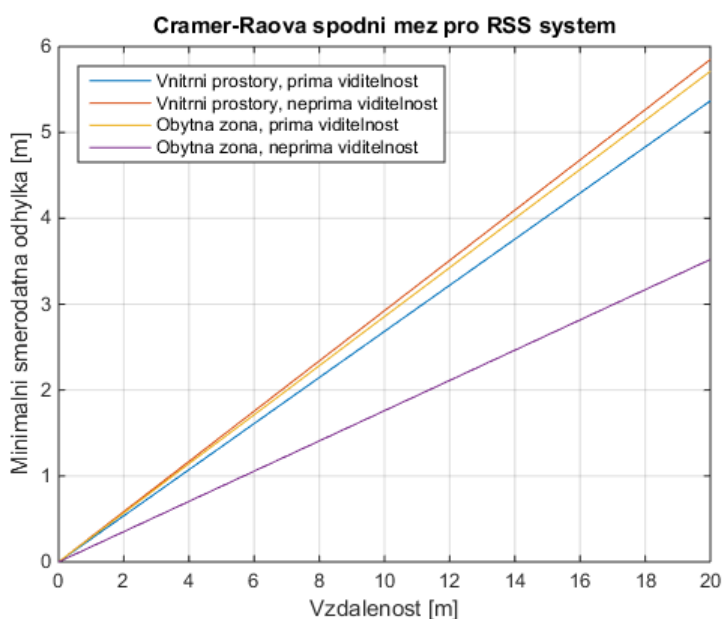
Dalším významným faktorem může být značná nehomogenita prostředí, v našem případě např. výskyt zdí s různou dielektrickou konstantou. Okamžitý přijatý výkon P tak bude fluktuovat, což můžeme modelovat pomocí následujícího vztahu, viz [8]:

$$10 \log P(d) \sim N(\bar{P}(d), \sigma_{sh}^2) \quad (28)$$

Kde \bar{P} je průměrný přijatý výkon a σ_{sh}^2 je rozptyl náhodné proměnné N s gaussovským rozložením, viz [8]. Pro tento model lze vyjádřit tzv. Cramer-Raovu spodní mez (CRLB²) následujícím vztahem, viz [8], [9] a [10].

$$\sqrt{\text{Var}\{\hat{d}\}} \geq \frac{\ln 10}{10} \frac{\sigma_{sh}}{n} d \quad (29)$$

Kde \hat{d} je nevychýlený odhad vzdálenosti d a n je tzv. spádový koeficient dle rovnice (18).



Obr. 16 - CRLB pro RSS systém

Na obr. 16 lze pozorovat CRLB pro různé typy prostředí. V [11] je obsaženo několik základních typů kanálů pro různá prostředí.

| | Vnitřní prostory | | Obytná zóna | |
|-------------------------------------|-------------------|---------------------|-------------------|---------------------|
| | Přímá viditelnost | Nepřímá viditelnost | Přímá viditelnost | Nepřímá viditelnost |
| n [-] | 1,63 | 3,07 | 1,79 | 4,58 |
| σ_{sh} [m] | 1,90 | 3,90 | 2,22 | 3,51 |

¹ Index „sh“ byl použit z toho důvodu, že v anglosaských literaturách se tento jev označuje jako „shadowing“.

² Zkratka pochází z anglického označení **C**ramer-**R**ao **L**ower **B**ound.

5.2 Doba šíření signálu

Určování odhadu vzdálenosti mezi přijímačem a vysílačem na základě měření doby šíření signálu vyžaduje časovou synchronizaci. Po provedení měření doby šíření signálu lze říci, že určovaný bod leží na kulové ploše (v dvoudimenzionálním případě na kružnici) o poloměru d :

$$d = c\tau \quad (30)$$

Kde c je velikost rychlosti šíření signálu (světla) v daném prostředí a τ je určená doba šíření signálu, viz [8]. Signál y na straně přijímače lze reprezentovat vztahem, viz [8]:

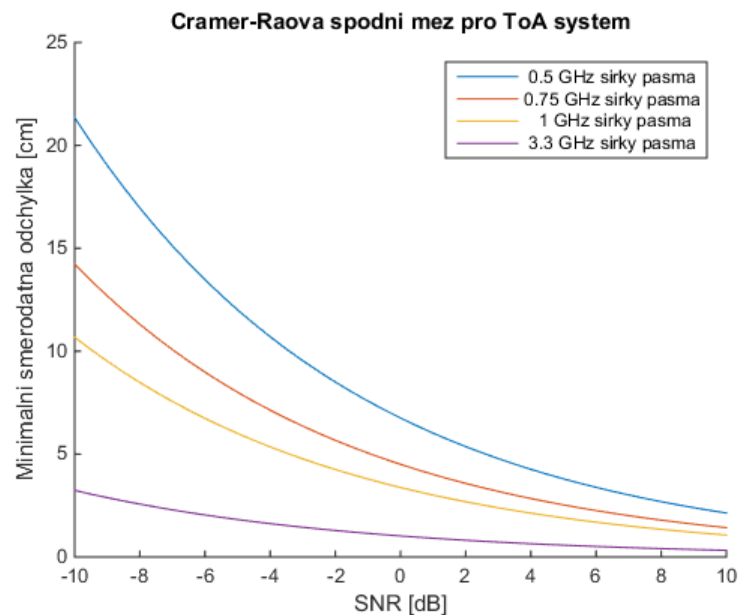
$$y(t) = \alpha x(t - \tau) + w(t) \quad (31)$$

Kde α reprezentuje všechny zesilující/tlumící prvky mezi vysílačem a přijímačem, x je vyslaný signál a w reprezentuje bílý gaussovský šum se spektrální výkonovou hustotou $\frac{N_0}{2}$, viz [1]. Je třeba však podotknout, že tento vztah je poměrně zjednodušený. V obvyklých aplikacích je vhodné do vstupně-výstupního vztahu zahrnout také vícecestné šíření. Pro tento model lze CRLB vyjádřit vztahem, viz [8] a [1]:

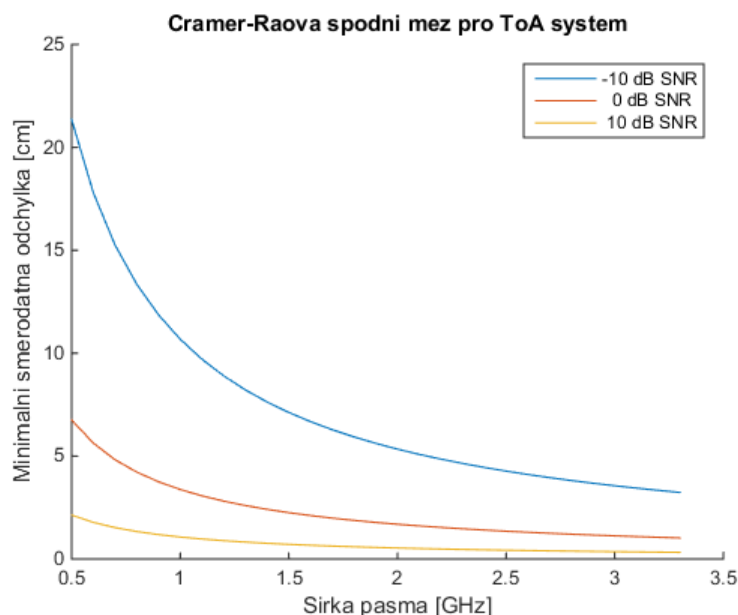
$$\sqrt{\text{Var}\{\hat{\tau}\}} \geq \frac{1}{2\sqrt{2\pi}\sqrt{\text{SNR}\beta}} \quad (32)$$

Kde $\hat{\tau}$ je odhadovaná doba šíření signálu, SNR je odstup signálu od šumu a β je tzv. efektivní šířka pásma.

Lze pozorovat, že ve srovnání s variantou RSS systému je zde značná závislost na šířce pásma. ToA systém může dosahovat poměrně malých odchylek – desítky cm, viz obr. 17 a obr. 18.



Obr. 17 – CRLB pro ToA systém (šířka pásma vynesena parametricky)



Obr. 18 - CRLB pro ToA systém (odstup signálu od šumu vyneseno parametricky)

5.3 Hyperbolická navigace

Princip hyperbolické navigace spočívá v určení rozdílu příchodu signálů od různých majáků (vyslaných ve stejnou dobu). Majáky musí být synchronizovány, pohyblivý bod nikoli. Tato metoda se v anglosaských literaturách označuje zkratkou TDoA. Časový rozdíl příchodu signálů od různých majáků je dán vztahem, viz [8]:

$$\tau_{TDOA} = \tau_1 - \tau_2 \quad (33)$$

Po vynásobení velikostí rychlosti světla v daném prostředí obdržíme hyperbolu, na které leží určovaný bod. Efekt šířky pásma je zde stejný jako v případě ToA systému, viz [8].

Druhá možnost, jak provést odhad rozdílu příchodu signálů, spočívá ve výpočtu korelační funkce, viz [12].

$$R_{12}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} s_1(t)s_2(t + \tau)dt \quad (34)$$

Kde s_1 a s_2 jsou přijaté signály a T je doba, po kterou pohyblivý bod naslouchá. Určení odhadu TDoA poté spočívá v hledání maxima korelační funkce přes všechny posuny τ , viz [8]:

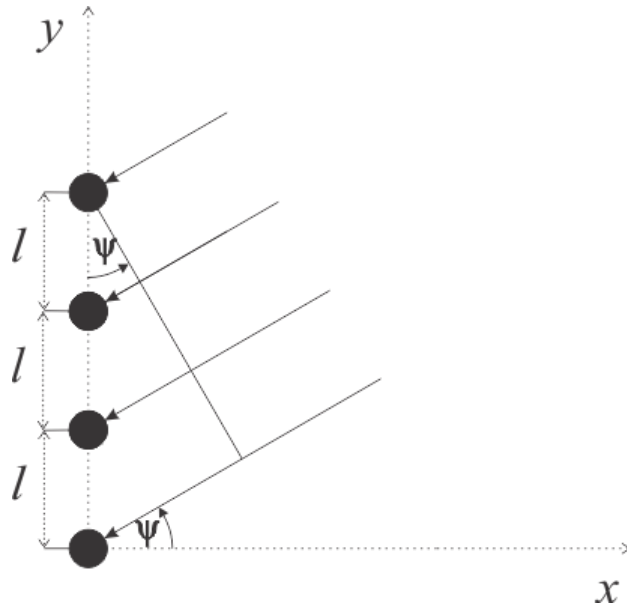
$$\hat{\tau}_{TDOA} = \arg \max_{\tau} |R_{12}(\tau)| \quad (35)$$

Úspěšnost této metody však značně klesá v případě výskytu vícecestného šíření, viz [8].

5.4 Určování polohy pomocí úhlů (AoA)

Tato metoda poskytuje údaj o směru, ve kterém se kýžený bod nachází. Výslednou polohu lze pak získat jako průsečík přímk. Princip metody lze pozorovat na obr. 19. K určení úhlu

Lze využít například anténní řadu (také obr. 19), kde úhel ψ lze určit z časového zpoždění přichozícího signálu na jednotlivých anténách.



Obr. 19 - Princip metody založené na určení úhlu pomocí anténní řady

Pokud je určovaný bod v dostatečné vzdálenosti (teoreticky platí pouze pro nekonečnou vzdálenost), lze dopadající vlnu označit za rovinnou, což vede ke značnému zjednodušení matematického popisu celé situace (obr. 19).

Časový rozdíl Δt příchodu signálu mezi sousedními anténními elementy lze vyjádřit pomocí následujícího vztahu

$$\Delta t = \frac{l \sin \psi}{c} \quad (36)$$

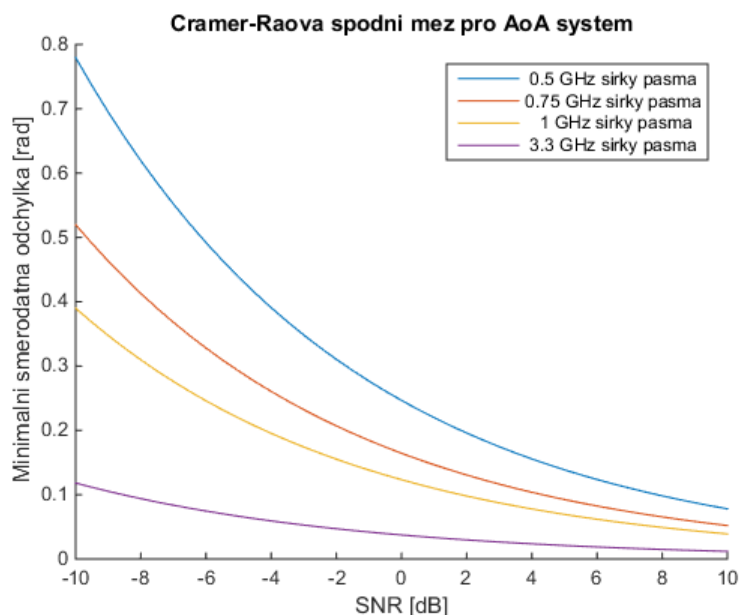
kde c je velikost rychlosti světla v daném prostředí a význam ostatních proměnných je patrný z obr. 19. Pro model kanálu lze zvolit obdobný vztah jako v případě měření doby šíření signálu

$$y_i(t) = \alpha x(t - \tau_i) + w_i(t) \quad (37)$$

kde $i = 1 \dots N_a$ má význam indexu antény (N_a je počet přijímačů). Ostatní prvky již byly popsány v předchozích kapitolách.

Pro takovýto model, za předpokladu nezávislosti šumu na jednotlivých anténách, lze vyjádřit CRLB podle následujícího vztahu, viz [8]:

$$\sqrt{\text{Var}\{\hat{\psi}\}} = \frac{c\sqrt{3}}{\sqrt{2\pi}\sqrt{\text{SNR}}\beta\sqrt{N_a(N_a^2 - 1)l \cos \psi}} \quad (38)$$

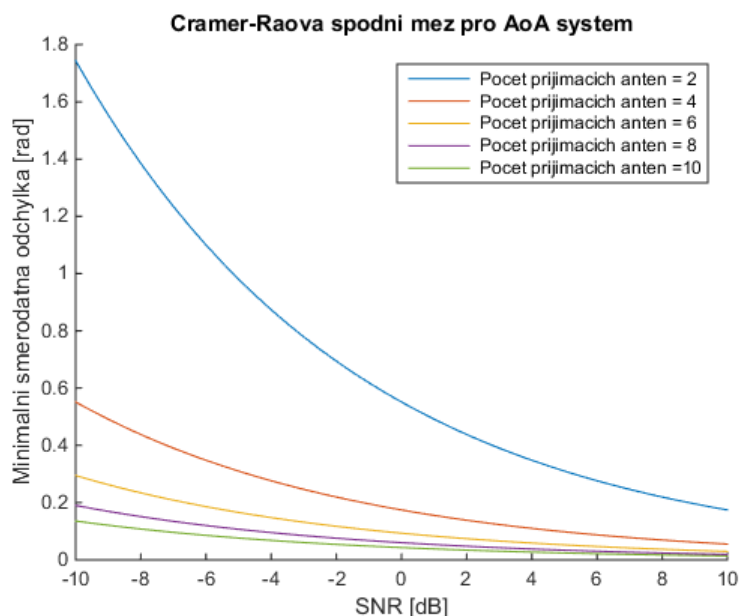


Obr. 20 - CRLB pro AoA systém (vliv šířky pásma)

Na obr. 20 lze pozorovat vliv odstupů signálu od šumu na minimální směrodatnou odchylku pro různé hodnoty efektivní šířky pásma (vyneseno parametricky). Jedná se o obdobnou závislost jako pro ToA systém. Simulace byla provedena pro hodnoty popsané následující tabulkou:

| N_a [-] | ψ [°] | l [m] |
|-----------|------------|---------|
| 5 | 30 | 0.1 |

Tab. 5 - Parametry simulace AoA systému



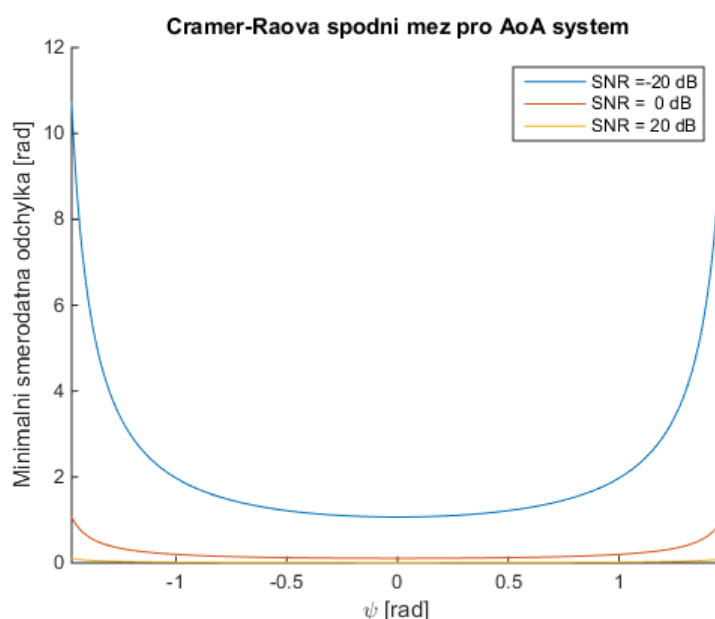
Obr. 21 - CRLB pro AoA systém (vliv počtu přijímacích antén)

Na obr. 21 lze poté pozorovat také vliv odstupů signálu od šumu na minimální směrodatnou odchylku, zde ovšem pro různé počty přijímacích antén při fixní efektivní šířce pásma. Parametry simulace popisuje následující tabulka:

| β [GHz] | ψ [°] | l [m] |
|---------------|------------|---------|
| 1 | 30 | 0.1 |

Tab. 6 - Parametry simulace AoA systému (fixní efektivní šířka pásma)

Následující 2 obrázky byly pak věnovány analýze vlivu hodnoty úhlu příchodu signálu Ψ na přesnost jejího měření. Z uvedeného vztahu vyplývá, že směrodatná odchylka nebude definována pro případ, kdy $\cos \Psi = 0$. Z toho důvodu byl zvolen rozsah úhlu natočení $-\pi/2 < \Psi < \pi/2$. Na obr. 22 lze tuto závislost pozorovat pro parametricky vynesené hodnoty odstupu signálu od šumu.



Obr. 22 - CRLB pro AoA systém (vliv odstupu signálu od šumu)

Parametry simulace popisuje následující tabulka:

| N_a [-] | BW [GHz] | l [m] |
|-----------|------------|---------|
| 5 | 1 | 0.1 |

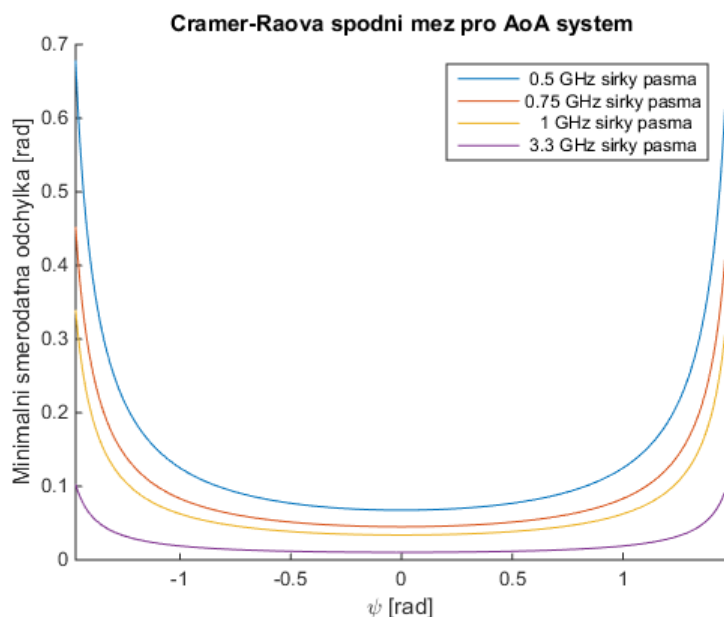
Tab. 7 - Parametry simulace AoA systému (různé hodnoty SNR)

Na obr. 23 je zobrazena obdobná závislost jako v předchozím případě s tím rozdílem, že byla zvolena fixní hodnota odstupu signálu od šumu a parametricky byla vynesena efektivní šířka pásma.

Parametry simulace:

| N_a [-] | SNR [dB] | l [m] |
|-----------|------------|---------|
| 5 | 10 | 0.1 |

Tab. 8 - Parametry simulace AoA systému (různé hodnoty efektivní šířky pásma)



Obr. 23 - CRLB pro AoA systém (vliv úhlu natočení odstupu signálu od šumu)

6 DecaWave EVK1000

Vývojová sada EVK1000 od společnosti DecaWave slouží k měření vzdálenosti mezi 2 body pomocí UWB signálu metodou „Two Way Ranging“ definované v [2]. Sada EVK1000 je na obr. 24.



Obr. 24 - Vývojová sada EVK1000, převzato z [13] (součástí balení jsou vždycky 2 kusy)

Deska obsahuje množství přepínačů sloužících např. k volbě kanálu, přenosové rychlosti, typu zařízení („tag“ nebo „anchor“) apod. Během provedených měření byl systém DecaWave nastaven ve výchozím módu od výroby, který je dle výrobce optimalizován pro dosažení co možná největšího dosahu. Nastavení systému DecaWave je v tab. 9. Střední frekvence módu 2 odpovídá 3,5 GHz. Šířka pásma v tomto kanálu odpovídá 500 MHz.

| Mód [-] | Kanál [-] | Přenosová rychlost [kb/s] | PRF [MHz] | Délka preamble [-] | Kód preamble [-] | SFD [-] |
|------------|--------------|---------------------------------|--------------|--------------------------|------------------------|--------------|
| 3 | 2 | 110 | 64 | 1024 | 9 | Nestandardní |

Tab. 9 - Nastavení systému DecaWave

Původní demonstrační aplikace v systému DecaWave obsahovala implementovanou metodu „Two Way Ranging“. V rámci diplomové práce byly provedeny následující úpravy:

1. Byla provedena implementace USB knihovny „*USB On-The-Go*“ (viz [14]) pro možnost komunikace s PC (posílání dat) po VCP rozhraní.
2. Zdrojový kód demonstrační aplikace byl upraven tak, aby byla umožněna komunikace mezi více uzly v síti (aby bylo možné řešit úlohu určení odhadu polohy).
3. Byla snížena perioda měření vzdálenosti demonstrační aplikace z 1s na 50 ms.
4. Byl implementován nový typ zařízení „*CTU listener*“ pro plnění funkce centrálního uzlu, který je nastaven v režimu neustálého příjmu.
5. Byl navržen formát zprávy (zpráva obsahuje informaci o naměřené vzdálenosti a ID dané kotvy) posílané centrálním uzlem do PC.

Dále byly provedeny další úpravy zdrojového kódu menšího rozsahu.

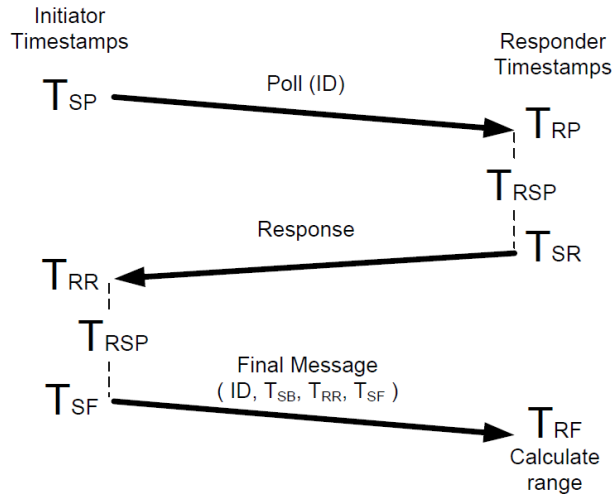
6.1 Určování vzdálenosti pomocí metody „Two Way Ranging“

Metoda nesoucí název „Two Way Ranging“ slouží k měření vzdálenosti dvou bodů na základě známé doby šíření signálu (TOF). Tento algoritmus je implementován ve vývojové sadě EVK1000.

Jak je dále popsáno v [15], jeden z uzlů měření vzdálenosti iniciuje (např. „*tag*“). Přijímací uzel (např. „*anchor*“) iniciátorovi odpovídá a provádí výpočet vzdálenosti („*tag*“ – „*anchor*“). Proces měření vzdálenosti metodou „Two Way Ranging“ lze popsat v několika následujících bodech:

1. Iniciátor měření vzdálenosti posílá tzv. „poll“ zprávu, přičemž si do paměti uloží dobu jejího odeslání T_{sp} . Poté iniciátor čeká na příchod odpovědi od protistrany (označována jako „responder“ – čili odpovídač).
2. Odpovídač si zaznamená dobu příchodu „poll“ zprávy – T_{RP} a poté odesílá zpět iniciátorovi „response message“ a zaznamená si dobu jejího odeslání T_{SR} .
3. Ve chvíli, kdy iniciátor přijme „response message“, zaznamená si opět čas jejího přijetí T_{RR} . V čase T_{SF} odesílá iniciátor tzv. „final response message“.
4. Odpovídač přijímající „final response message“ má nyní dostatek informací k tomu, aby mohla být určena vzdálenost mezi ním a iniciátorem.

Na obr. 25 je znázorněn „Two Way Ranging“ proces.



Obr. 25 – Princip metody "Two way ranging", převzato z [15], Initiator Timestamps = Časové značky iniciátora, Responder Timestamps = Časové značky „respondera“, Calculate range = Výpočet vzdálenosti

Doba doletu (TOF) je poté dána vztahem:

$$TOF = [(T_{RR} - T_{SP}) - (T_{SR} - T_{RP}) + (T_{RF} - T_{SR}) - (T_{SF} - T_{RR})] / 4 \quad (39)$$

Výraz lze dále zjednodušit na konečnou podobu:

$$TOF = [2(T_{RR} - T_{SR}) - T_{SP} + T_{RP} + T_{RF} - T_{SF}] / 4 \quad (40)$$

Vzdálenost mezi iniciátorem a odpovídačem lze určit ze součinu TOF a c , kde c je rychlost světla v daném prostředí.

$$d = TOF \cdot c \quad (41)$$

7 Penetrace UWB signálu skrz překážku

Tato část práce pojednává o šíření UWB signálu skrze překážku (například zdivo) z pohledu chyby určení vzdálenosti systémem založeným na měření doby šíření signálu v případě výskytu materiálu s rozdílnou relativní permitivitou na cestě mezi vysílačem a přijímačem. Taková metoda je obzvlášť vhodná pro UWB technologii z pohledu Cramer-Raovy spodní meze (u systému DecaWave bývá označována jako TOF). Kapitola obsahuje jak teoretický popis problému, tak obrázky ze simulací, které byly provedeny za pomoci programu Matlab R2014. V kapitole 7.2 je nastíněna možnost využití této chyby na určení relativní permitivity zdiva – tj. úloha opačná.

7.1 Analýza relativní odchylky určení vzdálenosti

Vzhledem ke skutečnosti, že systém DecaWave „nepočítá“ s překážkou umístěnou mezi vysílač a přijímač, dojde zákonitě k chybě výpočtu této vzdálenosti. Důvodem je to, že vložený materiál (uvnitř budovy se může jednat např. o zeď či strop) má jinou relativní permitivitu než okolní prostředí (vzduch). Elektromagnetické vlnění se v něm tedy šíří jinak rychle (zpravidla pomaleji). Na obr. 26 je znázorněno schéma analyzované situace.



Obr. 26 - analyzovaná situace

System DecaWave dokáže měřit dobu šíření signálu t . Tato je posléze násobena velikostí rychlosti světla ve vzduchu c_{air} , čímž je určen odhad vzdálenosti d_{DW} mezi vysílačem a přijímačem.

$$d_{DW} = tc_{air} \quad (42)$$

Velikost rychlosti světla ve vakuu c_0 lze spočítat dle následujícího vztahu:

$$c_0 = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \quad (43)$$

Kde ϵ_0 je permitivita vakua a μ_0 je permeabilita vakua. Hodnoty těchto konstant jsou obsaženy v následující tabulce:

| Permitivita vakua [F/m] | Permeabilita vakua [H/m] | Rychlost světla ve vakuu [m/s] |
|-------------------------|--------------------------|--------------------------------|
| $8,854 \cdot 10^{-12}$ | $4\pi \cdot 10^{-7}$ | $2,9979 \cdot 10^8$ |

Tab. 10 - Parametry volného prostředí

Jelikož relativní permitivita vzduchu činí 1,00054, bude se tedy i ve vzduchu elektromagnetické vlnění šířit nepatrně pomaleji ve srovnání s vakuem. Obecně pak bude platit, že velikost rychlosti světla ve zdivu bude $\sqrt{\epsilon_r}$ krát menší než rychlost šíření elektromagnetického vlnění ve vakuu.

$$c = \frac{c_0}{\sqrt{\epsilon_r}} \quad (44)$$

Jelikož se dle obr. 26 jedná o nepřímou viditelnost mezi vysílačem a přijímačem, je zřejmé, že v místě překážky dojde ke změně rychlosti šíření vlnění (vzhledem k rozdílné relativní permitivitě).

V programu Matlab byl vytvořen skript, který počítá relativní odchylku určení vzdálenosti mezi vysílačem a přijímačem od vzdálenosti skutečné:

$$\delta_d = 100 \frac{|d_{DW} - d|}{d} [\%] \quad (45)$$

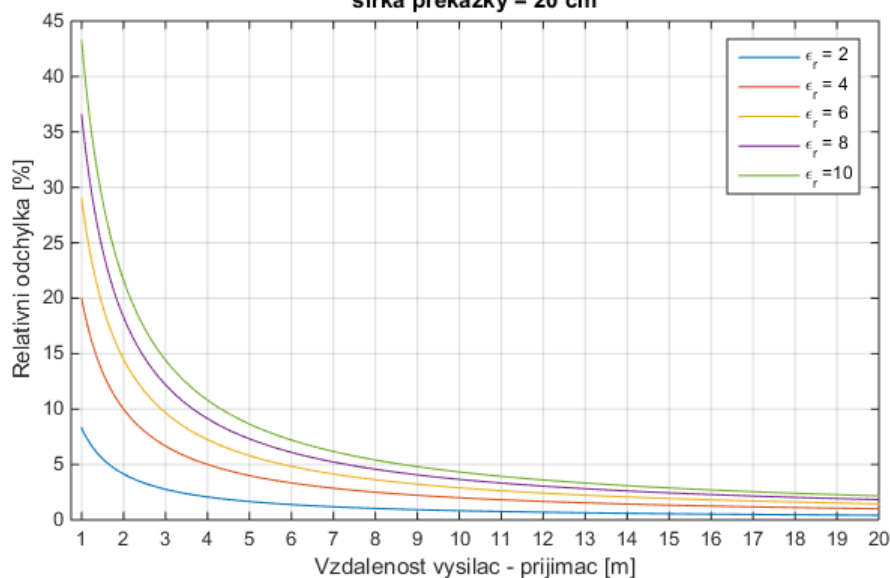
Kde d je skutečná vzdálenost mezi vysílačem a přijímačem. Doba šíření signálu bude dána součtem doby šíření signálu ve vzduchu a ve vloženém materiálu:

$$t = \frac{d - d_{wall}}{c_{air}} + \frac{d_{wall}}{c_{wall}} \quad (46)$$

Kde d_{wall} je šířka vloženého materiálu. Na obr. 27 lze pozorovat graf závislosti relativní odchylky určení vzdálenosti na celkové délce (vysílač – přijímač) pro překážky s různou relativní permitivitou. Šířka překážky (zdi) byla zvolena pevně na 20 cm. Tato hodnota odpovídá šířce zdi budovy ČVUT Fakulty elektrotechnické s označením T2:B2-825.

Ze sestrojeného grafu je patrné, že čím bude vysílač blíže přijímači, tím větší bude relativní odchylka určení vzdálenosti systémem DecaWave. Je to dáno tím, že zmenšení celkové vzdálenosti zvětšuje vliv překážky, jejíž šířka je pevná. Z grafu (obr. 27) je dále patrný vliv relativní permitivity (vynesena parametricky) na relativní odchylku určení vzdálenosti. Je zřejmé, že materiály s větší relativní permitivitou budou odhad vzdálenosti více „znepřesňovat“. Analýza byla provedena pro maximální vzdálenost 20 m mezi vysílačem a přijímačem. Na takové délce se již celková nepřesnost způsobená výskytem vloženého materiálu pohybuje jen v řádu procent.

Zavislost relativni odchylky na vzdalenosti mezi vysilacem a prijimacem pro ruzne materialy, sirka prekazky = 20 cm

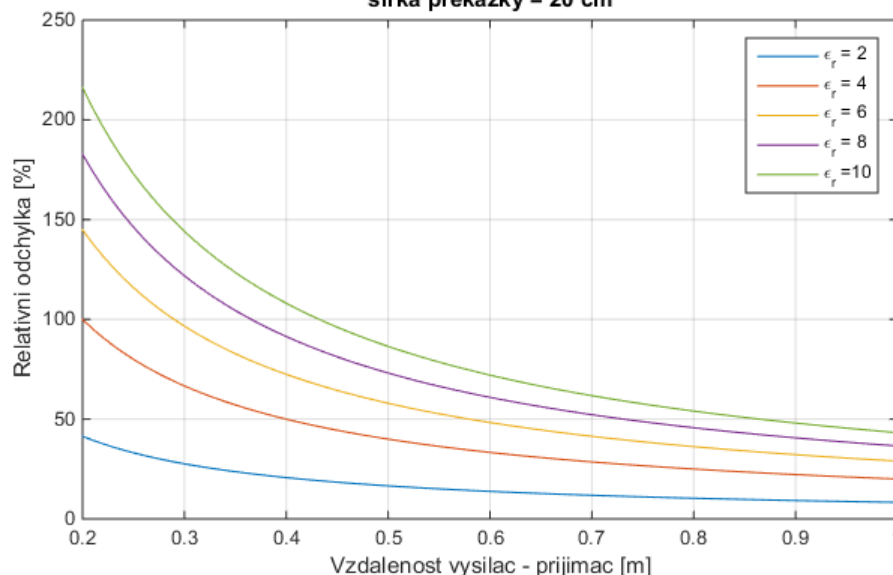


Obr. 27 - Závislost relativní odchylky na vzdálenosti

Graf na obr. 27 byl sestaven za několika zjednodušujících předpokladů. U vloženého materiálu se předpokládají magnetické vlastnosti stejné jako vlastnosti vakua ($\mu_r = 1$). Reálné materiály se dále pak mohou vyznačovat tím, že jejich relativní permitivita bude frekvenčně závislá. V tomto případě se předpokládá frekvenční nezávislost. Graf byl sestaven pro vzdálenosti od 1 do 20 metrů.

Na obr. 28 byl pak sestaven obdobný graf pro menší vzdálenosti mezi vysílačem a přijímačem. Minimální použitá vzdálenost odpovídá šířce překážky (100% zastoupení zdiva). Horní mez byla zvolena na 1 metr vzdálenosti (zároveň je tato dolní mezí obr. 27). Je zřejmé, že relativní odchylka vzdálenosti bude tím větší, čím větší bude procentuální zastoupení překážky s rozdílnou relativní permitivitou.

Zavislost relativni odchylky na vzdalenosti mezi vysilacem a prijimacem pro ruzne materialy, sirka prekazky = 20 cm



Obr. 28 - Relativní odchylka pro malé vzdálenosti

7.2 Odhad relativní permitivity vloženého materiálu

Opačným postupem vůči předchozímu textu by mohlo být možné získat odhad relativní permitivity vloženého materiálu. Předpokládejme výskyt materiálu s rozdílnou relativní permitivitou mezi vysílačem a přijímačem podle obr. 26. Doba šíření t lze rozdělit na 2 části – první bude doba šíření ve vzduchu a druhou bude doba šíření ve vloženém materiálu.

$$d_{DW} = \left(\frac{d - d_{wall}}{c_{air}} + \frac{d_{wall}}{c_{wall}} \right) c_{air} \quad (47)$$

Dále byla pravá strana rovnice roznásobena c_{air} .

$$d_{DW} = d - d_{wall} + \frac{d_{wall} c_{air}}{c_{wall}} \quad (48)$$

V dalším kroku vyjádříme rychlosti světla v různých materiálech pomocí rychlosti světla ve vakuu a relativní permitivity materiálu:

$$d_{DW} = d - d_{wall} + \frac{d_{wall} \frac{c_0}{\sqrt{\epsilon_{r_{air}}}}}{\frac{c_0}{\sqrt{\epsilon_{r_{wall}}}}} \quad (49)$$

Z předchozího vztahu vyplývá, že celková chyba určení vzdálenosti bude úměrná odmocnině z podílu relativní permitivity vloženého materiálu a vzduchu. Čím více se tento podíl bude blížit hodnotě 1, tím menší chyby určení vzdálenosti dosáhneme. Rychlosti světla ve vakuu lze vykrátit a dostaneme vztah:

$$d_{DW} = d - d_{wall} + \frac{d_{wall}\sqrt{\varepsilon_{r_{wall}}}}{\sqrt{\varepsilon_{r_{air}}}} \quad (50)$$

Osamostatněním $\varepsilon_{r_{wall}}$ poté dostaneme kýženou relativní permitivitu měřeného materiálu (např. zdiva):

$$\varepsilon_{r_{wall}} = \left(\frac{d_{DW} - d + d_{wall}}{d_{wall}} \sqrt{\varepsilon_{r_{air}}} \right)^2 \quad (51)$$

Tímto způsobem lze provést odhad relativní permitivity vloženého zdiva. Podmínkou je znalost geometrie celé úlohy, tj. znalost skutečné vzdálenosti d mezi vysílačem a přijímačem a šířky zdiva d_{wall} . Dále pak znalost relativní permitivity vzduchu (může se mírně lišit v závislosti na aktuálních podmínkách).

8 Výpočet odhadu pozice

Výpočet odhadu pozice pohyblivého bodu tzv. „tagu“ byl proveden pomocí Levenberg-Marquartova iteračního algoritmu. Tento algoritmus byl implementován jednak v programu Matlab R2014 (v rámci následného zpracování) a jednak v C# aplikaci pracující v reálném čase. V této kapitole bude proveden matematický popis zvoleného algoritmu včetně komentáře. Tento algoritmus byl zvolen s ohledem na jeho konvergenční vlastnosti (ve srovnání s Newton-Raphsonovým algoritmem, který má větší pravděpodobnost divergence).

8.1 Levenberg-Marquartův iterační algoritmus

Pro výpočet odhadu pozice „tagu“ je nutné znát souřadnice a naměřené vzdálenosti od jednotlivých kotev – „anchorů“. Levenberg-Marquartův algoritmus v každé iteraci zpřesňuje odhad pozice pohyblivého bodu. V C# aplikaci byla inicializační pozice zvolena v geometrickém středu kotev (aritmetický průměr jednotlivých souřadnic). V programu Matlab R2014b byl zvolen bod $[1,1,1]$. Tento bod se každou iterací přibližuje skutečné pozici „tagu“. Tolerance Levenberg-Marquartova algoritmu byla zvolena 1 mm. Tolerance má význam vzdálenosti mezi dvěma posledními odhady pozice.

V prvním kroku jsou počítány vzdálenosti odhadované pozice r_i (v prvním cyklu např. geometrický střed) od i -té kotvy. Kde i je index kotvy, v našem případě $i \in \{0,1,2,3\}$.

$$r_i = \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2 + (z_u - z_i)^2} = \|\mathbf{p}_u - \mathbf{p}_i\| \quad (52)$$

V dalším kroku je odhad polohy rozdělen na predikovanou (zpočátku nepřesnou – postupně se zpřesňující) a korekční část.

$$\mathbf{p}_u = \mathbf{p}_{up} + \mathbf{p}_{uc} \quad (53)$$

Stejným způsobem lze rozdělit vzdálenost měřenou i -tou kotvou.

$$r_i = r_{ip} + r_{ic} \quad (54)$$

Z této rovnice může být vypočten rozdíl měřené a predikované vzdálenosti. Z tohoto rozdílu je s využitím linearizace problému vypočtena korekce polohy (viz následující vztah)

$$\mathbf{p}_{uc} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \mathbf{r}_{ic} \quad (55)$$

kde \mathbf{G} je tzv. matice geometrie (matice směrových kosinů ochuzená o poslední sloupec), která obsahuje řádkové jednotkové směrové vektory směřující od příslušných kotev k „tagu“.

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_1^T \\ \mathbf{1}_2^T \\ \vdots \\ \mathbf{1}_M^T \end{bmatrix} \quad (56)$$

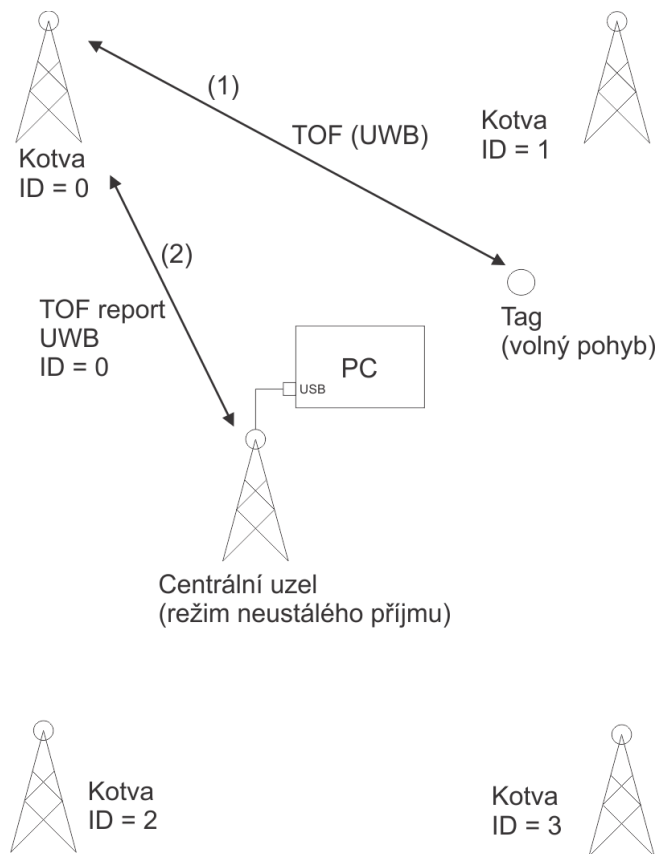
Vektor \mathbf{r}_{ic} je vektorem korektorů měřené vzdálenosti. Koeficient λ je parametrem Levenberg-Marquartovy metody (v C# aplikaci i v programu Matlab byla zvolena $\lambda = 0,01$). Velice zjednodušeně lze říci, že nižší hodnoty vedou k rychlejší konvergenci a vyšší hodnoty k větší stabilitě iterativní metody. Volba koeficientu je empirická. Vypočtený korekční vektor polohy je přičten k predikci. Získaný odhad je v další iteraci použit jako predikce. Tento cyklus je zastaven v případě, že velikost korekčního vektoru $\|\mathbf{p}_{uc}\|$ je menší než předem stanovená prahová hodnota (tolerance).

9 Testování systému DecaWave

Se systémem DecaWave bylo provedeno několik experimentů. Pro provedené experimenty byla využita soustava 6 vývojových sad EVK1000 (4x kotva – „anchor“ pro jednoznačné určení pozice pohyblivého bodu, 1x „tag“ – pohyblivý bod, 1x „CTU listener“ – centrální uzel). Topologie zapojení systému při provedených testech je znázorněna na obr. 29.

1. V prvním kroku se provede měření vzdálenosti mezi „tagem“ a „anchorem“ (kotvou) pomocí metody „Two Way Ranging“, přičemž „tag“ oslovuje příslušnou kotvu.
2. Poté, co je změřena doba šíření signálu TOF , je tato vynásobena velikostí rychlosti světla c v daném prostředí, čímž je určena vzdálenost mezi „tagem“ a příslušnou kotvou.
3. V dalším kroku je tato vzdálenost odeslána kotvou směrem k centrálnímu uzlu, který je nastaven v režimu neustálého příjmu UWB signálu.
4. Po přijetí naměřené vzdálenosti posílá centrální uzel do PC po virtuální sériové lince údaje o vzdálenosti mezi pohyblivým bodem a příslušnou kotvou.

Tyto úkoly trvají 50 ms. Po jejich dokončení „tag“ provede stejné operace s další kotvou.



Obr. 29 - Topologie pro testování

Centrální uzel byl napájen z USB konektoru na PC (Lenovo E545). Napájení ostatních kotev bylo vyřešeno pomocí USB power banky AXAGO MOBILE BATTERY 2200 (kapacita 2200 mAh; vstup DC 5V, 800 mA; výstup DC 5V, 800 mA).

9.1 Časování

Systém DecaWave byl nastaven tak, aby pohyblivý bod změřil vzdálenost od příslušné kotvy každých 50 ms. Pohyblivému bodu byl deaktivován „sleep mode“, takže přechod mezi 4. a 1. kotvou trval stejně dlouho jako ostatní přechody.

9.2 Komunikace s PC

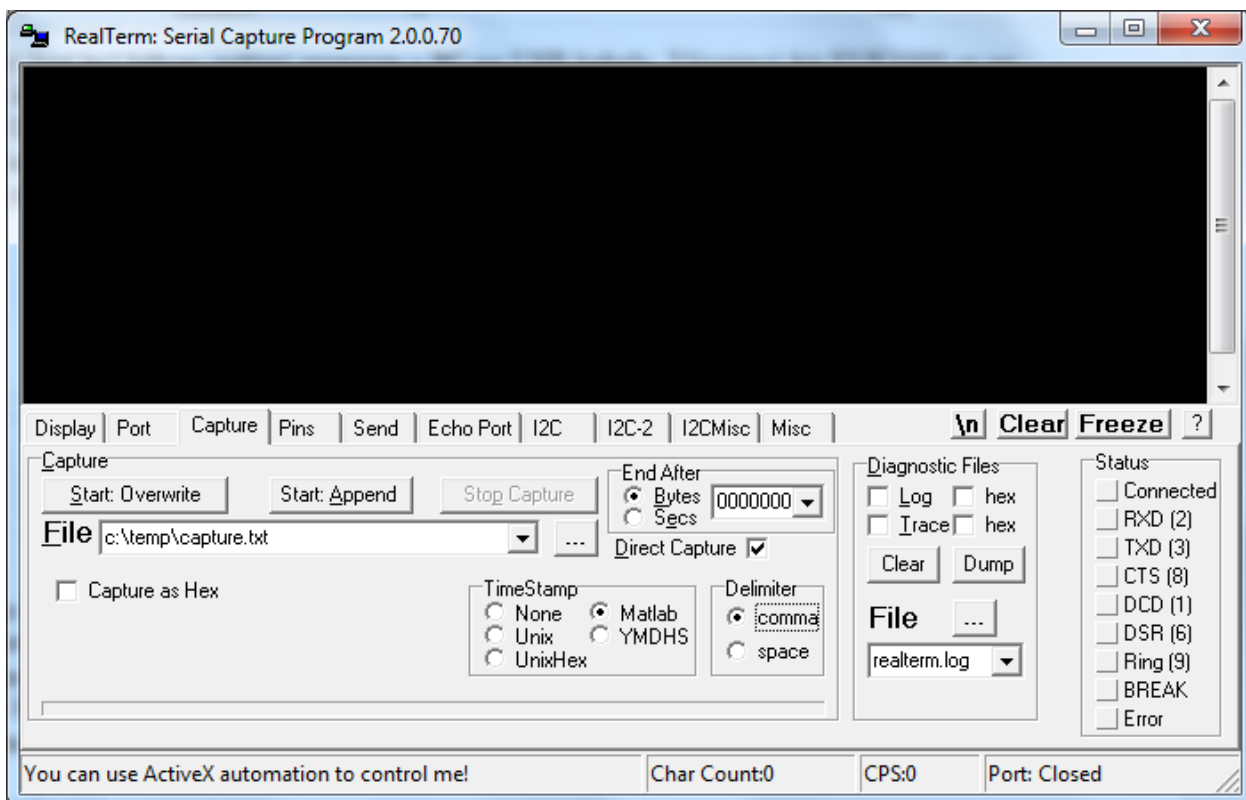
Centrální uzel – „CTU listener“ – byl během měření propojen s PC po USB kabelu. Vývojová sada EVK1000 se po připojení k PC identifikuje jako VCP (virtuální sériový port). Při každé nové změřené vzdálenosti posílá EVK1000 do PC zprávu s formátem: ID kotvy [-], změřená vzdálenost [m].

9.2.1 Komunikace pomocí programu RealTerm

Komunikace systému DecaWave s PC byla zpočátku zajištěna pomocí programu RealTerm, který podporuje možnost přidávání časových značek. Byla použita časová značka typu Matlab (ze všech možností umožňuje nejlepší rozlišení v čase) a jako oddělovač byla použita čárka. Formát Matlab je udáván jako počet dnů od počátku našeho letopočtu s pohyblivou desetinnou čárkou. Časové rozlišení je dáno rozlišením hodin v PC. V tomto případě bylo za celými dny 6 desetinných míst, což vede k časovému rozlišení 86,4 ms.

$$10^{-6} \text{ dnů} = 10^{-6} \cdot 24 \cdot 60 \cdot 60 \text{ s} = 86,4 \text{ ms} \quad (57)$$

Vzhledem k nastavení časování systému DecaWave bylo tedy zajištěno, aby naměřené vzdálenosti od jednotlivých kotev měly vždy jinou časovou značku. Nastavení programu RealTerm je znázorněno na obr. 30.



Obr. 30 - Nastavení programu RealTerm

Příklad naměřených dat tedy vypadal např. takto (první sloupec odpovídá časové značce):

```
736301.617032,2,20.84
736301.617033,3,4.62
736301.617034,0,18.05
736301.617035,1,4.77
736301.617035,2,20.88
736301.617036,3,4.64
736301.617037,0,18.06
736301.617037,1,4.76
736301.617038,2,20.82
736301.617039,3,4.71
736301.61704,0,18.03
```

Jednotlivé sloupce byly odděleny čárkou, jako desetinný oddělovač byla použita tečka. Takto naměřená data byla posléze zpracovávána pomocí programu Matlab.

9.2.2 Komunikace pomocí vlastní C# aplikace

Vzhledem k tomu, že v rámci této diplomové práce byla vytvořena i aplikace v jazyku C#, bylo příhodné do ní implementovat i možnost logování dat (tak jako to umí program

RealTerm). Druhou funkcí aplikace je zobrazování polohy pohyblivého bodu v reálném čase (ukázalo se totiž, že program Matlab R2014b vykresluje pozici pohyblivého bodu příliš pomalu a nelze ho tedy použít pro zobrazování v reálném čase). Princip fungování aplikace je popsán v samostatné kapitole, zde bude uveden pouze nový formát dat a časové rozlišení.

Aby bylo možné přidávat časovou značku na začátek přijaté zprávy, byl upraven její formát následujícím způsobem:

T156I03D001.24
T180I00D007.72
T183I01D007.84
T186I02D003.19
T218I03D001.16
T383I00D007.78
T427I01D007.83
T466I02D003.19

Kde modře označený text lze interpretovat jako časovou značku přidanou C# aplikací. Ta má oproti programu RealTerm výhodu v tom, že její časové rozlišení je 1 ms (RealTerm má 86,4 ms). Písmeno „T“ bylo použito jako začátek slova „*time*“. Dále byl upraven zdrojový kód k vývojové sadě EVK1000. Do posílané zprávy bylo na začátek přidáno písmeno „I“ (jako ID) a před posílanou vzdálenost písmeno „D“ jako „*distance*“. Zpráva posílaná vývojovou sadou EVK1000 je obarvena zeleně.

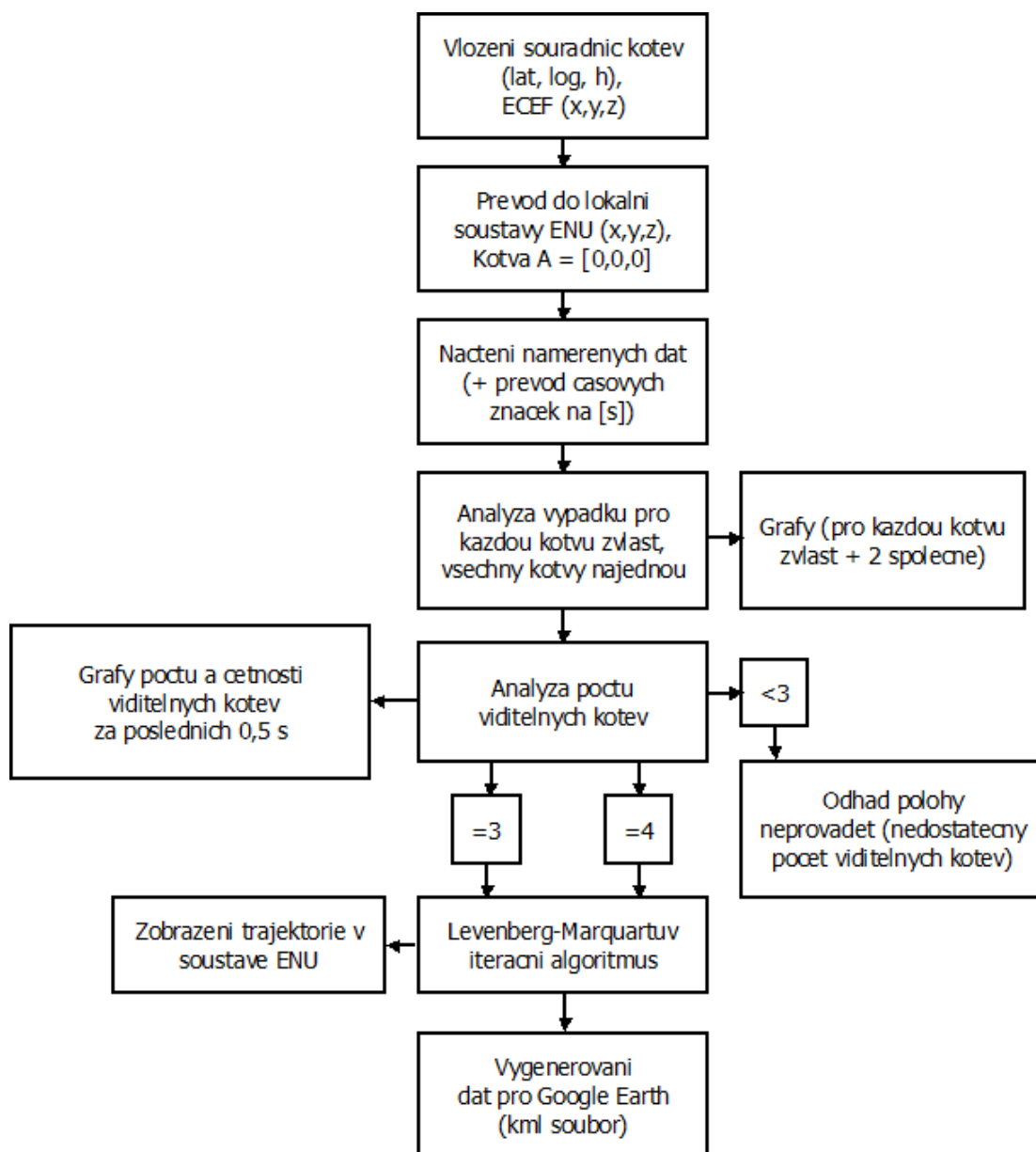
9.3 Princip fungování skriptu v programu Matlab R2014 b

Naměřené vzdálenosti včetně ID jednotlivých kotev byly zpracovávány až následně. Funkce skriptu v programu Matlab R2014b bude popsána v následujícím textu. Blokové schéma programu Matlab je znázorněno na obr. 31. Princip lze popsat v několika následujících krocích:

1. V prvním kroku je nutné znát souřadnice kotev. Ty byly určeny pomocí metody RTK. Souřadnice kotev byly zadány jednak ve formátu zeměpisná šířka, zeměpisná délka a výška nad elipsoidem a jednak v pravoúhlé soustavě ECEF, kde bod [0,0,0] odpovídá středu zeměkoule. Mezi souřadnicemi existuje jednoznačný přepočít.
2. V dalším kroku je proveden převod do lokální soustavy označované jako ENU („East – North – Up“), kde kotvě „A“ byly přisouzeny souřadnice [0,0,0].
3. Následně se provede načtení naměřených dat (buď programem RealTerm nebo vlastní C# aplikací s příznivějším časovým rozlišením).
4. V dalším kroku je provedena analýza výpadků měřené vzdálenosti. Výstupem jsou grafy závislosti změřené vzdálenosti na časové značce od jednotlivých kotev. V ideálním případě by byly naměřené vzdálenosti na časové ose pravidelně rozmístěny. Vlivem výpadků (např. kvůli nedodržení podmínky přímé viditelnosti – LOS) se v některých časových intervalech snižuje počet naměřených vzdáleností.
5. Následně je provedena analýza počtu viditelných kotev, jejímž výstupem jsou grafy počtu a četnosti viditelných kotev za posledních 0,5 s. V ideálním případě je zajištěna konektivita od všech 4 kotev (takové body jsou ve výsledném grafu zobrazeny červeně). V případě, kdy jsou viditelné pouze 3 kotvy (vyznačeno fialovou barvou), lze provést odhad pozice pomocí Levenberg-Marquartova algoritmu také, úloha je však symetrická podle horizontální roviny a nelze tedy

jednoznačně určit z souřadnic. Vzhledem k tomu, že vypočtená trajektorie pohyblivého bodu je následně zobrazena v horizontální rovině, nejednoznačnost v ose z není z tohoto úhlu pohledu problém.

6. V posledním kroku je proveden export dat do kml souboru, což umožňuje vykreslení a zobrazení bodů v Google Earth. K tomu byl využit Google Earth Toolbox.



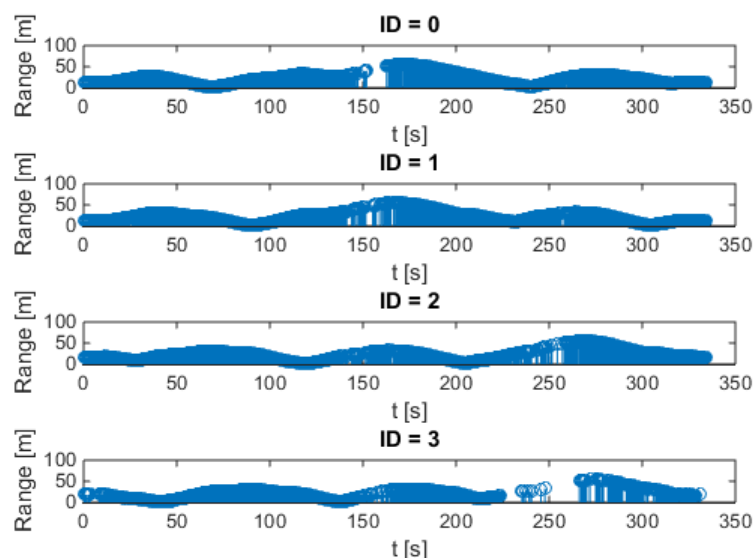
Obr. 31 - Blokové schéma skriptu v programu Matlab R2014b

9.4 Měření ve venkovním prostředí

V následující kapitole budou shrnuty výsledky z měření provedeného dne 15. 2. 2015 v Dejvicích poblíž budovy elektrotechnické fakulty ČVUT. Byly zvoleny celkem 3 trajektorie pohyblivého bodu. První varianta byla zaměřená na relativně blízké okolí kotev („tag“ byl vzdálen nanejvýš cca 60 m od nejvzdálenější kotvy). Další varianty byly zaměřeny na určení maximálního dosahu systému DecaWave. Měřená pozice byla porovnávána s metodou RTK.

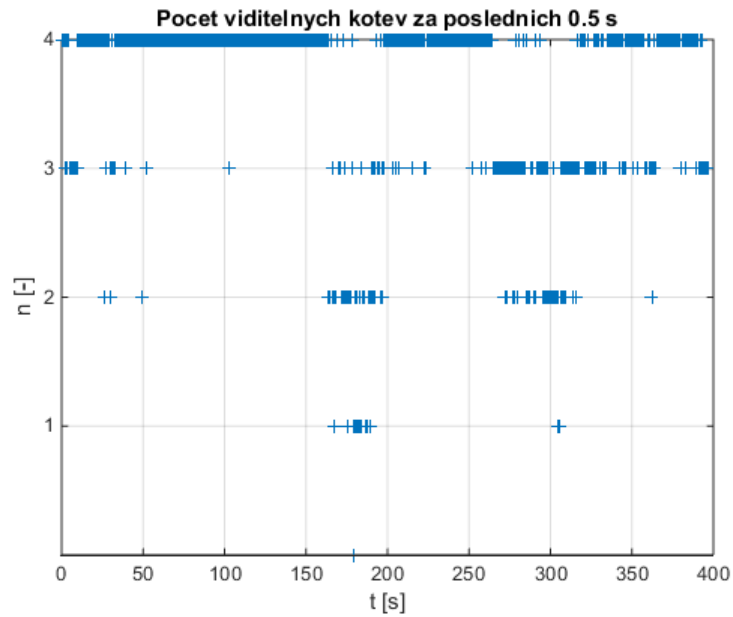
9.4.1 Blízké okolí kotev

V první řadě byla naměřená data zkoumána se zaměřením na výpadky konektivity od jednotlivých kotev, které mohou být způsobeny např. nedodržením podmínky přímé viditelnosti nebo překročením maximálního dosahu systému. Na obr. 32 lze pozorovat graf naměřených vzdáleností mezi pohyblivým bodem a kotvou s příslušným ID. Je zřejmé, že mezi „tagem“ a kotvou s ID = 0 došlo k výpadku kolem 150 s od počátku měření. Pro kotvu s ID = 3 došlo k dominantnímu výpadku kolem 250 s od započetí experimentu. Ostatní kotvy neobsahovaly výpadky dlouhodobějšího charakteru.

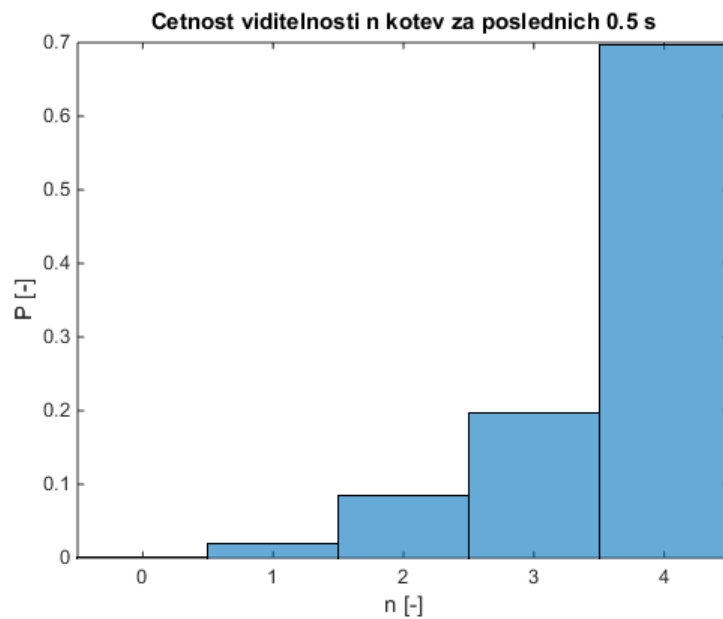


Obr. 32 - Výpadky od jednotlivých kotev (venkovní měření)

Následně byla zkoumána závislost počtu připojených kotev za posledních 0,5 s na čase – viz obr. 33. V dalším kroku byl v rámci analýzy výpadků zhotoven histogram z počtu viditelných kotev za posledních 0,5 s, který lze pozorovat na obr. 34. Z grafu je patrné, že viditelnosti od všech 4 kotev bylo dosaženo u cca 70 % případů (v takovém případě lze jednoznačně určit polohu pohyblivého bodu $[x, y, z]$). U zhruba 20 % případů bylo za posledních 0,5 s dosaženo konektivity od 3 kotev (konektivita od 1 kotvy byla ztracena). V takovém případě lze určit polohu „tagu“ pouze v horizontální rovině. V případě konektivity od 2,1, či žádné kotvy (cca 10%) nelze vůbec provést odhad pozice „tagu“.

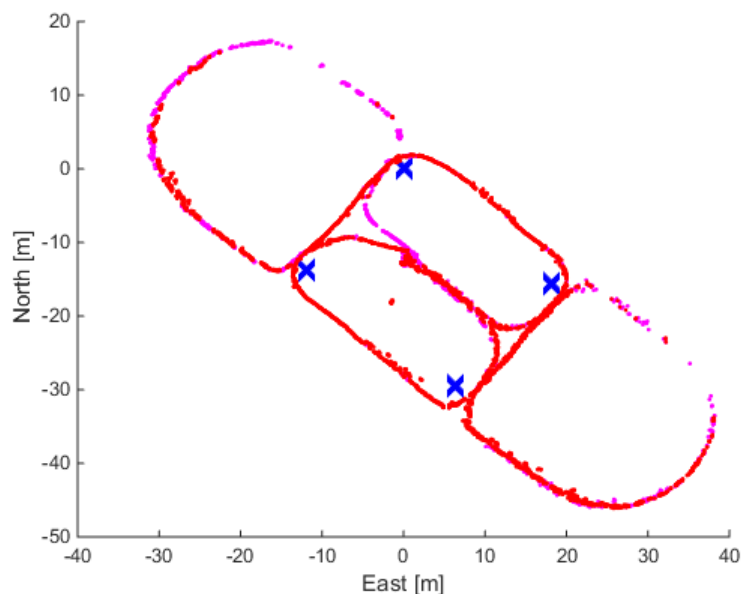


Obr. 33 - Závislost počtu viditelných kotev za posledních 0,5 s (venkovní prostředí)



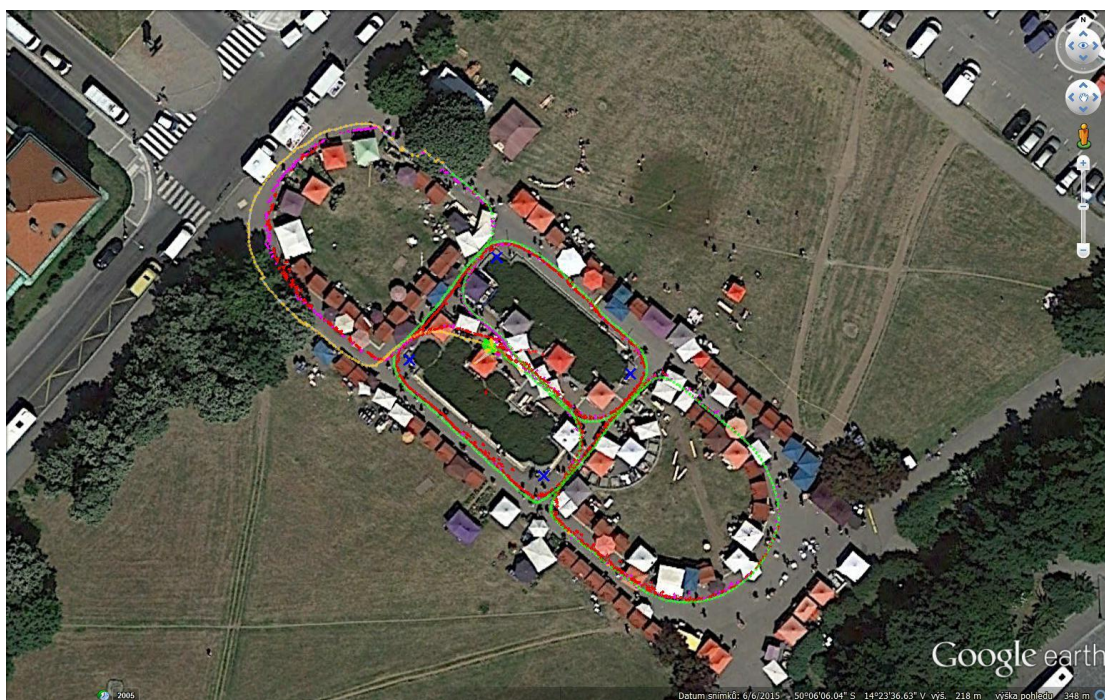
Obr. 34 – Četnost viditelných kotev za posledních 0,5 s (venkovní prostředí)

Následně byl počítán odhad pozice pohyblivého bodu pomocí Levenberg-Marquartova iteračního algoritmu. Na obr. 35 je znázorněn odhad trajektorie „tagu“ v ENU souřadnicích (osa x míří kladnějšími hodnotami na východ, osa y na sever). Modrou barvou byly vykresleny pozice kotev, červenou barvou pak odhady pozice „tagu“ v případě viditelnosti všech 4 kotev, fialově jsou vykresleny odhady v případě viditelnosti od 3 kotev.



Obr. 35 - Pozice pohyblivého bodu (venkovní měření, blízké okolí kotev)

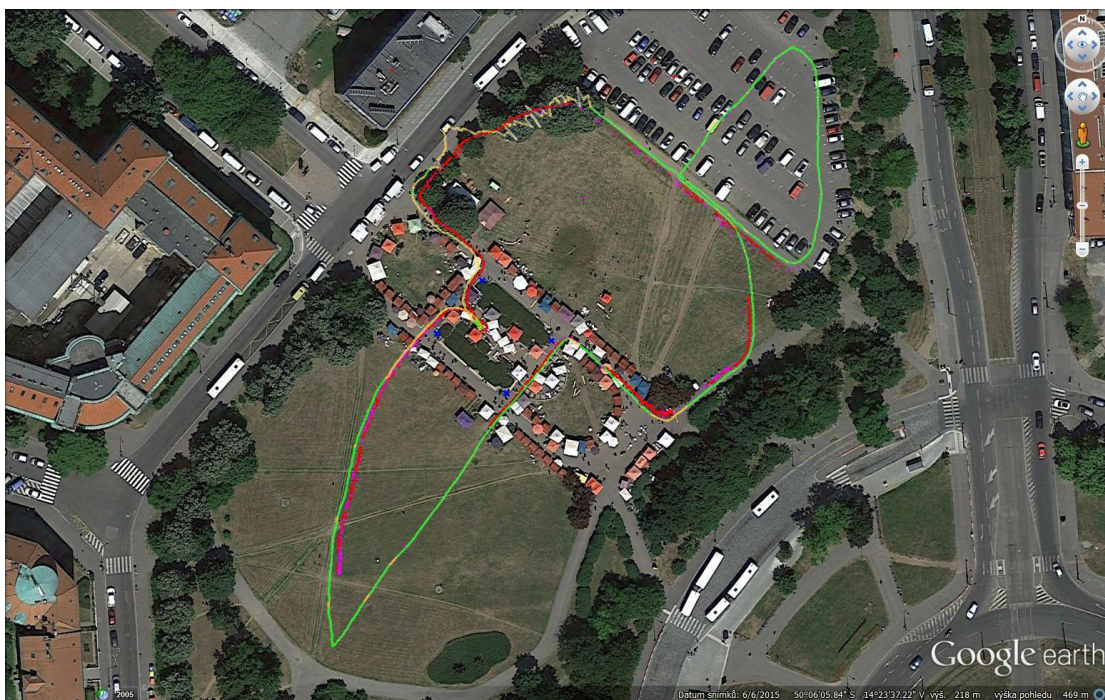
Na obr. 36 je trajektorie pohyblivého bodu promítnutá do Google map pomocí Google Earth toolboxu. Body vykreslené žlutě případně zeleně zobrazují data získaná metodou RTK.



Obr. 36 - Venkovní měření (blízké okolí kotev, Google Earth)

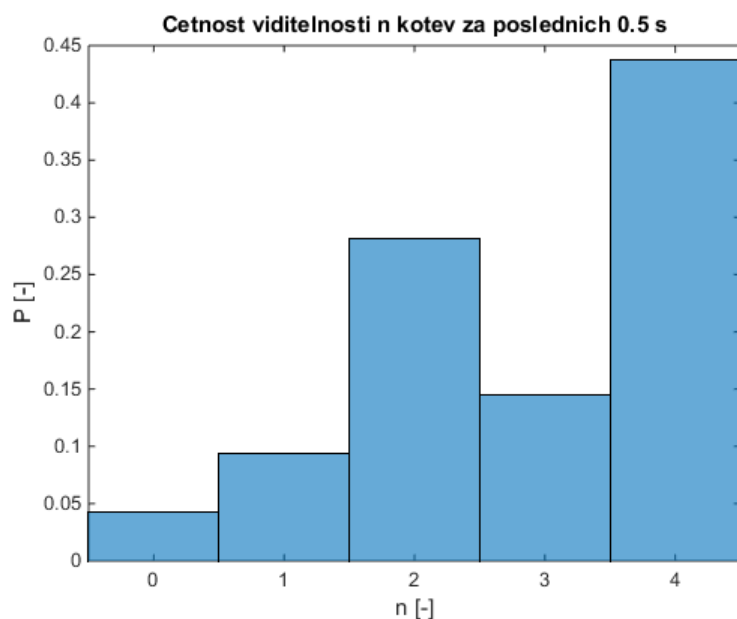
9.4.2 Vzdálenější okolí kotev

V případě, kdy byl pohyblivý bod více vzdálen od kotev, začalo docházet k častějším výpadkům konektivity od jednotlivých kotev. Trajektorii lze pozorovat na obr. 37. Lze pozorovat, že v místě zastínění GPS signálu vegetací (na obrázku vlevo od parkoviště) dosahoval systém DecaWave v porovnání s metodou RTK větší přesnosti. V případě větších vzdáleností vykazoval systém DecaWave současné výpadky od více kotev, což znemožnilo provedení výpočtu odhadu pozice „tagu“.



Obr. 37 - Venkovní měření (vzdálenější okolí kotev)

Na obr. 38 je četnost viditelnosti n kotev. Všechny kotvy současně byly viditelné jen u 44 % případů. Zajímavé je porovnání s obr. 34, kde docházelo k méně četným výpadkům.



Obr. 38 - Četnost viditelnosti - venkovní měření (vzdálenější oblast)

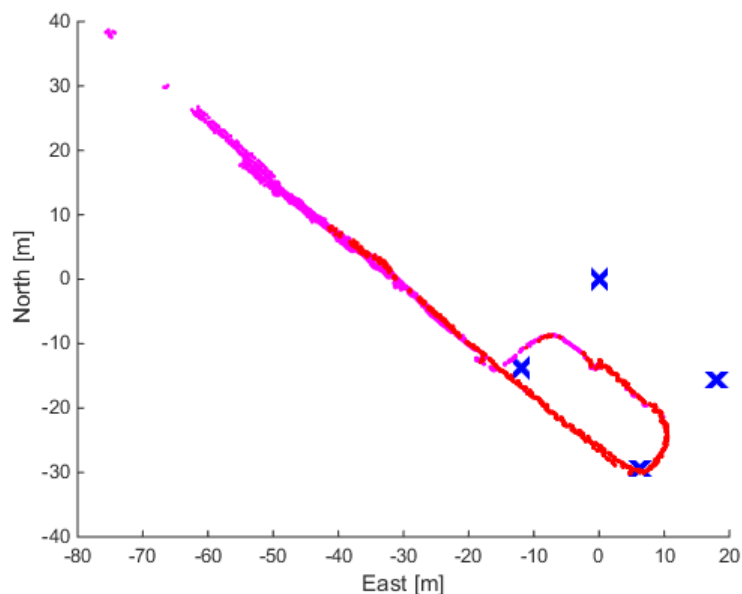
9.4.3 Trajektorie Technickou ulicí (Dejvice)

Další varianta trajektorie směřovala do Technické ulice (dokud nebylo dosaženo výpadků znemožňujících funkci systému DecaWave – tj. odhad polohy „tagu“ bylo možné určit alespoň v horizontální rovině). Promítnutí do ENU souřadnic je na obr. 39. Z obrázku je patrné, že v případě větší vzdálenosti došlo k výpadku 1 kotvy (fialové body). U menších vzdáleností byl odhad pozice počítán ze 4 změřených vzdáleností. Je třeba podotknout, že

mnohdy nebylo dosaženo přímé viditelnosti mezi kotvami a „tagem“. Například poté, co jsme překročili silnici, projel po ní autobus, který zamezil přímé viditelnosti.

Promítnutí do Google mapy je na obr. 40. Je třeba podotknout, že během měření nebyly na místě přítomny stánky – ty jsou na mapě zobrazeny zřejmě z důvodu jiné doby pořízení fotografie.

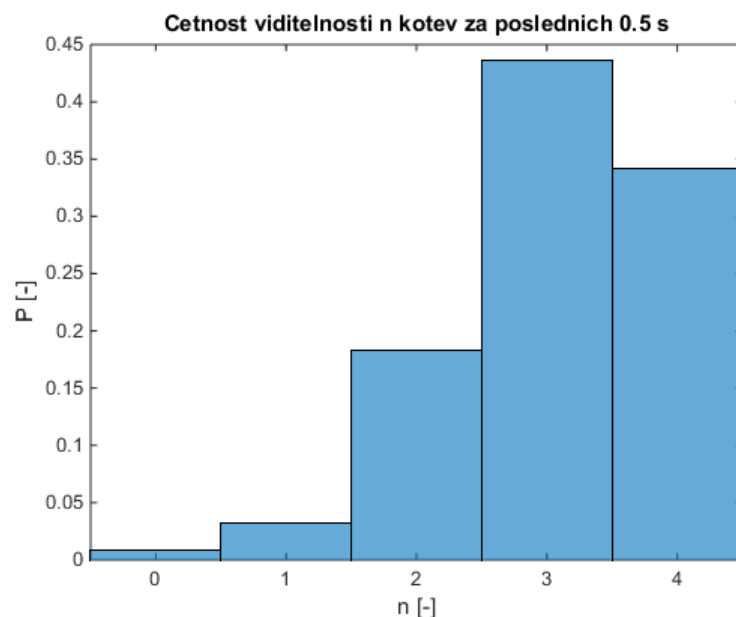
Na obr. 41 lze opět pozorovat četnost viditelnosti n kotev za posledních 0,5 s. Je vidět, že dominovala konektivita od 3 a 4 kotev současně.



Obr. 39 - Venkovní měření (Technická, ENU)



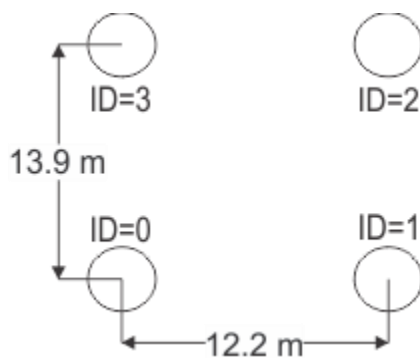
Obr. 40 - Venkovní měření (Technická ulice, Google Earth)



Obr. 41 - Venkovní měření (Technická, četnost viditelnosti)

9.5 Měření ve vnitřním prostředí

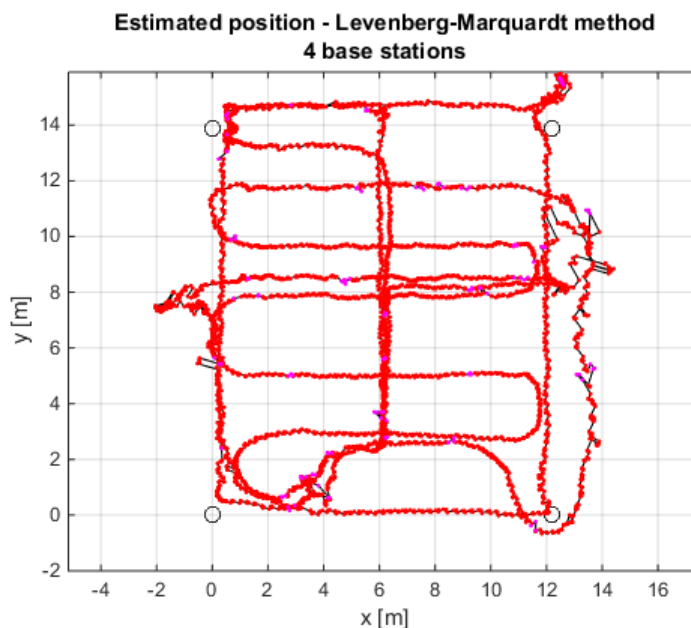
V následující kapitole budou komentovány výsledky měření systému DecaWave, které bylo provedeno dne 17. 02. 2016 v posluchárně T2:D3-309. Na obr. 42 lze pozorovat zvolené rozmístění kotev v prostoru posluchárny. Byly provedeny celkem 3 různé datové záznamy.



Obr. 42 - Rozmístění kotev v posluchárně 309

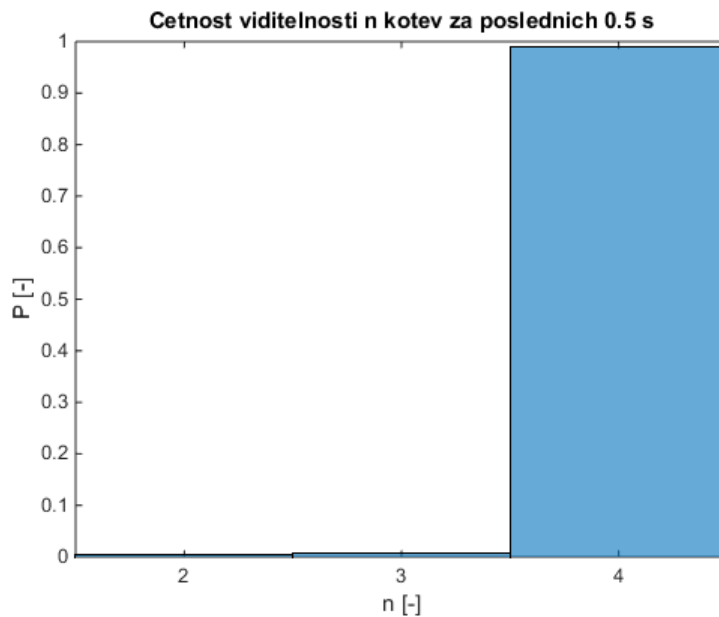
9.5.1 Trajektorie 1

V případě prvního provedeného měření lze odhad trajektorie pohyblivého bodu pozorovat na obr. 43. Fialové body naznačují, že odhad pozice „tagu“ byl počítán pouze od 3 kotev. V pravé části obrázku je na $y = 10$ m větší rozptyl naměřených hodnot. Ten byl způsoben výskytem nosného sloupu.



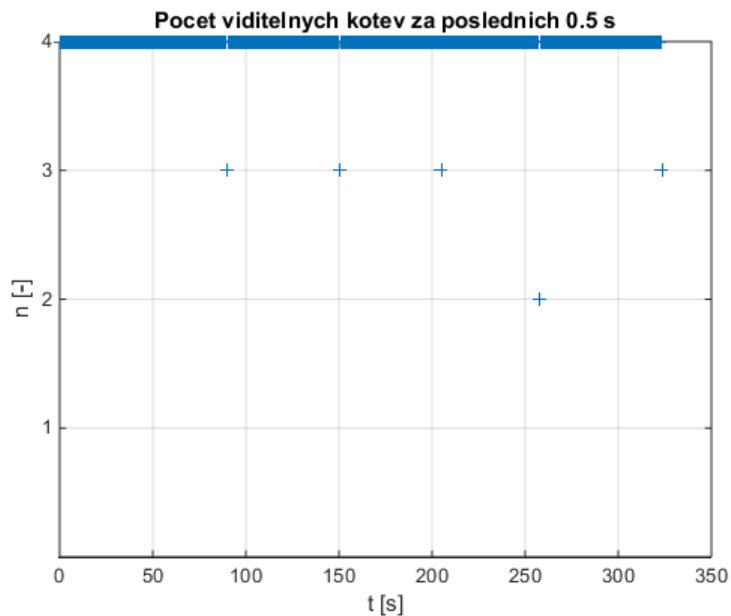
Obr. 43 - Měření v učebně 309 (trajektorie)

V rámci tohoto měření téměř nedocházelo k výpadkům, což dokládá obr. 44. Konektivity od všech 4 kotev za posledních 0,5 s bylo dosaženo v drtivé většině případů.



Obr. 44 - Vnitřní měření (četnost viditelnosti za posledních 0,5 s)

Minimální množství výpadků dále dokládá obr. 45. Za posledních 0,5 s nebylo dosaženo konektivity od všech 4 kotev pouze v 5 případech (ve 4 z těchto případů lze však i přes to určit polohu „tagu“).

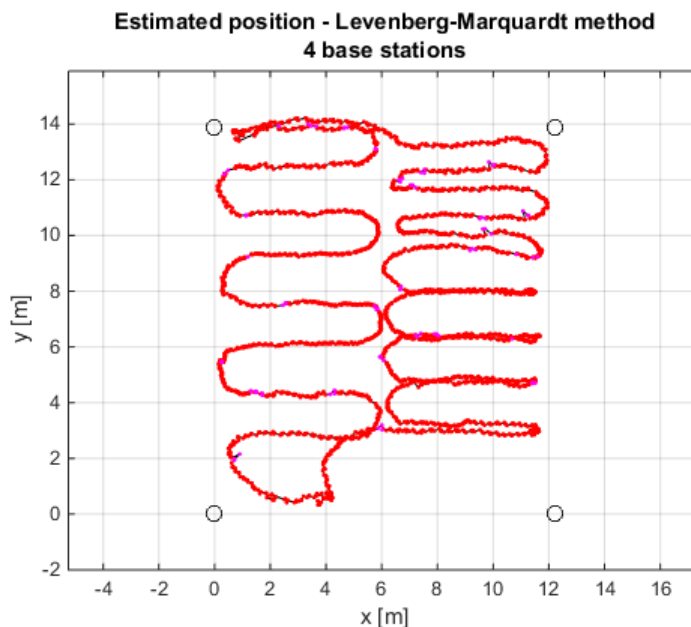


Obr. 45 - Závislost počtu viditelných kotev za posledních 0,5 s (učebna 309)

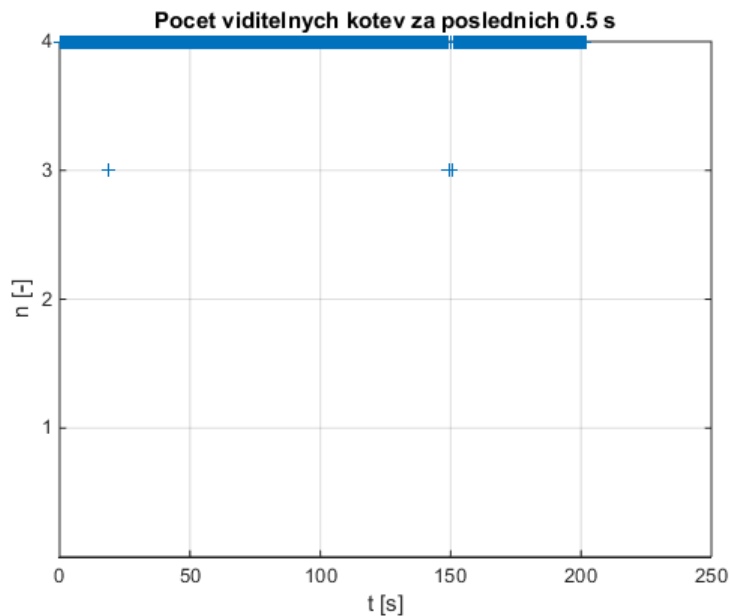
9.5.2 Trajektorie 2

Další trajektorie byla zvolena tak, že na levé straně posluchárny byl „tag“ přenášen kolem lavic ob uličku, na pravé straně posluchárny pak uličkou každou. Vypočtenou trajektorii lze pozorovat na obr. 46.

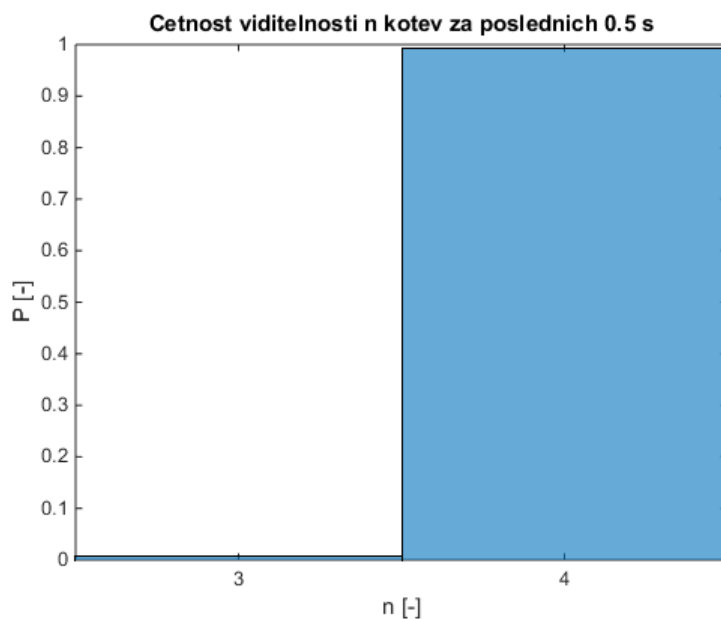
I v tomto případě docházelo, byť k zanedbatelným, výpadkům konektivity od jednotlivých kotev. Na obr. 47 lze pozorovat graf závislosti počtu viditelných kotev za posledních 0,5 s na čase. K výpadkům došlo v naprosto zanedbatelném množství případů - obr. 48.



Obr. 46 - Trajektorie (učebna 309)



Obr. 47 - Počet viditelných kotev za posledních 0,5 s (učebna 309)



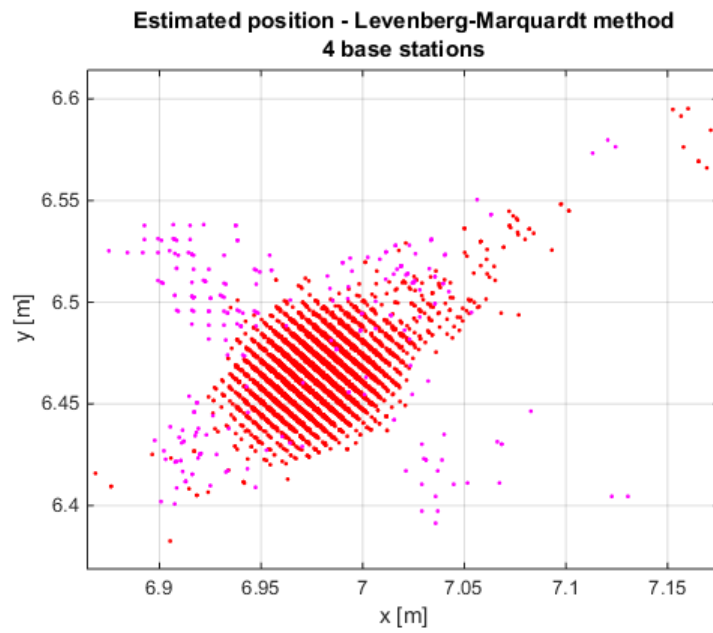
Obr. 48 - Četnost viditelnosti (učebna 309)

9.5.3 Statický bod

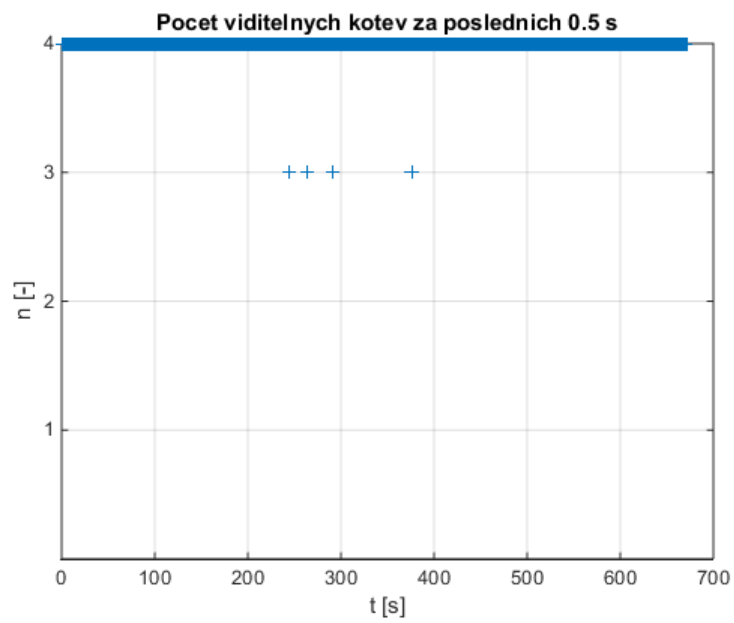
V poslední části měření byly vyhodnocovány výpadky a poloha „tagu“ stojícího na místě. Na obr. 49 jsou zobrazené odhadované pozice „tagu“ z naměřených vzdáleností od jednotlivých kotev. V ose x bylo dosaženo rozkmitu hodnot od cca 6,85 m do 7,2 m, v ose y poté od 6,35 m do 6,6 m. Je dobré si povšimnout, že vzdálenější pozice od celkového průměru byly většinou počítány od 3 kotev (vyznačeno fialovou barvou).

Z analýzy naměřených dat dále vyplývá, že za posledních 0,5 s došlo k výpadku od jedné z kotev ojediněle (4x) – viz obr. 50.

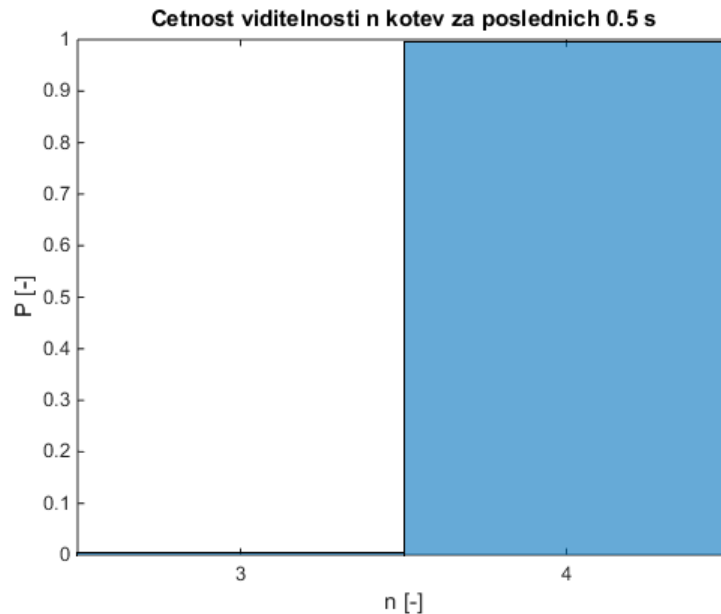
Z toho vyplývá, že četnost viditelnosti n kotev za posledních 0,5 s vypadala velmi příznivě. V drtivé většině případů bylo dosaženo viditelnosti od všech 4 kotev (obr. 51).



Obr. 49 - Statický bod (měření v učebně 309)



Obr. 50 - Statický bod (počet viditelných kotev za posledních 0,5 s, učebna 309)



Obr. 51 - Četnost viditelnosti (statický bod, učebna 309)

9.5.3.1 Statistické zpracování odhadů polohy statického bodu

Vzhledem k tomu, že „tag“ byl v tomto případě pevně ukotven na jednom místě, byla dále provedena analýza naměřených dat z pohledu směrodatné odchyly a výběrového rozptylu. Analýza byla provedena jednak pro jednotlivé odhady pozice (souřadnice x , y , z), tak pro naměřené vzdálenosti od jednotlivých kotev (d_0 , d_1 , d_2 , d_3).

Aritmetický průměr \bar{x} byl vypočet pomocí známého vztahu:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (58)$$

Pro výpočet výběrové směrodatné odchyly aritmetického průměru $\sigma(\bar{x})$ byl použit vztah (viz [16]):

$$\sigma(\bar{x}) = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (59)$$

Výběrový rozptyl jednoho měření $D(x)$ lze odvodit ze vztahu (viz [16]):

$$\sigma(\bar{x}) = \sqrt{\frac{D(x)}{N}} \rightarrow D(x) = \sigma^2(\bar{x})N \quad (60)$$

Výběrový rozptyl aritmetického průměru $D(\bar{x})$ je poté svázán s výběrovým rozptylem jednoho měření počtem měření N (viz [16]):

$$D(\bar{x}) = \frac{D(x)}{N} \quad (61)$$

Vypočtené hodnoty:

Statistické zpracování souřadnic x, y, z bylo provedeno pro $N_x = N_y = N_z = 10362$ odhadů polohy. Výběrový rozptyl aritmetického průměru byl v následující tabulce označen jako \overline{D} , aritmetický průměr jako μ .

| | μ [m] | σ [m] | D [m ²] | \overline{D} [m ²] | N [-] |
|-------|-----------|--------------|-----------------------|----------------------------------|---------|
| x | 6,97686 | 0,00020 | 0,00040 | $3,81 \cdot 10^{-8}$ | 10362 |
| y | 6,46408 | 0,00016 | 0,00025 | $2,45 \cdot 10^{-8}$ | 10362 |
| z | 0,65225 | 0,00162 | 0,02707 | $261,27 \cdot 10^{-8}$ | 10362 |
| d_0 | 9,57154 | 0,00063 | 0,00102 | $39,20 \cdot 10^{-8}$ | 2605 |
| d_1 | 8,30455 | 0,00037 | 0,00035 | $13,45 \cdot 10^{-8}$ | 2584 |
| d_2 | 9,14723 | 0,00035 | 0,00032 | $12,29 \cdot 10^{-8}$ | 2592 |
| d_3 | 10,17933 | 0,00032 | 0,00027 | $10,51 \cdot 10^{-8}$ | 2612 |

Tab. 11 - Statistické zpracování (statický bod)

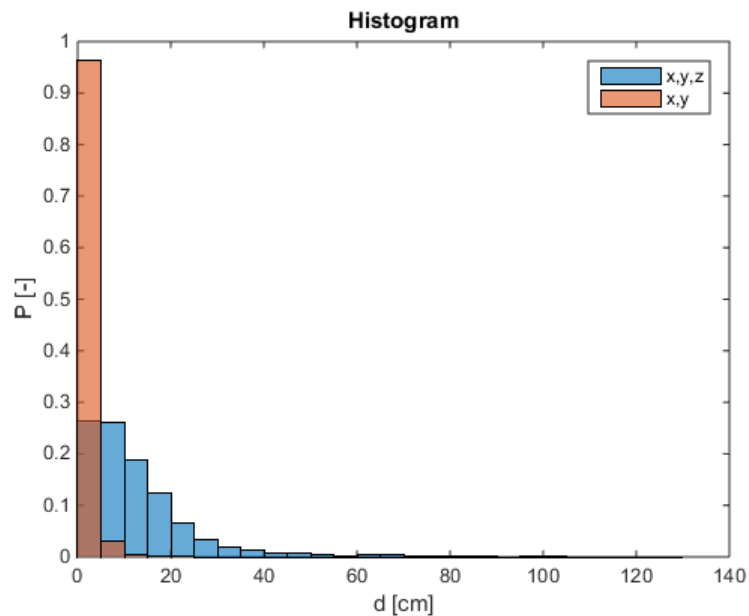
Další část práce se zabývala určením histogramu a distribuční funkce vzdáleností mezi i odhadem polohy a aritmetickým průměrem. Funkce byly zhotoveny pro vzdálenost v celém prostoru a v rovině x, y :

$$d_{x,y,z}^i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2}$$
$$d_{x,y}^i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \quad (62)$$

Horní index u vzdálenosti i nemá význam mocniny, ale jen indexu. Je však nutné podotknout, že následující obrázky jsou závislé na konkrétním rozmístění kotev v prostoru.

Histogram:

Pro vytvoření histogramu byl pro obě varianty zvolen krok odpovídající 5 cm. Obě varianty byly zobrazeny do jednoho grafu. Lze pozorovat, že v případě, kdy nás zajímá polohy „tagu“ pouze v horizontální rovině, jsou téměř všechny (cca 96 %) odhady polohy vzdálené do 5 cm od aritmetického průměru. V případě určení polohy v celém prostoru x, y, z jsou polohy „tagu“ vzdálené od aritmetického průměru do 5 centimetrů pouze v cca 27 % případů. Histogram lze pozorovat na obr. 52.

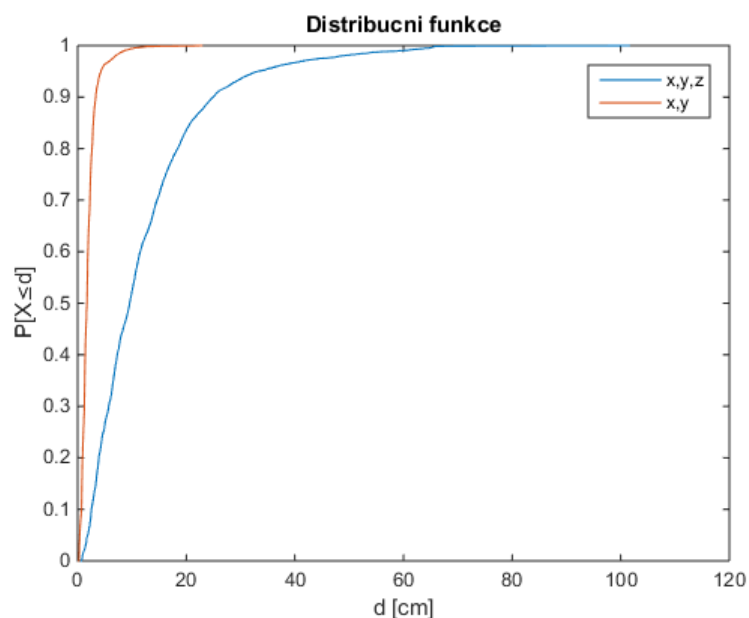


Obr. 52 - Histogram - statický bod

Distribuční funkce:

Z vypočtených vzdáleností mezi i odhadem pozice a aritmetickým průměrem byla dále sestavena empirická distribuční funkce – viz obr. 53.

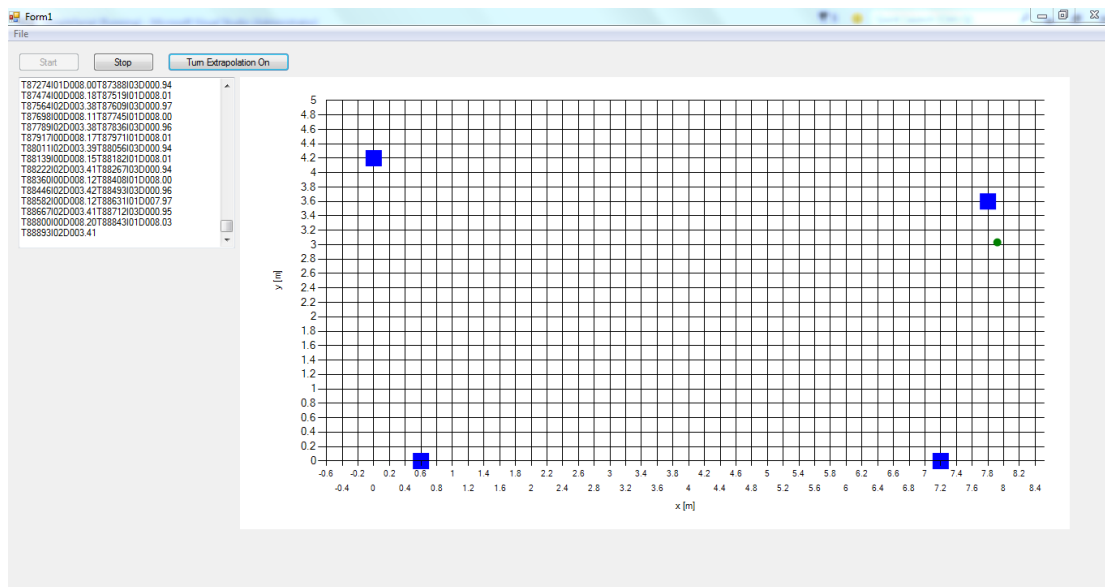
V případě odchylky vzdálenosti v celém prostoru je 95% pravděpodobnost, že se odhadovaná pozice nenachází dál než 33,2 cm. Pokud vzdálenost určujeme jen v horizontální rovině, odpovídá 95 % pravděpodobnosti vzdálenost 4,31 cm.



Obr. 53 - Distribuční funkce - statický bod

10 C# aplikace

Další část práce byla zaměřena na tvorbu C# aplikace sloužící jednak jako samostatný logger dat – viz kapitola 9.2.2, a jednak jako real-time zobrazovač odhadu pozice pohyblivého bodu. Pohled do C# aplikace je na obr. 54.



Obr. 54 - C# aplikace

10.1 Kontextové menu „File“

Kontextové menu „File“ obsahuje 3 položky:

- „Load Anchors Positions“ – položka slouží k nahrání pozic kotev v txt souboru, přičemž každý řádek odpovídá 1 kotvě. Očekává se seřazení kotev $ID = 0$ pro první řádek, $ID = 1$ pro druhý řádek atd. Souřadnice kotev jsou zadávány ve formátu x, y, z , hodnoty jsou zadány v metrech. Např. pro učebnu T2:B2-825 vypadal txt soubor se souřadnicemi kotev následovně:

```
0.6,0,0.8
0.4.2,0.8
7.2,0,0.8
7.8,3.6,0
```

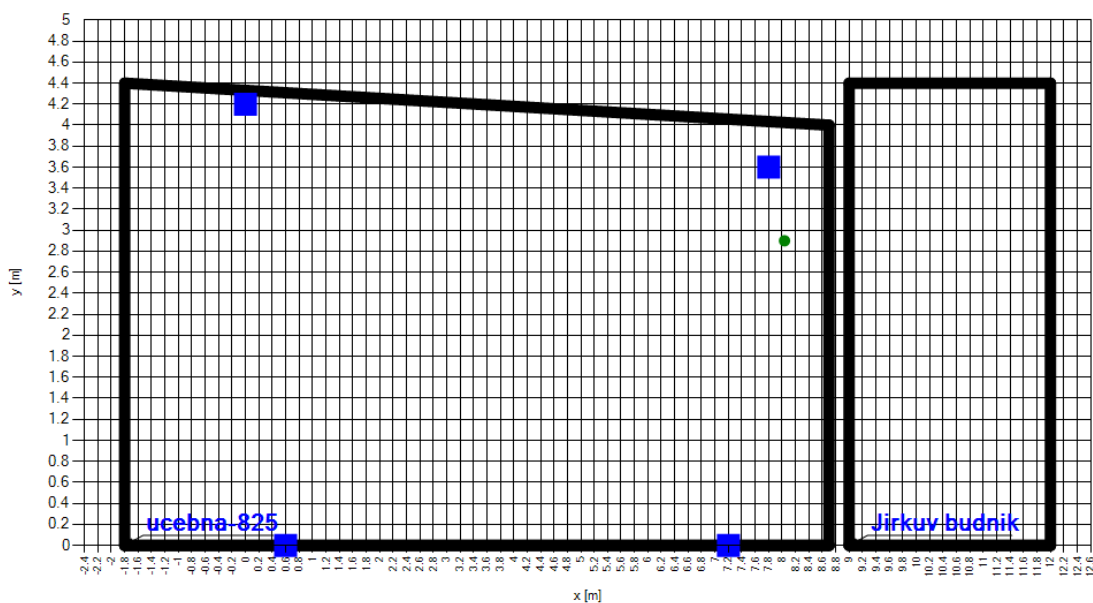
Kotva s $ID = 0$ měla tedy souřadnice $x = 0,6$ m, $y = 0$ m a $z = 0,8$ m. Po správném načtení souřadnic se vykreslí do 2D grafu $[x, y]$ souřadnice jednotlivých kotev. Po nahrání pozice kotev je volána funkce `init_pos = initPosition(ANCHOR_pos)`, která slouží k výpočtu geometrického středu zadaných souřadnic. Pole `init_pos` (obsahující souřadnice $[x, y, z]$) poté slouží jako první odhad pozice po vstupu do implementovaného Levenberg-Marquartova iteračního algoritmu. Nahrání souřadnic kotev do aplikace je povinné.

- „Load Floor Plan“ – položka slouží k načtení půdorysu případných místností a ostatních objektů. Půdorys je nahráván z textového souboru, podobně jako načtení poloh kotev. Očekává se následující formát souboru: Na prvním řádku je očekáváno

jméno objektu. Další řádky poté obsahují x , y souřadnice jednotlivých bodů. Konec objektu signalizuje středník. Pro testovací účely v místnosti T2-B2:825 měl soubor s půdorysem následující podobu:

```
ucebna-825
-1.8,0
-1.8,4.4
8.7,4
8.7,0;
Jirkuv budnik
9,0
9,4.4
12.4,4
12,0;
```

Jména jednotlivých objektů se v C# aplikaci poté zobrazí v levém dolním rohu objektu. Očekává se, že objekt je uzavřená křivka. Ke správnému zobrazení popisku by body měly být zadány pokud možno se vzestupnou x souřadnicí. Situace po vykreslení půdorysu může být pozorována na obr. 55. Nahrání půdorysu do aplikace je nepovinné.



Obr. 55 - Graf s půdorysem

- „Save Data“ – položka slouží k uložení naměřených dat do txt souboru. Je uložen veškerý obsah textového pole. C# aplikace navíc přidává k naměřeným datům časovou značku (rozdílovací schopnost 1 ms).

10.2 Tlačítka aplikace

Aplikace disponuje celkem 3 tlačítky.

10.2.1 Tlačítko start

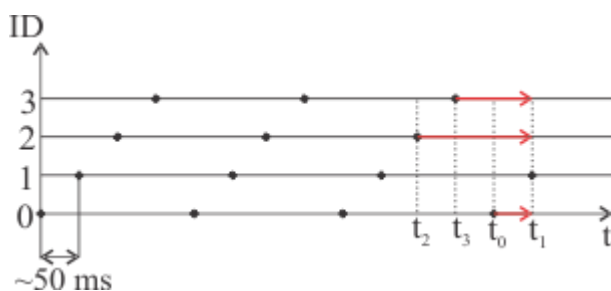
Tlačítko start otevírá příslušný COM port a zapíná časovače určené k měření doby příchodu dat od jednotlivých kotev.

10.2.2 Tlačítko stop

Tlačítko stop zavírá příslušný sériový port. Vzhledem k tomu, že při zavírání sériového portu docházelo poměrně často k nesprávnému ukončení aplikace, bylo nutné sériový port zavírat v novém vlákně.

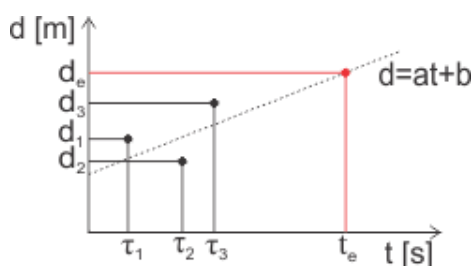
10.2.3 Tlačítko „Turn Distance Extrapolation On/Off“

Tlačítko slouží k zapnutí/vypnutí lineární extrapolace (pomocí metody nejmenších čtverců) vzdálenosti mezi „tagem“ a jednotlivými kotvami. Předpokládáme, že aktuálně (např. v čase t_1 viz obr. 56) přišla vzdálenost od kotvy s $ID = 1$. Od ostatních kotev přišly naměřené vzdálenosti v jiný čas – *někdy* v minulosti (doba mezi jednotlivými příchody odpovídá cca 50 ms). Dle obr. 56 bude platit, že $t_1 = t_0 + \Delta t_{01}$, $t_1 = t_3 + \Delta t_{03}$, $t_1 = t_2 + \Delta t_{02}$, kde časové rozdíly Δt jsou na obr. 56 zobrazeny červeně.



Obr. 56 - Příchody vzdáleností od jednotlivých kotev

Podle naměřených vzdáleností od jedné kotvy (minimálně musí být 3) lze pomocí linearizace problému určit odhad vzdálenosti od této kotvy v libovolném čase. V uvedeném případě (obr. 56) chceme určit (extrapolovat) vzdálenost „tagu“ od kotev ($ID = 0,3,2$) v čase t_1 .



Obr. 57 - Lineární extrapolace (vzdálenost)

Na obr. 57 lze pozorovat extrapolovanou vzdálenost d_e v čase t_e (na obr. 56 odpovídá t_e kýženému času t_1). Předpokládá se rovnoměrný přímočarý pohyb „tagu“ vzhledem k dané kotvě (čímž se dopouštíme značného zjednodušení problému). Vyjádříme-li jednotlivé vzdálenosti pomocí rovnice přímky, dostaneme:

$$\begin{aligned} d_1 &= a\tau_1 + b + \varepsilon_1 \\ d_2 &= a\tau_2 + b + \varepsilon_2 \\ d_3 &= a\tau_3 + b + \varepsilon_3 \end{aligned} \quad (63)$$

Kde ε reprezentuje chybu od proložené přímky. Cílem je odhadnout parametry přímky a a b . Pokud zanedbáme odchylku od proložené přímky, dostaneme:

$$\begin{aligned}d_1 &= a\tau_1 + b \\d_2 &= a\tau_2 + b \\d_3 &= a\tau_3 + b\end{aligned}\tag{64}$$

Což lze přepsat do maticové rovnice:

$$\mathbf{d} = \mathbf{T}\mathbf{p}\tag{65}$$

Kde řádkový vektor $\mathbf{d} = (d_1 \quad d_2 \quad d_3)$, matice $\mathbf{T} = \begin{pmatrix} \tau_1 & 1 \\ \tau_2 & 1 \\ \tau_3 & 1 \end{pmatrix}$ a sloupcový vektor $\mathbf{p} = \begin{pmatrix} a \\ b \end{pmatrix}$. Z maticové rovnice se snažíme osamostatnit sloupcový vektor \mathbf{p} (v prvním kroku vynásobíme celou rovnici zleva transpozicí matice \mathbf{T}):

$$\mathbf{T}^T \mathbf{d} = \mathbf{T}^T \mathbf{T} \mathbf{p}\tag{66}$$

V dalším kroku je již možné osamostatnit \mathbf{p} :

$$\mathbf{p} = (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T \mathbf{d}\tag{67}$$

Čímž jsou určeny parametry přímky. Vzdálenost d_e (obr. 57) lze poté určit z rovnice:

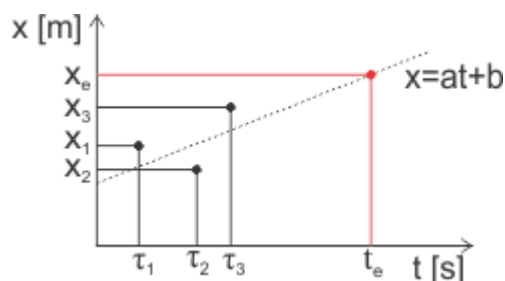
$$d_e = a\tau_e + b\tag{68}$$

Takto jsou vzdálenosti od ostatních kotev přepočítány na stejný čas (na obr. 56 je tímto myšlen čas t_1).

10.2.4 Tlačítko „Turn Position Filtration On/Off“

Tlačítko slouží k zapnutí filtrace odhadu pozice „tagu“, kterou lze rozdělit na 2 kroky. V prvním kroku se provede lineární extrapolace pozice z předchozích odhadů pozice „tagu“ (pokud jsou k dispozici alespoň 3), v druhém kroku je poté proveden aritmetický průměr mezi aktuálním odhadem a provedenou extrapolací.

Extrapolace pozice se provádí obdobně, jako bylo popsáno v předchozí kapitole. Problém byl rozdělen zvlášť podle jednotlivých souřadnic. Extrapolace je tedy prováděna pro souřadnici x, y a z . Na obr. 58 je uveden příklad pro souřadnici x .



Obr. 58 - Lineární extrapolace (pozice)

Obdobným způsobem jako v předchozí kapitole lze provést výpočet odhadu jednotlivých souřadnic:

$$\begin{aligned}
 x_e &= a_x t_e + b_x \\
 y_e &= a_y t_e + b_y \\
 z_e &= a_z t_e + b_z
 \end{aligned}
 \tag{69}$$

Poté je spočten aritmetický průměr těchto (extrapolovaných) hodnot s aktuálně vypočteným odhadem pozice „tagu“:

$$\begin{aligned}
 x &= \frac{x_e + x_{LM}}{2} \\
 y &= \frac{y_e + y_{LM}}{2} \\
 z &= \frac{z_e + z_{LM}}{2}
 \end{aligned}
 \tag{70}$$

Kde x_{LM}, y_{LM}, z_{LM} je pozice „tagu“ určená pomocí Levenberg-Marquartova iteračního algoritmu (viz kapitola 8.1). Po zapnutí filtrace je pohyb „tagu“ viditelně plynulejší.

10.3 Textové pole

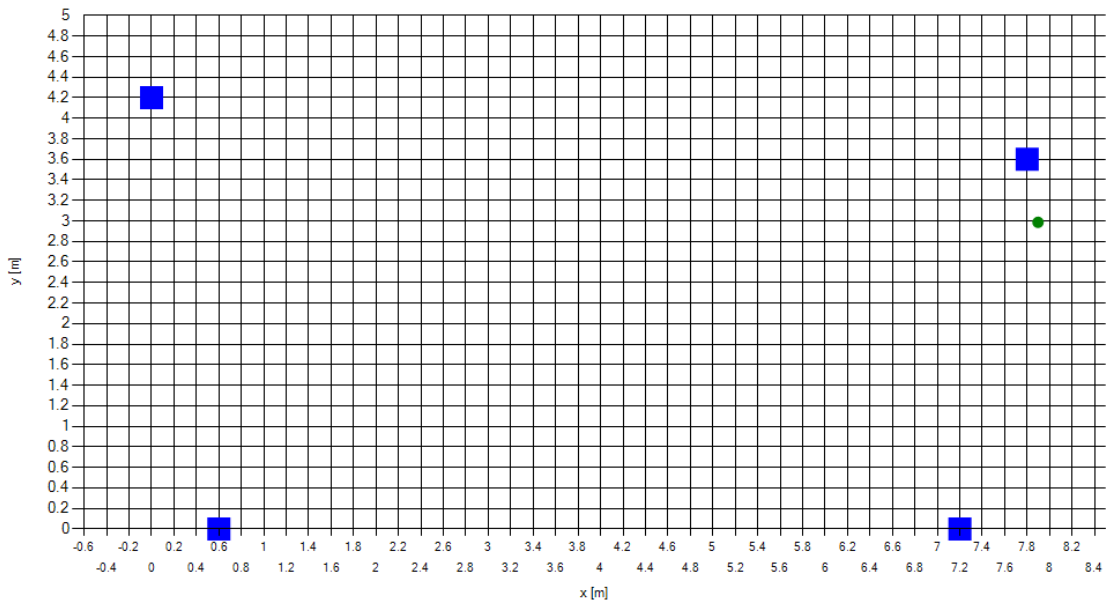
Textové pole zobrazuje naměřená data s přidanou časovou značkou. Formát dat byl popsán v kapitole 9.2.2.

10.4 Graf

2D graf slouží k zobrazení kotev (vykresleny modrými čtverečky) a pozice „tagu“, ten může nabývat 3 barev podle toho, od kolika kotev jsou data přijímána.

- Pozice pohyblivého bodu je zobrazena zeleně v případě, kdy přišly vzdálenosti od všech 4 kotev.
- Pozice pohyblivého bodu je zobrazena oranžově v případě, kdy přišly vzdálenosti jen od 3 kotev.
- Pozice pohyblivého bodu je zobrazena červeně, pokud nepřišel dostatečný počet vzdáleností od různých kotev. Je zobrazena pozice posledního úspěšně provedeného odhadu.
- Pozice pohyblivého bodu je zobrazena fialově, pokud Levenberg-Marquartův iterační algoritmus „nezkonvergoval“ za předem stanovený počet cyklů (`int MAX_COUNT = 20`).

Detailnější pohled do grafu aplikace je na obr. 59. V rámci nastavení grafu byla zapnuta mřížka 20x20 cm.



Obr. 59 - Graf C# aplikace

10.5 Příjem dat od sériové linky

V případě, kdy dojde k příjmu dat od sériové linky (podmínkou je, že sériová linka musí být otevřená, tj. musí být načteny pozice kotev a musí být zmáčknuto tlačítko start), je přijatý `string` uložen znak po znaku do objektu typu `Queue`, který dokáže data frontovat. Následně je volána funkce `this.Invoke(new EventHandler(DisplayText))` pro zobrazení nově přichozích dat do textového pole.

V případě, kdy je přijata zpráva od centrálního uzlu EVK1000 v pořádku (je přijato odřádkování `'\nr'`), je volána funkce `SaveRange()` pro zpracování naměřené vzdálenosti a ID kotvy, od které přišla. Pokud jsou poslední 4 případně alespoň 3 vzdálenosti s různým ID kotvy, je volána funkce `positionEst(d4, -1, 4)`, ve které je implementován Levenberg-Marquartův iterační algoritmus.

Pokud není přesáhnut maximální počet cyklů a chybová vzdálenost vyjde menší, než je stanovená tolerance, je volána funkce `plotTag(base_count)`, která zobrazuje aktuální pozici „tagu“ barvou podle počtu viditelných kotev, jak je popsáno v kapitole 10.4.

10.6 CSML knihovna

Vzhledem k tomu, že Levenberg-Marquartův iterační algoritmus používá maticové operace, bylo nutné přidat do programu maticovou knihovnu. Byla použita CSML knihovna, viz [17]. K této knihovně existuje podrobná dokumentace. Součástí knihovny je v druhé části také polynomiální aritmetika, ta však pro účely C# aplikace nebyla využita.

Knihovnu bylo nutné přidat jako novou referenci k projektu a v kódu bylo nutné kompilátoru oznámit její používání slovy `using CSML`.

Z této knihovny byla využita funkce `Transpose()` sloužící k provedení transpozice matice a dále funkce

```
x = CSML.Matrix.Solve(A_trans * A + eye, A_trans * b)
```

kteřá slouží k výpočtu maticové rovnice. Daný zápis je ekvivalentní s rovnicí

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{b} \quad (71)$$

kde \mathbf{I} je jednotková matice příslušného rozměru.

10.7 Výpočet parametrů DOP

Parametry DOP slouží k výpočtu směrodatné odchylky chyby polohy. K jejich výpočtu slouží kovarianční matice chyby určení polohy:

$$(\mathbf{A}^T \mathbf{A})^{-1} = \begin{pmatrix} XDOP^2 & r_{xy} & r_{xz} \\ r_{yx} & YDOP^2 & r_{yz} \\ r_{zx} & r_{zy} & ZDOP^2 \end{pmatrix} \quad (72)$$

Kde r_{xy} je korelační koeficient chyb souřadnic x a y . V případě nezávislosti platí ($r_{xy} = r_{xz} = r_{yx} = r_{yz} = r_{zx} = r_{zy} = 0$). Parametr $PDOP$ lze poté vypočítat z následujícího vztahu:

$$PDOP = \sqrt{\text{tr}((\mathbf{A}^T \mathbf{A})^{-1})} = \sqrt{XDOP^2 + YDOP^2 + ZDOP^2} \quad (73)$$

Parametr $HDOP$ související s horizontální přesností je poté dán vztahem:

$$HDOP = \sqrt{XDOP^2 + YDOP^2} \quad (74)$$

A konečně parametr $VDOP$ lze spočítat pomocí:

$$VDOP = \sqrt{ZDOP^2} = ZDOP \quad (75)$$

Směrodatná odchylka chyby polohy ve třírozměrném prostoru je poté dána vztahem:

$$\sigma_P = \sigma_d PDOP \quad (76)$$

Podobně lze určit směrodatnou odchylku chyby určení polohy v horizontální rovině:

$$\sigma_H = \sigma_d HDOP \quad (77)$$

A nakonec může být určena směrodatná odchylka svislé polohy:

$$\sigma_V = \sigma_d VDOP = \sigma_d ZDOP \quad (78)$$

Parametr σ_d má význam směrodatné odchylky měření vzdálenosti a pro systém DecaWave byl stanoven na 20 cm (`double sigma_d = 0.2`)

11 Programové vybavení

V rámci praktické části diplomové práce bylo použito několik programů pro příjem dat, zpracování dat, úpravy zdrojového kódu, „přeflashování“ paměti EVK1000, analýzu naměřených dat apod. Programové vybavení je shrnuto v následující tabulce.

| Program | Verze | Použití |
|-----------------------|------------------------------|---|
| Matlab | R2014b | Následné zpracování naměřených dat, Levenberg-Marquardtův iterační algoritmus, analýza výpadků konektivity od jednotlivých kotev |
| RealTerm | 2.0.0.7.0 | Příjem dat přicházejících ze sériové linky (resp. VCP), přidání časové značky typu Matlab (počet dnů od počátku našeho letopočtu) |
| Eclipse | Luna Service Release (4.4.1) | Úpravy a tvorba nového zdrojového kódu do vývojové sady EVK1000 |
| STM32 ST-LINK Utility | 3.7.0 | „Flashování“ paměti EVK1000 |
| Open OCD | 0.9.0. | „Debugování“ vývojové sady EVK1000 v reálném čase |
| Visual Studio | 4.6.01055 | Tvorba aplikace v jazyku C# (lokalizace, příjem dat) |

Tab. 12 - Programové vybavení

K rychlé analýze naměřených dat byl dále použit program PSPad editor. Dále pak byly použity další podpůrné programy, které však oproti vyjmenovanému vybavení nemají zásadní význam.

12 Seznam příloh

V následující tabulce je uveden seznam příloh. Jsou zde obsaženy naměřené hodnoty, skripty v programu Matlab R2014b, C# aplikace apod. Přílohy byly uloženy na CD, které je součástí práce.

| Název souboru/složky | Cesta | Popis |
|----------------------|------------------|--|
| CRLB | Kořenový adresář | Složka obsahuje skripty napsané v programu Matlab R2014b sloužící k analýze Cramer-Raovy spodní meze pro jednotlivé metody odhadu. |
| ED_2016_02_15 | Kořenový adresář | Složka obsahuje všechna naměřená data, skripty v programu Matlab, obrázky apod. z měření systému DecaWave 15. 2. 2016 ve venkovním |

| | | |
|--|---|--|
| | | prostředí v blízkosti FEL ČVUT v Praze. |
| dw_ENU_kulatak_4.m | ED_2016_02_15\ DW_processed | Skript v programu Matlab R2014b zpracovávající naměřené vzdálenosti od jednotlivých kotev. Skript provádí analýzu výpadků konektivity, počítá odhad pozice „tagu“ pomocí Levenberg-Marquartova algoritmu a zobrazuje tyto odhady v grafu. Výstupem skriptu je kml soubor, po jehož otevření lze sledovat trajektorii „tagu“ promítnutou do Google map. |
| capture1.txt capture2.txt capture3.txt capture4.txt | ED_2016_02_15\ DW_processed\ zaznam | Soubory obsahují naměřené vzdálenosti od jednotlivých kotev programem RealTerm s přidanými časovými značkami typu Matlab. |
| RTK_data | ED_2016_02_15\ RTK_data | Složka obsahuje RTK metodou naměřená data pozic kotev (pro možnost promítnutí dat do Google map) a trajektorii pohyblivého bodu (pro porovnání) |
| RTK_processed | ED_2016_02_15\ RTK_processed | Složka obsahuje Matlab skripty pro zpracování naměřených dat metodou RTK a vygenerované souřadnice v kml souborech. |
| EU_2016_02_17 | Kořenový adresář | Složka obsahuje všechna naměřená data, skripty v programu Matlab, obrázky apod. z měření systému DecaWave 17. 2. 2016 v učebně T2:B2-309 budovy FEL ČVUT v Praze. |

| | | |
|--|--------------------------|--|
| dw_ucebna_309.m | EU_2016_02_17 | Skript v programu Matlab sloužící k analýze výpadků konektivity od jednotlivých kotev, počítání odhadu trajektorie „tagu“, statistickému zpracování odhadů pozic v případě statického bodu. Skript je určen pro zpracování dat z vnitřních prostor – neprovádí se export do kml souboru. |
| capture1.txt staticky_bod.txt capture3.txt | EU_2016_02_17\ zaznam | Soubory obsahují naměřené vzdálenosti od jednotlivých kotev programem RealTerm s přidávanými časovými značkami typu Matlab. |
| Penetrace | Kořenový adresář | Složka obsahuje skript v programu Matlab, který slouží k analýze směrodatné odchylky určení doby šíření signálu v případě nedodržení přímé viditelnosti. |
| Signaly | Kořenový adresář | Složka obsahuje skripty v programu Matlab R2014b sloužící k analýze používaných impulsů, kanálové kapacity a vykreslení frekvenčního průběhu regulací. |
| SimpleSerialCS | Kořenový adresář | Složka obsahuje projekt v programu Visual Studio napsaný v jazyku C#. Aplikace slouží k zobrazování pozice „tagu“ v reálném čase, výpočtu parametrů DOP, příjmu a zpracování naměřených vzdáleností od jednotlivých kotev, zobrazení půdorysu budovy apod. |
| Standardy | Kořenový adresář | Složka obsahuje kompletní generátor UWB signálu |

| | | |
|--|--|----------------------------------|
| | | (soubor IEEE802154a_signal.m) |
|--|--|----------------------------------|

Tab. 13 - Seznam příloh

Neveřejnou přílohou je zdrojový kód k EVK1000 a dokumentace k němu.

13 Závěr

Tato práce se ve své úvodní části věnovala teoretickému rozboru UWB technologie a popisu standardu IEEE 802.15.4a. V programu Matlab R2014b byl následně vytvořen skript, který generuje UWB signál v souladu s tímto standardem. Další část práce byla zaměřena na charakteristiku impulsů používaných v UWB technologii. Dále byla v rámci teoretické části práce provedena analýza kanálové kapacity. Vzhledem ke značné šířce pásma této technologie bylo dále nutné se zaměřit na limity vyzařování UWB signálu.

Další část práce se zabývala určením chyby vzdálenosti v případě průchodu signálu skrze vložený materiál (zdivo). Ukázalo se, že chyba bude tím větší, čím větší bude procentuální zastoupení zdiva na vzdálenosti vysílač – přijímač. V této práci bylo dále odvozeno, že relativní chyba je úměrná odmocnině z podílu relativní permitivity zdiva a okolního vzduchu. V případě, že je 100% materiálu na cestě mezi vysílačem a přijímačem vyplněno jiným materiálem, než je vzduch, dochází také k největší relativní odchylce určení vzdálenosti. I když lze dosáhnout poměrně vysokých relativních odchylek, je dobré podotknout, že k nim dochází zpravidla na malých vzdálenostech (v provedené simulaci je nejmenší vzdáleností mezi vysílačem a přijímačem 20 cm). Absolutní chyba určení vzdálenosti vykazuje v nejhorším analyzovaném případě ($\epsilon_{r_{wall}} = 10$) 43,23 cm.

V další části práce je poté nastíněna metoda odhadu relativní permitivity zdiva využívající právě chyby určení vzdálenosti. Kapitola 7.2 obsahuje výsledný vztah včetně jeho odvození. Zajímavé by mohlo být srovnání s ostatními používanými metodami pro získání parametrů zdiva.

Další sekce teoretické části práce byla zaměřena na rozbor metod lokalizace pomocí UWB signálu. Ukázalo se, že vliv šířky pásma na Cramer-Raovu spodní mez je příznivý zejména u metod založených na určení doby šíření signálu (případně u úhlové navigace).

Následně již byla práce zaměřena na popis vývojové sady EVK1000. V první části praktické práce bylo nutné implementovat VCP/USB knihovnu do EVK1000 pro možnost komunikace s PC. Následně byl zdrojový kód upraven tak, aby byla umožněna komunikace mezi více uzly. Dále byl pak vytvořen nový typ uzlu typu „CTU listener“, který byl nastaven v režimu neustálého příjmu a sloužil jen jako komunikační uzel pro posílání dat do PC.

Další část práce byla poté zaměřena na implementaci Levenberg-Marquartova algoritmu pro účely určení polohy „tagu“. Algoritmus byl zpočátku testován a používán v programu Matlab R2014b v rámci následného zpracování naměřených vzdáleností od jednotlivých kotev, dále byl implementován i v samotné C# aplikaci pro účely zobrazení pozice v reálném čase.

Zásadní částí práce bylo poté testování navrženého systému. Ukazuje se, že limitujícím faktorem je zde zejména dosah systému a dodržení podmínky nezastínění přímé viditelnosti mezi „tagem“ a kotvami případně centrálním uzlem - „CTU listener“. Dosah systému se v testovaném prostředí pohyboval do cca 100 m.

Poslední část práce byla zaměřena na vývoj C# aplikace zobrazující polohu „tagu“ v reálném čase. Kromě toho aplikace umožňuje pod tlačítkem spuštění filtrace, jejíž aktivace eliminuje případně větší odchylky iteračního algoritmu. Aplikace dále umožňuje nahrání půdorysu daného objektu pro případnou možnost detekce místnosti, v níž se „tag“ nachází.

Vzhledem k tomu, že přesnost určení polohy je určena rozmístěním kotev v prostoru, kalkuluje aplikace dále i parametry DOP, z nichž jsou počítány přesnosti určení polohy v jednotlivých rovinách a souřadnicí.

14 Seznam zkratek

| | | | |
|------|--|---|---|
| AoA | "Angle of Arrival" - směr příjmu signálu, 24, 26, 27, 28 | "Medium Access Control" - Vrstva definující přístup k médium, 9 | |
| AWGN | "Additive White Gaussian Noise" - aditivní bílý gausovský šum, 19 | PHY | "Physical layer" - Fyzická vrstva, 9, 10 |
| BPM | "Burst position modulation" - Modulace, ve které je manipulováno s pozicí impulsu v časové oblasti, 10, 11, 12 | PRF | "Pulse repetition frequency" - Opakovací frekvence impulsu, 11 |
| BPSK | "Binary phase shift keying" – Binární klíčování fázovým posunem, 10, 11 | RSS | "Received Signal Strength" - úroveň přijímaného signálu, 22, 23 |
| BW | "Bandwidth" - šířka pásma, 9, 27 | RTK | "Real Time Kinematic" - metoda zlepšující přesnost určení polohy navigačního systému, 38, 39, 42, 61 |
| CRLB | "Cramer-Rao Lower Bound" - Cramer-Raova spodní mez, 22, 23, 24, 25, 26, 27, 28 | SHR | "Synchronization header" - Hlavička synchronizace, 10 |
| DOP | "Dilution of Precision" - snížení přesnosti, 59, 63 | SNR | "Signal to Noise Ratio" - Odstup signálu od šumu, 19, 23, 27 |
| FCC | Federal Communications Commission, 17, 20 | TDoA | "Time Difference of Arrival" - časový rozdíl příjmu signálů, 24 |
| HDR | "Header" - Hlavička, 10 | ToA | "Time of Arrival" - čas příjmu signálu, 23, 24, 26 |
| LOS | "Line of Sight" - přímá viditelnost, 38 | TOF | "Time of Flight" - doba šíření signálu, 29, 30 |
| LSB | "Least significant byte" - Nejméně významný bajt, 11 | UWB | "Ultra-Wideband" - Širokopásmová technologie, 1, 2, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 28, 30, 35, 62, 63 |
| MAC | | | |

15 Citovaná literatura

- [1] Z. Sahinoglu, S. Gezici a G. Ismail, Ultra-wideband Positioning Systems, Cambridge: Cambridge university press, 2012.

- [2] IEEE, „802.15.4-2011 - IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),“ 16. června 2011. [Online]. Available: <http://standards.ieee.org/findstds/standard/802.15.4-2011.html>. [Přístup získán 9. prosince 2014].
- [3] H. Sheng, P. Orlik, A. M. Haimovich, L. J. Cimini a J. Zhang, „On the Spectral and Power Requirements for Ultra-Wideband Transmission,“ IEEE International Conference on Communications (ICC), Broadway, Cambridge, Massachusetts, 2003.
- [4] M. Brookes, „6: Fourier Transform,“ 2014. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/courses/E1Fourier/00600_FourierTransform.pdf. [Přístup získán 14. srpna 2015].
- [5] M. Ghavami, L. B. Michael a R. Kohno, „Ultra wideband signals and systems in communication engineering,“ Wiley, Chippenham, 2004.
- [6] J. Sýkora, Teorie digitální komunikace, Praha: Vydavatelství ČVUT, 2002.
- [7] P. Pechač a S. Zvánovec, Základy šíření vln pro plánování pozemních rádiových spojů, Praha: BEN, 2007.
- [8] M. Yavari a B. G. Nickerson, „Ultra Wideband Wireless Positioning Systems,“ Faculty of Computer Science University of New Brunswick, Fredericton, 2014.
- [9] C. Centile, N. Alsindi, R. Raulefs a C. Teolis, „Geolocation Techniques Principles and Applications - Chapter 2 Ranging and Localization in Harsh Multipath Environments,“ Springer, New York, 2013.
- [10] L. Mailaender, „ON THE CRLB SCALING LAW FOR RECEIVED SIGNAL STRENGTH (RSS) GEOLOCATION,“ LGS Innovations, New Jersey, 2012.
- [11] IEEE, „IEEE 802.15.4a channel model - final report,“ IEEE, 2005.
- [12] F. Vejražka, Signály a soustavy, Praha: ČVUT, 1992.
- [13] Deca Wave, „ScenSor EVK1000 Evaluation Kit,“ 2015. [Online]. Available: <http://www.decawave.com/products/evk1000-evaluation-kit>. [Přístup získán 19. června 2015].
- [14] ST, „UM1021 User manual,“ březen 2012. [Online]. Available: <http://www.st.com/web/en/catalog/tools/FM147/CL1794/SC961/SS1743/LN1734/PF257882>. [Přístup získán září 2014].
- [15] Deca Wave, „DW1000 USER MANUAL,“ 31. července 2014. [Online]. Available: <http://www.decawave.com/support>. [Přístup získán 10. prosince 2014].
- [16] M. Černý, „Miroslav Černý - personal homepage,“ [Online]. Available: <http://physics.fme.vutbr.cz/~mcerney/vyuka/chyby.pdf>. [Přístup získán 29. března 2016].

- [17] CODE PROJECT, „C# Matrix Library,“ 28. listopadu 2007. [Online]. Available: <http://www.codeproject.com/Articles/19032/C-Matrix-Library>. [Přístup získán 11. března 2016].
- [18] IEEE International Conference on Communications (ICC), „On the Spectral and Power Requirements for Ultra-Wideband Transmission,“ Mitsubishi Electric Research Laboratories, Inc., Cambridge, 2003.
- [19] M. Leixner, „NÁVRH PAKETOVÉHO ANALYZÁTORU PRO UWB PÁSMO DLE STANDARDU IEEE 802.15.4A,“ VYSOKÉ UCENÍ TECHNICKÉ V BRNĚ, Brno, 2014.
- [20] P. Protiva, J. Mrkvica a J. Macháč, „Estimation of Wall Parameters From Time-Delay-Only Through-Wall Radar Measurements,“ IEEE, Praha, 2011.
- [21] A. Safaai-Jazi a A. Muqaibel, „A New Formulation for Characterization of Materials Based on Measured Insertion Transfer Function,“ IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 2003.
- [22] A. Safaai-Jazi, S. M. Riad, A. Muqaibel a A. Bayram, „Ultra-wideband Propagation Measurements and Channel Modeling,“ DARPA NETEX, Virginia, 2002.
- [23] V. Schejbal, P. Bezoušek, D. Čermák, Z. Němec a O. Fišer, „UWB Propagation through Walls,“ RADIOENGINEERING, Pardubice, 2006.
- [24] Deca Wave, „DW1000 Data Sheet,“ 30. září 2014. [Online]. Available: <http://www.decawave.com/support>. [Přístup získán 10. prosince 2014].
- [25] Deca Wave, „DW1000 USER MANUAL,“ 13. května 2015. [Online]. Available: <http://www.decawave.com/sites/default/files/resources/dw1000-datasheet-v2.04.pdf>. [Přístup získán 13. května 2015].
- [26] IEEE, „Ultrawideband through-the-wall propagation,“ 6. prosince 2005. [Online]. Available: http://www.researchgate.net/publication/224625127_Ultrawideband_through-the-wall_propagation/links/09e4150e58acf0761b000000. [Přístup získán 19. prosince 2014].
- [27] B. Mike, „6: Fourier Transform,“ 2014. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/courses/E1Fourier/00600_FourierTransform.pdf. [Přístup získán 14. srpna 2015].