VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

VÝVOJ ALGORITMU PRO AUTOMATICKÉ ROZMÍSŤOVÁNÍ REFERENČNÍCH JEDNOTEK BEZDRÁTOVÉHO POZIČNÍHO SYSTÉMU

ALGORITHM DEVELOPMENT FOR AUTOMATIC PLACEMENT OF POSITIONING SYSTEM REFERENCE UNITS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. DAVID PRŮDEK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. VÁCLAV ŠIMEK**
SUPERVISOR

BRNO 2022

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2021/2022

Zadání diplomové práce

Student: **Průdek David, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Kybernetická bezpečnost
Název: **Vývoj algoritmu pro automatické rozmísťování referenčních jednotek bezdrátového pozičního systému**
Algorithm Development for Automatic Placement of Positioning System Reference Units
Kategorie: Algoritmy a datové struktury

Zadání:

1. Zabývejte se problematikou určování polohy v rámci lokalizačního systému pomocí metod měření vzdáleností (TWR) a rozdílu času (TDoA).
2. Podrobně se seznamte s poziční platformou Sewio na bázi technologie UWB (širokopásmová komunikace) a stávajícími algoritmy pro výpočet polohy na základě multilaterace.
3. Na základě praktických experimentů vyhodnoťte míru vzájemné viditelnosti mezi referenčními jednotkami v lokačním systému SEWIO s využitím technik dle bodu 1) zadání.
4. Navrhněte algoritmus pro automatické rozmísťování referenčních jednotek lokalizačního systému Sewio na základě vzájemného měření vzdálenosti pro všechny režimy pozicování v dané platformě (1D, 2D).
5. Proveďte praktické ověření parametrů algoritmu dle bodu 4) zadání. Pokuste se přístup modifikovat i pro členitější vnitřní prostory a rozšířit o detekci chybného zaměření referenčních jednotek.
6. Zhodnoťte dosažené výsledky a pokuste se navrhnout případná vylepšení či další směry vývoje.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 2 zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 18. května 2022
Datum schválení: 29. října 2021

Abstrakt

Tato diplomová práce se zabývá návrhem a implementací algoritmu pro automatické rozmisťování referenčních jednotek real-time lokalizačního systému. Tento algoritmus je určen pro RTLS firmy Sewio Networks. Součástí práce je i průzkum lokalizačních technik v senzorových sítích se zaměřením na bezdrátovou technologii UWB. Práce popisuje způsob použití navrženého systému a fungování jednotlivých částí důležitých pro správný výsledek. Dále jsou v této práci popsány implementační detaily a způsob integrace vyvíjené funkce do systému Sewio RTLS. Poslední část práce pojednává o testování vytvořeného systému a navrhuje možnosti jeho dalšího vylepšení.

Abstract

This diploma thesis describes designing and implementing the algorithm for automatic placement of the real-time localization system reference units. The algorithm is supposed to work with RTLS of the company Sewio Networks. The content of this thesis covers localization options in sensor networks and it is focused on UWB wireless technology. This thesis defines the use case of the designed feature and the principles of the most important parts for the correct outcome. Furthermore, the thesis contains implementing details and the way of integration into the Sewio RTLS. The last part presents the testing results of the created function and proposes future improvements.

Klíčová slova

RTLS, UWB, TWR, Sewio, lokalizace, kotvy

Keywords

RTLS UWB, TWR, Sewio, localization, anchors

Citace

PRŮDEK, David. *Vývoj algoritmu pro automatické rozmísťování referenčních jednotek bezdrátového pozičního systému.* Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Vývoj algoritmu pro automatické rozmísťování referenčních jednotek bezdrátového pozičního systému

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Václava Šimka. Potřebné informace mi poskytl Ing. Ľubomír Mráz. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
David Průdek
17. května 2022

Poděkování

Děkuji vedoucímu práce Ing. Václavu Šimkovi za vedení a cenné rady při tvorbě mé diplomové práce. Dále děkuji Ing. Ľubomíru Mrázovi, Ing. Tomáši Kočanovi a celé společnosti Sewio Networks s.r.o. za téma, konzultace a poskytnuté technické a datové zdroje.

Obsah

1 Úvod **3**

2 Principy určování polohy pomocí pasivních senzorových sítích **4**
2.1 Výhody UWB technologie z hlediska lokalizace 5
2.2 Algoritmy určování pozice 6
2.2.1 Triangulace 6
2.2.2 Trilaterace 7
2.2.3 Multilaterace 8
2.2.4 Další algoritmy určování pozice 8
2.3 Techniky lokalizace 8
2.3.1 TDOA 8
2.3.2 TWR 9

3 Lokalizační platforma Sewio RTLS **11**
3.1 Hardwarové komponenty systému 11
3.1.1 Kotva 11
3.1.2 Tag 12
3.2 Lokalizační server RTLS Studio 13
3.2.1 Anchor Auto-Deployment 15
3.3 Lokalizace v platformě Sewio 15
3.3.1 2D 15
3.3.2 1D 17
3.3.3 0D 18

4 Využití metody TWR pro měření vzdáleností **19**
4.1 Výsledky TWR z 2D instalace 19
4.1.1 Prostředí 20
4.1.2 Příprava měření 20
4.1.3 Analýza výsledků 21
4.1.4 Zhodnocení 23
4.2 Výsledky TWR z 1D instalace 24
4.2.1 Prostředí 24
4.2.2 Analýza výsledků 25
4.2.3 Zhodnocení výsledků 26

5 Koncepce automatického rozmisťování referenčních jednotek **28**
5.1 Detekce nesprávného zaměření kotev 28
5.1.1 Nepřesné vzdáleností mezi kotvami 29

5.1.2 Prohozené kotvy 29
5.2 Automatické rozmisťování kotev pro 1D 30
5.2.1 Návrh použití v RTLS Studio 30
5.3 Automatické rozmisťování kotev pro 2D 31
5.3.1 Limitace Auto-Deployment gen1 31
5.3.2 Návrh změn pro Auto-Deployment gen2 33

6 Implementace **36**
6.1 Detekce nesprávného zaměření kotev 36
6.2 Automatické rozmisťování kotev pro 1D 39
6.3 Automatické rozmisťování kotev pro 2D 41

7 Testování a zhodnocení funkce **44**
7.1 Ověření detekce prohozených kotev 44
7.2 1D Auto-Deployment 46
7.3 2D Auto-Deployment 46
7.3.1 Kompenzace výškového rozdílu 47
7.3.2 Filtr nepřesných TWR měření 47
7.3.3 Umísťování kotev rozsáhlých instalací 48
7.4 Limitace a možná vylepšení 51

8 Závěr **52**

Zkratky **53**

Literatura **54**

A Auto-Deployment aktivity diagram **55**

B Obsah přiloženého paměťového média **56**

Kapitola 1

Úvod

Znát místo děje, je pro dnešní rychlý a globalizovaný svět nesmírně důležité. Lidé si v běžném životě vystačí se znalostí, kde se právě nachází a jak se dostat ke svému cíli. Toto dokážou s potřebnou přesností nabídnout globální poziční systémy, například GPS, GLONASS anebo evropské Galileo. Znalost aktuální pozice je užitečná nejen v osobním životě, ale zejména v průmyslové výrobě, logistice nebo například i v zábavním odvětví.

Lokalizace v nejrůznějších prostředích může přinést zvýšenou efektivitu, pomoc při rozhodování a automatizaci operací, vyšší bezpečnost personálu i materiálových zdrojů a nakonec i celkovou vyšší úroveň zabezpečení. Lokalizace uvnitř budov může být implementována různými technologiemi, například Wi-Fi a Bluetooth Low Energy (BLE), avšak tyto technologie nebyly přímo určeny pro tyto účely a tak nenabízí potřebnou přesnost a škálovatelnost [5]. Z tohoto důvodu se v dnešní době aplikují do praxe technologická řešení využívající Ultra-wideband (UWB), mezi něž také patří lokalizační systém Sewio.

Takový systém vyžaduje referenční jednotky zvané kotvy, které je nutné zaměřit s přesnosti na centimetry. V instalacích s vysokými desítkami či stovkami kotev může být instalace systému pro všechny účastníky velmi náročná. Cílem této práce je vytvořit systém, který automaticky zaměří tyto referenční jednotky pro maximální počet typů instalací a případů užití.

V první části této práce jsou představeny možnosti lokalizace v senzorových sítích. Kapitola srovnává různé bezdrátové technologie vhodné pro tyto účely se zaměřením na standard UWB, který je použit v lokalizační platformě Sewio. Dále tato kapitola blíže specifikuje různé algoritmy a techniky lokalizace. Některé z technik popsaných v této kapitole budou využity ve fázi návrhu koncepce systému pro zaměřovaní referenčních jednotek a jeho implementaci. Následující kapitola 3 představuje real-time lokalizační platformu firmy Sewio, pro kterou je vyvíjená funkcionalita této práce určena. V této části jsou popsány hardwarové a softwarové prvky tohoto lokalizačního systému.

Obsah kapitoly 4 pojednává o provedeném měření v reálné instalaci RTLS pro získání prvotních dat, která jsou základem pro práci vyvíjeného algoritmu. Smyslem měření je získat povědomí o schopnostech lokalizační platformy a vybrané techniky pro měření vzdáleností. Tyto poznatky jsou dále uplatněny v kapitole 5. Ta popisuje účel použití jednotlivých částí vyvíjeného systému a přináší řešení problémů, které vyvstaly v průběhu měření a návrhu funkce. Popis implementace návrhu se nachází v kapitole 6. Pro každou část navrženého systému je v této kapitole představeno, jakým způsobem jsou jednotlivé problémy implementačně řešeny. Dále se kapitola věnuje integraci navržených a implementovaných funkcí do celého systému Sewio RTLS. Reálná funkce implementovaného systému je ověřena v kapitole 7.

Kapitola 2

Principy určování polohy pomocí pasivních senzorových sítích

Tato kapitola se zabývá metodami pasivní lokalizace pomocí senzorových sítí a srovnáním systémů z kategorie Indoor positioning system (IPS). Tyto systémy lze klasifikovat do 4 kategorií na základě použité technologie pro určování pozice. Mezi tyto kategorie patří Radio Frequency (RF) systémy, technologie založené na světle, zvuku a magnetickém poli. Každá z těchto technologií má své výhody i nevýhody. RF systémy obecně přinášejí nižší pořizovací náklady, avšak napříč těmito systémy najdeme mnoho odlišností týkající se rádiových a fyzikálních možností dané technologie [3].

V dnešní době nejrozšířenějšími lokalizačními systémy na světe jsou Global Navigation Satellite System (GNSS) a konkrétně Global Positioning System (GPS). Satelity systému GPS posílají své orbitální informace GPS přijímačům a ty na základě přijatých signálů z několika satelitů vypočítají svou polohu. Poloha je vypočtena pomocí reverzního Time Difference of Arrival (TDOA), které je vysvětleno dále v této kapitole. Hlavní limitací GPS je požadavek na Line of sight (LOS), který zajišťuje přímou viditelnost mezi satelitem a přijímačem a z toho důvodu je tento systém nevhodný pro vnitřní prostory [3].

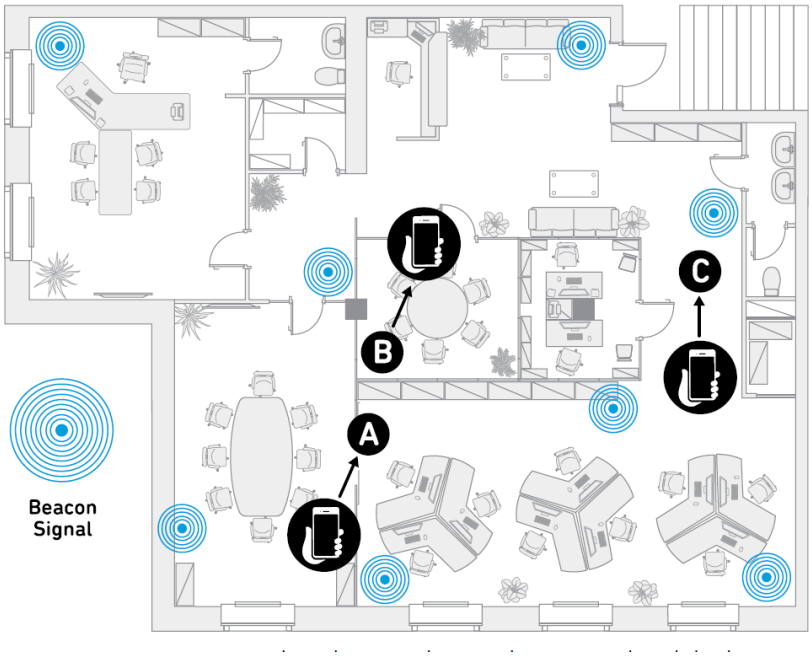
Mezi rádiové systémy pro lokalizaci ve vnitřních prostorech patří zejména UWB, Wi-Fi, BLE a Radio-frequency identification (RFID). Systémy Wi-Fi a BLE nebyly původně navrženy pro účely lokalizace. Díky jejich rozšířenosti v nejrůznějších typech zařízení však získaly určité využití v oblasti určování polohy. Lokalizace pomocí BLE spočívá v hledání zařízení označovaných jako beacon[1] a detekci přiblížení k těmto zařízením pomocí měření síly signálu Received Signal Strength Indication (RSSI) [5]. U BLE jsou snahy pro využití v lokalizaci o něco větší, jelikož vylepšení samotné technologie pro tyto účely jsou zanášeny do samotné specifikace Bluetooth[2], v podobě podpory funkce Angle of Arrival (AoA), anebo jsou vytvářeny jiné standardy pro aplikaci beaconu pomocí technologie BLE, například iBeacon[3].

Na druhou stranu UWB je standard IEEE 802.15.4a/z navržený pro aplikace lokálního určování pozice. Zajišťuje vysokou přesnost pozice udávanou v jednotkách centimetrů a dále pak také vysokou spolehlivost, jež zajišťuje odolnost vůči odrazům rádiových signálů.

[1] maják, označení pro referenční zařízení v BLE lokalizaci

[2] https://www.bluetooth.com/specifications/bluetooth-core-specification/

[3] https://developer.apple.com/ibeacon/



Obrázek 2.1: Lokalizace mobilních telefonů pomocí BLE beaconů. Převzato z [5].

2.1 Výhody UWB technologie z hlediska lokalizace

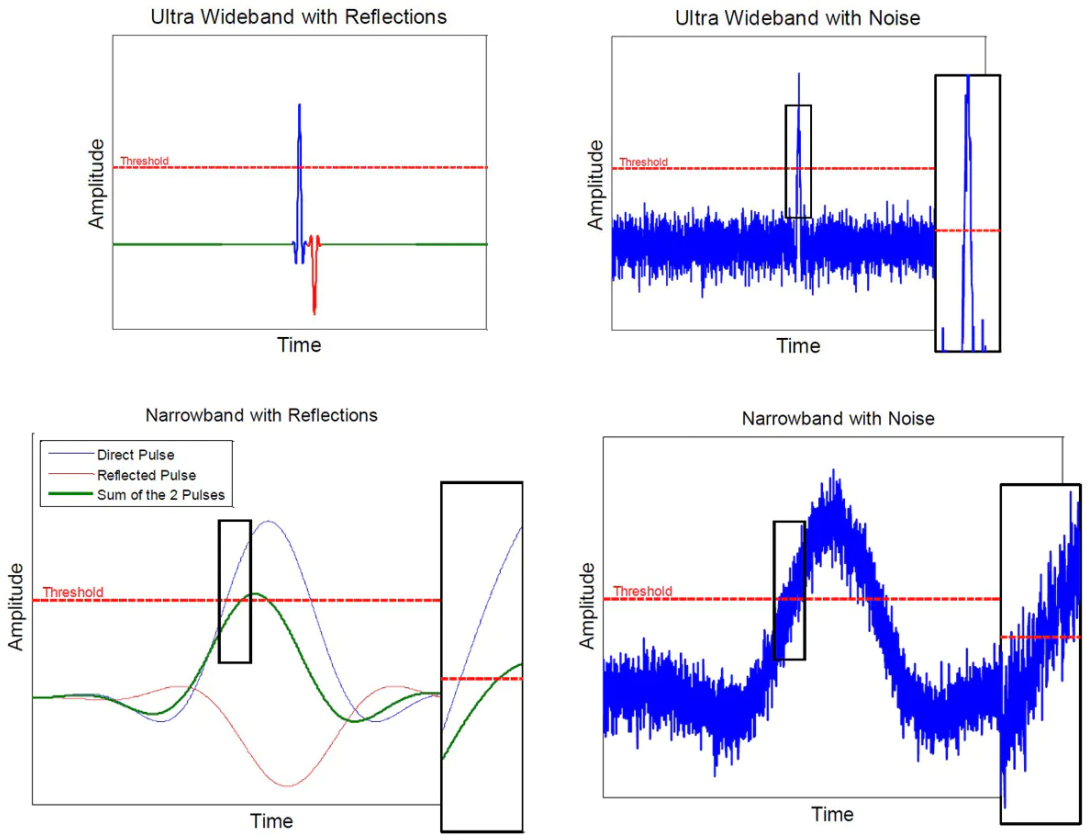
UWB je od roku 2002 pro komerční použití definované FCC regulacemi jako radiová technologie pracující ve frekvenčním pásmu mezi 3,1 GHz až 10,6 GHz s minimální šířkou pásma 500 MHz [4].

Oproti amplitudové a frekvenční modulaci pro přenos informace, UWB využívá sekvence velmi úzkých pulzů modulovaných kombinací Burst position modulation (BPM) a Binary phase-shift keying (BPSK). Každý symbol je složen z aktivního shluku UWB pulzů, který může mít variabilní délku a podporuje také variabilní rychlost přenosu. Tento způsob modulace pro přenos dat pomáhá zmírnit citlivost vůči rušení od jiných narrowband[4] signálů a zvyšuje schopnost vypořádat se odraženými signály [4].

Obrázek 2.2 znázorňuje tyto rádiové jevy pro UWB i narrowband signál. V případě rušeného signálu šumem, rychlá nástupná hrana úzkého pulzu UWB dokáže přesně a spolehlivě překročit amplitudový práh, což umožní korektní změření času příchodu signálu. Šum u pomalejší nástupné hrany narrowband signálu může ovlivnit signál natolik, že překročení prahu bude detekováno příliš brzo anebo pozdě. Reflektovaný signál je fázově i časově posunutý, což může pomalý pulz také ovlivnit, ale u úzkého pulzu je tato šance mnohem menší [2].

Hlavním rozdílem oproti jiným radiovým řešením pro lokalizaci ve vnitřních prostorech je skutečnost, že UWB nepoužívá hodnotu síly signálu RSSI k určení vzdálenosti od měřeného bodu. Problém s RSSI je fakt, že síla signálu zvyšováním vzdálenosti od zdroje signálu intenzivně klesá s druhou mocninou. Tento fyzikální jev je nazván Zákon převrácených čtverců. Rovnice 2.1 popisuje tento fyzikální jev. Proměnná Q značí sílu působící na elektrický náboj e_0 a r reprezentuje rádius od zdroje signálu [5].

[4]šířka pásma zprávy nepřesahuje šířku pásma koherence kanálu



Obrázek 2.2: Srovnání chování ultra-wideband a narrowband signálu při odrazu a rušení signálu. Převzato z [2].

$$E = \frac{Q}{4\pi e_0 r^2} \tag{2.1}$$

Se ztrátou intenzity signálu se zhoršuje schopnost přesného měření hodnoty RSSI. Ta může výrazně kolísat v závislosti vlivu prostředí na rádiové vlny. Pro zmírnění nedostatku přesného měření hodnoty RSSI se využívá metoda fingerprinting. Jednotlivé beacony si mezi sebou změří sílu signálu, z čehož vznikne takzvaná databáze otisků. Ty se dále využívají pro zpřesnění polohy z RSSI měření [5].

Charakteristika rádiových vln UWB a způsob modulace signálu umožňuje efektivní měření vzdálenosti mezi zařízeními využitím jiných spolehlivějších metod než je RSSI a to například Time of Flight (ToF), Time of Arrival (ToA) a AoA [2].

2.2 Algoritmy určování pozice

2.2.1 Triangulace

Jednou z nejstarších technik pro hledání neznámého bodu je triangulace. Uplatnění našla v geodézii, metrologii, astrometrii a v mnoha dalších odvětvích. Využívá dvou referenčních bodů, které společně tvoří linii, od které se pak dále měří úhel k neznámému bodu.

Problémem této techniky v rádiových systémech je nezbytnost funkce AoA, která dokáže změřit úhel příchodu signálu na anténu rádia. Takový systém je komplikované navrhnout a z tohoto důvodu se triangulace v rádiových systém moc nepoužívá [1].

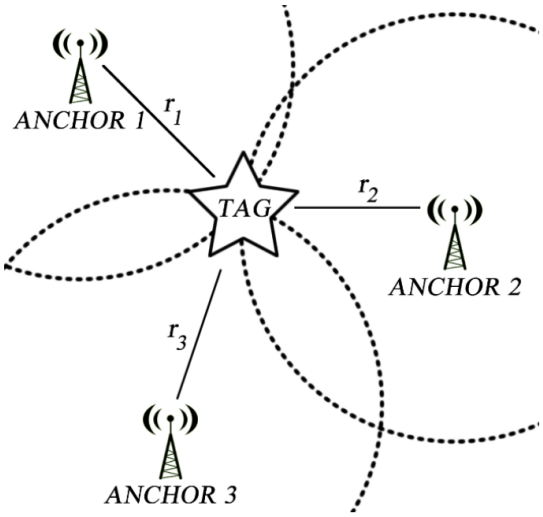
Metoda funguje následovně. Ve 2D prostoru musíme znát polohu dvou bodů. Díky tomu známe přesnou vzdálenost mezi nimi. Z pozic známých bodů se změří úhel mezi neznámým bodem a druhým známým bodem. Z těchto informací lze vypočítat výšku trojúhelníku z neznámého bodu, tedy vzdálenost neznámého bodu ke známé straně. Poté se dopočítá délka strany k neznámému bodu svírající naměřený úhel, čímž jsme schopni dopočítat jeho přesnou polohu [12].

2.2.2 Trilaterace

Trilaterace je hledání souřadnic neznámého bodu na základě znalosti 3 referenčních bodů a vzdálenosti referenčních bodů od neznámého. Tato metoda je použita například v systému GPS. Výsledná pozice leží na průniku kružnic s poloměry o velikosti dané vzdálenosti od referenčního bodu. V ideálním případě se vytvoří přesný průnik všech tří kružnic [3]. Pozici neznámého bodu lze vypočítat pomocí tři rovnic o dvou neznámých x a y viz soustava rovnic 2.2.

$$\begin{aligned}(x - x_1) + (y - y_1) &= r_1 \\ (x - x_2) + (y - y_2) &= r_2 \\ (x - x_3) + (y - y_3) &= r_3\end{aligned} \tag{2.2}$$

Pro výpočet pozice pomocí trilaterace jsou potřebné 3 rádiusy, které mohou být pro lokalizační systém získány například z vlastností rádiové komunikace. Rádiusy, tedy vzdálenosti od kotev k neznámému bodu, jsou označené jako r_1, r_2 a r_3. Pozice kotev musí být předem známé a v soustavě rovnic jsou označené proměnnými x a y s indexy 1 až 3. Proměnné x a y tvoří souřadnice neznámého bodu. Trilaterace je znázorněna na obrázku 2.3.



Obrázek 2.3: Znázornění trilaterace pro získání pozice tagu. Převzato z [3].

2.2.3 Multilaterace

Pojem multilaterace se objevuje nejčastěji ve spojení z technikou TDOA. Multilaterace je jednou z nejvíce využívaných metod pro lokalizaci v omezených oblastech. Vyžaduje informaci o rozdílu času přijatého signálu mezi dvěma kotvami. Tento rozdíl času do jisté míry definuje vzdálenost od těchto dvou kotev [12].

Podobně jako u trilaterace i zde pro vypočítání polohy ve 2D stačí 3 referenční body. Rozdíl v multilateraci oproti trilateraci je v geometrické reprezentaci naměřené vzdálenosti pro nalezení průniku a současně i polohy bodu. Místo kruhu či koule ve 3D prostoru se zde počítají hyperboly mezi referenčními body, popř. hyperboloid ve 3D [12]. Tyto geometrické útvary definují, kde všude se zařízení vysílající signál mohlo nacházet. Pro zjednodušení: pokud lokalizujeme ve 2D prostoru, pak hyperbola mezi kotvami prochází vždy tak, že nejkratší vzdálenost zařízení k oběma kotvám se nachází přímo mezi kotvami. S posunem zařízení dále v prostoru se vzdálenost od obou kotev zvětšuje.

2.2.4 Další algoritmy určování pozice

Metoda nejmenších čtverců je empirická metoda lineární regrese pro řešení nelineárních problémů. V úloze určování pozic ji lze využít máme-li k dispozici více referenčních bodů, které vytvářejí různé nelineární pozice neznámého bodu a my chceme najít tu správnou optimální pozici [3].

Mezi další metody patří určování pozice pomocí centroidu. Tato metoda pro určení pozice neznámého bodu využívá hledání centroidu, středu polygonu, který je vytvořen z průniků rádiusů s poloměry neměřených vzdáleností tagu od referenčního bodu, například pomocí RSSI. Hledání centroidu může být také váhované, což může nedostatky RSSI metody snížit a zvýšit přesnost vypočtené pozice [3].

2.3 Techniky lokalizace

2.3.1 TDOA

Time Difference of Arrival (TDOA) je technika lokalizace, která využívá měření ToA signálu na fixních referenčních zařízeních dále označovaných jako kotvy. Princip TDOA spočívá v měření časových rozdílů příchodu signálu na kotvy. Na základě těchto časových rozdílů a znalostí přesné pozice kotev se počítá místo, odkud signál přišel. Nutnou podmínkou je, aby všechny kotvy použité pro lokalizaci, měly přesně synchronizované hodiny. Pro správnou a přesnou lokalizaci musí být časová známka příchodu signálu na kotvu velice přesná a minimálně ovlivněná odlišnostmi hodinových krystalů zúčastněných kotev [10].

Signál, který kotvy přijímají, je zaslán z pohybujícího se sledovaného zařízení, které budeme dále označovat jako tagy. Tag zasílá v případě TDOA lokalizace broadcastem jednu zprávu všem kotvám v okolí. Ty kotvy, které korektně přijmou zprávu s dostatečnou sílou signálu, mohou být využity pro lokalizaci daného tagu. Obsah zprávy, kterou zasílá tag, není nijak důležitý. Obsahuje zejména identifikátor tagu, například v podobě MAC[5] adresy a dále pak hodnoty senzorů vestavěných v tagu [10].

Výhodnou systému, který používá TDOA lokalizaci, jsou nízké nároky na hardware tagu. Ten pouze periodicky vyšle jednu zprávu (blink), popřípadě naměří data ze senzorů a posléze může přejít do spánkového režimu, během kterého zařízení odebírá minimální

[5] fyzická adresa síťového zařízení

elektrický proud pro zajištění vetší výdrže na baterii. Takové zařízení může být v provozu na jednu knoflíkovou baterii více než rok. Ostatní zařízení sice mají na starost náročnější výpočetní operace, avšak ty už mohou být napájeny z elektrické sítě. Opačný případ nastává v alternativní implementaci TDOA nazvané Reverse TDOA. V tomto případě zařízení v roli tag přijímá signál od fixních referenčních kotev a samotný tag na základě nich počítá pozici, podobně jako v systémů GPS. Vypočtena pozice pak může být dále poslaná na server [5].

Tagy pro účely lokalizace nemusí také uchovávat informaci ohledně kotev, kterým posílá zprávy, jelikož tyto blinky nejsou nikomu konkrétně adresovány, jako v případě systémů založeném na Two-way Ranging (TWR). Pokud se tedy v systému něco změní, například počet kotev anebo jiná nastavení serveru zajištující RTLS, pak tagy ve většině případů nevyžadují rekonfiguraci, což může být v instalacích z tisícovkami tagů zásadní [10].

Efektivní komunikace typu 1 blink se rovná 1 pozice, umožňuje v systémů provozovat velké množství tagů předtím, než dojde k nadměrnému využití radiového pásma a k následnému snížení úspěšnosti vypočtené pozice z důvodu rušeni. Jednou z vlastností systému TDOA je proto vysoká škálovatelnost celé lokalizace. Systém je limitován hlavně rádiovou, síťovou a serverovou utilizací. [10].

2.3.2 TWR

Druhá technika lokalizace pomocí TWR spočívá v určení doby letu signálu ToF, který po vynásobením rychlosti signálu neboli rychlostí světla, udává vzdálenost mezi kotvou a tagem. Tato technika je velice přesná, pro určení ToF vyžaduje oboustrannou komunikaci mezi oběma účastníky [3].

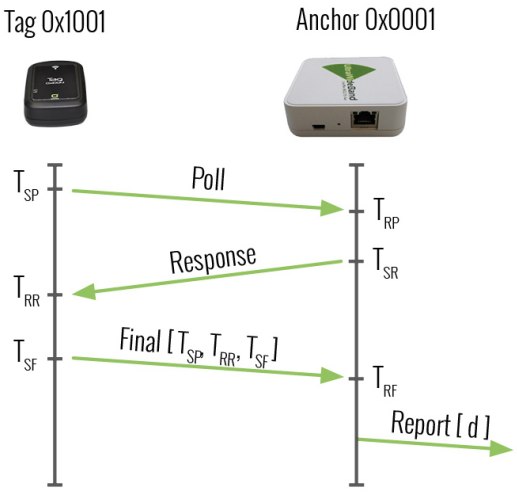
Jeden z účastníků musí inicializovat celou komunikaci. Příklad takové komunikace lze vidět na obrázku 2.4. V tomto případě tag zahajuje měření, avšak z pohledu lokalizace toto není důležité. Tag zasílá první zprávu a zaznamená si její čas. Kotva za dobu ToF signálu přijme zprávu, z čehož však tuto hodnotu nemůže zjistit, protože nezná čas vyslání signálu. Kotva počká předem definovaný čas a vyšle odpověď. Tag odpověď přijme, zaznamená si čas příchodu a přepošle další zprávou všechny potřebné informace kotvě [3]. Ta následně vypočítá ToF pomocí vzorce 2.3.

$$TOF = \frac{t_2 - t_1 - t_{reply}}{2} \tag{2.3}$$

Pro výpočet vzdálenosti mezi tagem a kotvou se hodnota ToF vynásobí rychlostí rádiových vln, která je shodná s rychlostí světla ve vakuu [11].

Z hlediska lokalizace je technika TWR měření přesná a jednoduchá na implementaci. Další výhodou této techniky je, že nevyžaduje synchronizaci času mezi kotvami, což přináší mnoho potencionálních problémů na různých úrovních RTLS systémů. Avšak zásadní limitací systému založeného na TWR je jeho škálovatelnost. Problém spočívá v nadměrné komunikaci mezi tagem a kotvou. Aby bylo možné detekovat vzdálenost mezi oběma účastníky, musí projít 3 unicastové zprávy. V lokalizační buňce se čtyřmi kotvami se tento proces bohužel nedá nijak rozumně paralelizovat a zprávy zasílat broadcastem všem kotvám, takže celý proces se musí zopakovat s každou z nich [11].

Takový lokalizační systém vyžaduje nejrůznější zdroje. Přibývající počet lokalizovaných tagů způsobuje větší využití rádiového pásma. Zvyšující se počet kotev zabírá delší časový interval pro určování pozic jednoho tagu. Delší čas strávený pro lokalizaci zvyšuje využití hardwarových zdrojů tagů a kotev, což v případě tagu snižuje životnost baterie. Delší



Obrázek 2.4: TWR měření vzdálenosti mezi tagem a kotvou. Změření ToF vyžaduje vzájemnou výměnu několika zpráv. Převzato z [11].

čas potřebný pro lokalizaci také limituje množství možných pozic jednoho tagu. To vše dohromady snižuje škálovatelnosti celého RTLS systému [11].

Kromě těchto vlastností, které limitují spíše lokalizaci typu tag-kotva, má systém také mnoho důležitých předností. První z nich je jednoduchá a nenákladná instalace lokalizačního systému. Další výhodou je, že systém nevyžaduje časovou synchronizaci a také možnost jednoduché analýzy výpočtu pozice v případě odstraňování problémů [11].

Kapitola 3

Lokalizační platforma Sewio RTLS

Tato kapitola představí real-time lokalizační platformu firmy Sewio, pro kterou je vyvíjené řešení této práce určeno. Jsou zde popsány všechny součásti daného systému a rovněž lokalizační režimy, které daná platforma umožňuje. Tyto režimy jsou z hlediska práce klíčové, protože do jisté míry určují způsob umisťování referenčních jednotek pozičního systému.

Sewio RTLS je ucelený lokalizační systém využívající UWB bezdrátovou technologii. Tato technologie i její výhody již byly popsány v části 2.1. Systém firmy Sewio vyžaduje 3 druhy zařízení. První je server, na kterém běží lokalizační software RTLS Studio. Další částí je pak anchor neboli kotva, která reprezentuje pevný a známý referenční bod pro lokalizaci. Posledním typem zařízení je lokalizovaný tag. Platforma Sewio využívá techniku určování pozice pomocí TDOA a tudíž vyžaduje přesnou synchronizaci času, aby všechny dostupné kotvy reportovaly korektní čas přijetí signálu z tagu.

3.1 Hardwarové komponenty systému

Jak už bylo zmíněno v úvodu kapitoly, systém Sewio využívá pro lokalizaci dva typy zařízení: kotvu a tag.

3.1.1 Kotva

V Sewio RTLS je kotva pevné zařízení, které přijímá UWB pakety čili blinky, vyslané z tagu. Aktuální produktová řada firmy Sewio obsahuje dva typy kotev[1]:

- Anchor Vista Omni - tato kotva je vybavena všesměrovou 360° anténou. Kotva může v EU regionu operovat na UWB kanále 2 (3744 - 4243.2 MHz) a 5 (6240 – 6739,2 MHz). Dále má industriální kryt a odolnost vůči vodě a prachu IP65.
- Anchor Vista DirectFive - tato kotva má na rozdíl oproti kotvy Omni směrovou anténu s vysokým ziskem. Pro provoz této kotvy je určen pouze kanál 5.

Obě kotvy již dále sdílejí společný HW a jsou založené na UWB čipu od firmy Decawave, konkrétně DW1000. Tento čip podle dokumentace[2] umožňuje lokalizaci v režimu TWR a TDOA s přesností na 10 cm. DW1000 je napájen napětím v rozsahu 2,8 V až 3,6 V. Čip podporuje také energetický nenáročné spánkové režimy, ve kterých odebírá pouze 1 uA

[1] https://docs.sewio.net/docs/anchor-overview-30147655.html
[2] https://www.decawave.com/wp-content/uploads/2020/04/DW1000_Datasheet.pdf

Číslo kanálu	Skupina	Frekvence (MHz)	Šířka pásma (MHz)
2	Low band	3993,6	499,2
5	High band	6489,6	499,2

Tabulka 3.1: UWB kanály využívané v RTLS Sewio. Převzato z [4]

anebo dokonce 50 nA v případě režimu Deep Sleep. Kotvy Omni i DirectFive jsou napájeny pomocí PoE 802.3af nebo pasivním 48V PoE. Pro síťovou komunikaci je možné použít Fast Ethernet port anebo Wi-Fi. Poslední vlastností, kterou kotvy sdílejí, je integrace barometru.

Kotvy musí mít v rámci lokalizovaného plánu přesně známou relativní pozici, aby bylo možné pomocí multilaterace a TDOA vypočítat pozici vyslaného blinku. To, jakým způsobem musí být kotvy rozložené, závisí zejména na prostředí, typu lokalizace a použitých kotvách. Různé typy lokalizace, které Sewio RTLS podporují, bude popsáno dále v textu.

Jak už bylo zmíněno, hlavním rozdílem mezi Anchor Vista Omni a DirectFive je použitá anténa. Všesměrová anténa z kotvy Omni má velmi rovnoměrnou vyzařovací charakteristiku po celých 360° okolo kotvy. To dává integrátorovi možnost umístit kotvu do různě členitých uzavřených prostředí anebo propojit lokalizaci jiné ucelené lokalizační buňky. To je velice důležité, protože TDOA lokalizace pomocí UWB vyžaduje LOS nejen mezi kotvami, ale i mezi kotvou a tagem. Všesměrovost antény na kotvě Omni je pouze v horizontálním pohledu na kotvu. Směrová anténa na kotvě DirectFive sice ztrácí výhodu provozu UWB kolem celé kotvy, ale zato ve správném směru dokáže vyzařovat intenzivněji. Pro příklad: 4 kotvy Omni s maximální doporučenou vzdáleností v ideálním prostředí bez překážek pokryjí 225 m^2, zato varianta se směrovou anténou pokryje spolehlivě 625 m^2. Proto kotva DirectFive může ve specifických prostředích výrazně snížit počet kotev a také náklady na pořízení a instalaci.

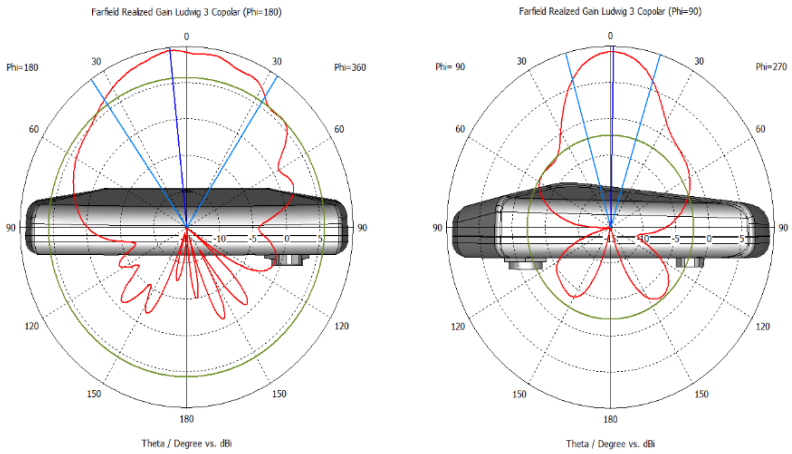
Vyzařovací charakteristika na obrázku 3.1 z horního pohledu ukazuje hlavní lalok vyzařování a několik malých zpětných laloků. Tento hlavní lalok má největší vyzařování v rozsahu cca 65°. Vyzařování je nejsilnější zhruba ve středu kotvy a od něj se drží nad přijatelnou hranicí $+-45°$, což dohromady znamená 90° kvalitního pokrytí směrovou anténou kotvy DirectFive. Vertikální vyzařování z profilu ukazuje cca 30° rozpětí nejsilnějšího signálu, dále už pak intenzita výrazně klesá. Tento fakt je z hlediska instalace kotev také velmi důležitý, jelikož je potřeba myslet i na správné vertikální sklopení kotvy. Za zmínku také stojí i zpětné laloky kotvy, které mohou ovlivnit jiné kotvy v blízkosti [8].

3.1.2 Tag

Tag je malé energeticky nenáročné zařízení, které periodicky vysílá zprávu blink, na základě které se vypočte jeho pozice. Firma Sewio má aktuálně v nabídce Tag Leonardo[3], který je nabízen ve čtyřech variantách, rozdělených podle případu užití. Mezi tyto 4 varianty patří:

- Leonrado Personal - tato varianta je určená pro lokalizování osob a je charakteristická použitou Li-on baterií, bezdrátovým QI nabíjením. Kromě akcelerometru, který slouží k probuzení tagu ze spánkového režimu, je tag Leonardo Personal vybaven také barometrem.

[3] https://docs.sewio.net/docs/tags-overview-3244803.html



Obrázek 3.1: Vyzařovací charakteristika kotvy DirectFive z horního a bočního pohledu. Převzato z [8].

- Leonardo IMU - varianta iMU vychází z tagu Personal a rozšiřuje jej o dodatečné Inertial measurement unit (IMU) senzory, tedy konkrétně gyroskop a magnetometr. Díky těmto novým senzorům a akcelerometru z barometrem můžeme zjistit nejen výšku tagu, ale i jeho natočení. Kombinace dat z těchto senzorů vytváří takzvané quaterniony, které slouží k definici rotací v 3D prostoru.
- Leonardo Asset - je určen pro jakýkoliv asset, tedy materiál, který je potřeba sledovat. Často bývá použit ve skladech a výrobních prostorech pro sledování takzvaného material flow. Materiál se většinou nepohybuje vysokou rychlostí a tudíž nepotřebuje vysoký refresh interval, tedy interval opakování vyslání blinku. Důležitá je zejména výdrž baterie. Asset je proto napájen knoflíkovou baterií CR2477 1000 mAh. Při refresh intervalu 1s tag dokáže fungovat na jednu baterií zhruba 2,8 roků. Z hlediska senzorů Asset nabízí pouze akcelerometr pro probouzení tagu ze spánku pohybem.
- Leonardo Vehicle - varianta Vehicle je určená pro lokalizování velkých pohyblivých strojů, zejména vysokozdvižných vozíků. Ty jsou napájené ze svého zdroje, který musí být použit také i pro napájení tagu, jelikož Leonardo Vehicle je napájen externě DC v rozmezí 7 - 35V. Tag je osazen také kompletním IMU senzorem stejně jako Leonardo IMU.

Kromě všech těchto vlastností tagy Leonardo disponují BLE pro aktualizaci verzí firmwaru, NFC pro rekonfiguraci tagu. Tagy Leonardo jsou zobrazeny na obrázku 3.2.

3.2 Lokalizační server RTLS Studio

RTLS Studio je serverová část Sewio RTLS systému. Může být provozovaná na Ubuntu 18.04 anebo v linuxových Docker kontejnerech.

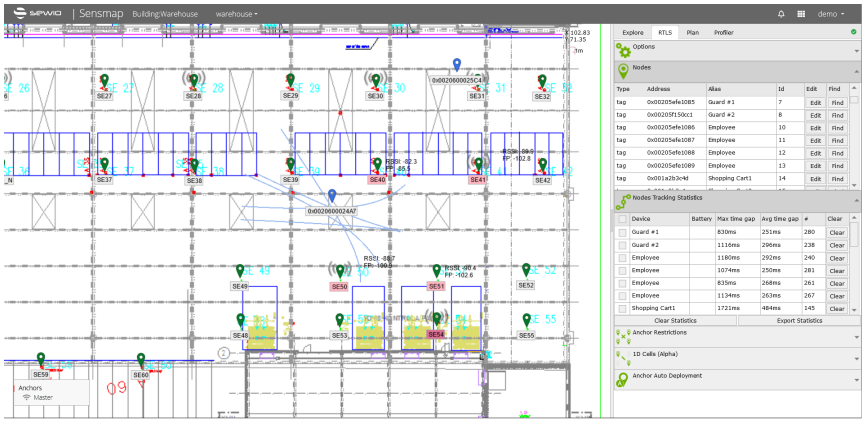
RTLS Studio nabízí několik dílčích aplikací. RTLS Manager slouží pro správu kotev i tagů. Je zde integrovaná funkce inicializace, která má za cíl ohodnotit rádiové spojení všech kotev a vybrat nejlepší master kotvu, která bude zajišťovat synchronizaci času mezi TDOA kotvami pro danou lokalizační buňku. Další funkcí RTLS Manager je Sync stability,

Obrázek 3.2: Ukázka rodiny tagů Leonardo. Varianta iMU využívá stejného pouzdra jako Personal. Převzato z [9].

která pro každou master kotvu ukazuje aktuální druhou diferenci (2nd Difference), First path a RSSI. Hodnota druhé diference udává stability synchronizace kotev za použití vybraného mastera. Tento parametr agreguje data ze čtyř po sobě jdoucích synchronizačních paketů. Výsledné číslo je interpretováno tak, že pokud je menší než 2, pak synchronizace je dostatečně kvalitní pro lokalizaci. Kromě veškerého nastavení kotev je zde seznam registrovaných tagů, ve kterém lze vidět nastavení každého z nich i status posledního příchozího blinku. V této části je možné překonfigurovat tag pomocí UWB. Pro rekonfiguraci se vybere jedna kotva, potřebné nastavení a MAC adresu tagu. Po zapnutí rekonfigurace vybraná kotva místo příjmu blinku posílá konfiguraci a pokud se tag přepne do RX modu, tuto konfiguraci přijme. Konfigurace může být také poslaná broadcastem pro všechny tagy a tím pádem pak konfiguraci aplikuje jakýkoliv tag, který zprávu od kotvy přijme. RTLS Manager dále ukazuje aktuální statistiky kotev a interní stav synchronizace. Nakonec RTLS Manager umožňuje měnit nastavení RTLS Serveru, který zajišťuje výpočet pozic.

Druhou aplikací je Sensmap, zobrazený na obrázku 3.3. Tato část RTLS Studia slouží pro vizualizaci pozic tagu v půdorysu lokalizovaného prostředí. Sensmap umožňuje mít definovaných více budov a plánů a pro každý z nich mít určené určité kotvy. Při vytváření a nahrání plánu určí uživatel úsek a k němu příslušnou velikost v reálném světě. To definuje, kolik pixelů plánu reprezentuje 1 m ve skutečnosti. V aplikaci Sensmap lze definovat Anchor restrictions, což jsou pravidla definující, které kotvy se můžou využívat s konkrétním masterem, čím se oddělí kotvy pro různé lokalizační buňky a zamezí se vyžití nevhodných kotev pro lokalizaci. Další funkcí Sensmap je Profiler, který pro každý blink daného tagu vizualizuje hyperboly multilaterace a ukazuje hodnoty, s jakými kotvy přijaly blink. Další informace z Profileru ukazují, jakým způsobem výpočet pozice prošel všemi úrovněmi výpočtu. Pokud pozici nebylo možné vypočítat, lze z něj zjistit, ve které části výpočet selhal. Sensmap existuje i ve 3D variantě, kde dokáže zobrazovat pohyb tagů ve třech osách a díky quaternionu i jejich orientaci.

Aplikace SAGE slouží pro analýzu a vizualizaci historických dat z databáze RTLS Studia. Pro různé časové úseky může ukázat pohyb tagů, vstupy a výstupy ze zón, heatmapu pohybu a mnoho dalších analytických pohledů. RTLS Monitor ukazuje systémové informace serveru a vytížení serveru v realném čase a umožňuje ovládat jednotlivé služby RTLS Studia. RTLS IO je služba, které propojuje události vzniklé v lokalizaci, například pokud tag vstoupí do zóny definované v Sensmap, sepne port automatizačního zařízení Advantech.



Obrázek 3.3: Aplikace Sensmap, která ukazuje pozice jednotlivých tagů na plánech. Pro jeden vybraný tag dokáže vizualizovat hyperboly multilaterace a další detailní informace o výpočtu pozice.

3.2.1 Anchor Auto-Deployment

Součástí aplikací RTLS Manager a Sensmap je funkce nazvaná Auto-Deployment. Je integrovaná od verze RTLS Studio 2.0.0. Ke své činnosti vyžaduje firmware kotvy od verze 3.0.5, kvůli integrace TWR funkce. Účelem této verze Auto-Deploymentu bylo zpřístupnit Sewio RTLS novým uživatelům pomocí RTLS kitů, které obsahují veškeré příslušenství potřebné k provozu systému. Samotný kit obsahuje 5 kotev, což souvisí i s hlavními limitacemi aktuální funkce Auto-Deployment.

Ta je schopná spolehlivě rozmístit do plánu maximálně 5 kotev, které jsou výhradně ve stejné výšce a rozmístěné ve čtverci. Dále funkce vyžaduje, aby všechny kotvy měly navzájem LOS. Význam této funkce je čistě pro demonstrativní a výukové účely, pro které tyto limitace nejsou výrazný problém. Výpočetní část Auto-Deploymentu je integrovaná do procesu Anchor Initialization. Ve webovém rozhraní inicializace je možné tuto funkce povolit anebo zakázat. Pokud je v systému méně než 6 kotev, pak auto-deployment je předen povolený, jinak vždy zakázaný s možností manuálního povolení.

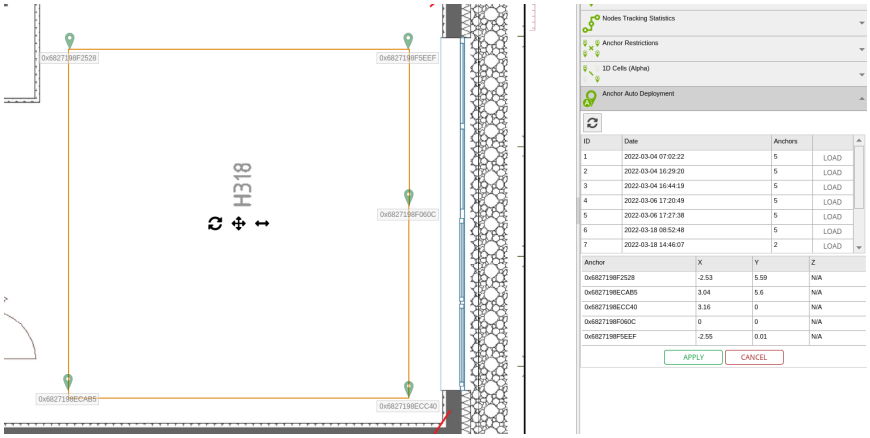
Auto-Deployment umožňuje uživateli jednoduše a přívětivě nasadit kotvy do plánu v aplikaci Sensmap. V sekci Auto-Deployment aplikace Sensmap se uživateli zobrazí seznam inicializací, se kterými jsou spjata i jednotlivá TWR měření. Když uživatel načte některou z inicializací kotev, pak webové rozhraní zobrazí box s definovanými kotvami a pomocí geometrických transformací (posun, rotace, převrácení) uživatel umístí kotvy na požadované místo viz obrázek 3.4.

3.3 Lokalizace v platformě Sewio

Lokalizace v RTLS Sewio je zaměřená zejména na plné určování pozice ve 2D s možností 3D pomocí dat z barometru. Avšak oproti této úrovně lokalizace systém nabízí další režimy využívající bezdrátovou technologií UWB.

3.3.1 2D

Lokalizace 2D, také zvána True Location, je druh určování pozice, který poskytuje X a Y souřadnice tagu v ideálním prostředí s přesností do 30 cm. Lokalizace využívá metody



Obrázek 3.4: Sensmap GUI pro Auto-Deployment.

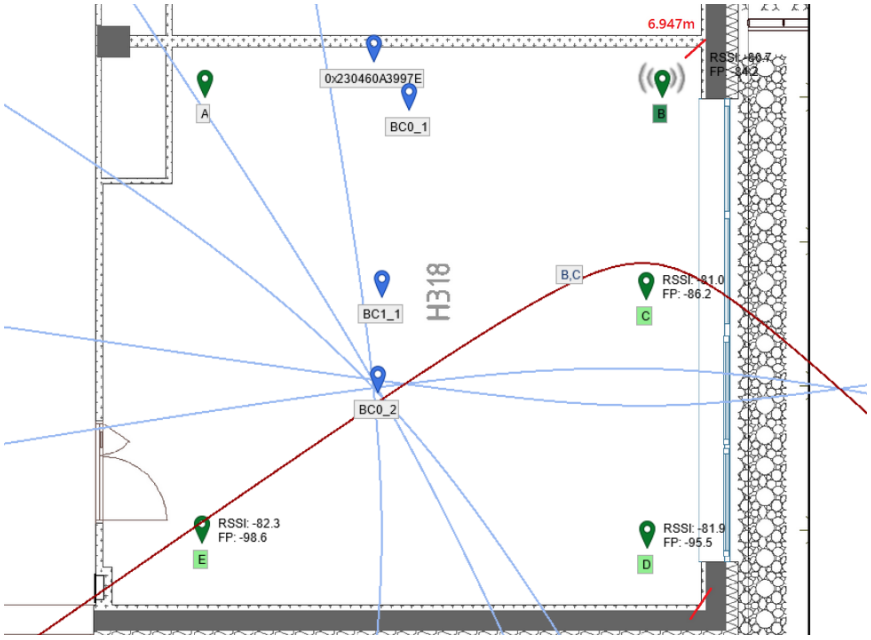
TDOA a na každé kotvě měří čas příchodu signálu, z čehož se vypočítá rozdíl příchodu mezi kotvami. V tomto systému ve výchozím nastavení je potřeba přijmout blink z tagu alespoň na čtyřech kotvách pro zachování dostatečné přesnosti.

Čas příchodu signálu na kotvy, které jsou blíže tagu, je ze své podstaty kratší. Aby tento fakt platil, kotva se snaží detekovat přímý signál a ne odražený od okolního prostředí, čemuž pomáhají rádiové vlastnosti technologie UWB. Z toho důvodu nutnou podmínkou pro přesnou lokalizaci pomocí UWB a TDOA je mít LOS z tagu na kotvy.

Multilaterace je v tomto systému vybraná metoda výpočtu pozice použitím informace TDOA zaslaného blinku na okolní kotvy. Jak už bylo v kapitole 2.2.3 uvedeno, výsledná pozice je průnik geometrických tvarů, v tomto případě hyperbol, které jsou vytvořeny z 2 referenčních bodů. Hyperboly všech kombinací referenčních bodů by se v ideálním případě měly protnout. Toho však v reálném světě nedosáhneme, takže výslednou pozici musí určit vybraný optimalizační algoritmus. Na obrázku 3.5 lze vidět vizualizaci multilaterace v systému Sewio, který dokáže tyto hyperboly pro jeden vybraný tag zobrazovat. Zeleně vyznačené kotvy jsou použity pro výpočet této konkretní pozice. Tmavě zelená kotva je master kotva zajišťující časovou synchronizaci s ostatními kotvami. Většina hyperbol je velmi málo zkosená. To je dáno skutečností, že tag je ve většině kombinací kotev zhruba ve středu mezi nimi. Odlišná hyperbola je pouze v případě kotev B a C. Tag se v tomto případě nachází mnohem blíž kotvě C než ke kotvě B, což způsobí, že hyperbola prochází ve větší blízkosti kotvy C a je kolem ní výrazně stočená.

TDOA lokalizace vyžaduje časovou synchronizaci mezi kotvami. Rozdíl jedné nanosekundy v časování kotev může vytvořit 30 cm v přesnosti lokalizace. V systému Sewio je hned několik způsobů synchronizace. Každá se liší svou škálovatelnosti anebo například použitím v sítích s vysokou latencí, například na kotvách připojených přes Wi-Fi. Z hlediska synchronizace rozlišujeme dva typy kotev, master kotvy a sousední kotvy. Master kotva je referenční kotva z hlediska synchronizace a v pravidelném intervalu 1s rozesílá UWB sync zprávy ostatním kotvám. V nejjednodušším případě synchronizace probíhá následovně. Server anebo kotva pošle přes síť broadcast[4] všem kotvám s tím, že za určitou dobu mají očekávat UWB synchronizační zprávu. V předem definovaném intervalu se všechny kotvy přepnou do synchronizačního UWB profilu a master kotvy začnou vysílat sync zprávy ve svém časovém intervalu. Sousední kotvy tyto sync zprávy následně přijímají.

[4] vysílání všem příjemcům



Obrázek 3.5: Vizualizace výsledku multilaterace tagu BC0_2 za použití čtyř kotev (zeleně vyznačené). Zvýrazněná hyperbola byla vypočtena z hodnot kotev B a C.

3.3.2 1D

Lokalizace typu 1D slouží pro určování pozice v úzkých koridorech a uličkách, kde poloha tagu leží na předem definované přímce, takže samotný výpočet pozice je ve skutečnosti pouze v jedné dimenzi.

Přímka je matematicky definovaná dvěma body. V tomto případě tato lokalizační oblast je určena dvěma kotvami. Sewio systém podporuje pro 1D lokalizaci pouze kotvy Vista DirectFive, takže lokalizace je těmito směrovými kotvami také i ohraničena, tudíž funguje pouze mezi těmito body. Maximální doporučená vzdálenost mezi kotvami je 30 metrů. Pro delší uličky je zapotřebí tyto lokalizační uličky řetězit přímo za sebou viz obrázek 3.6.

Obrázek 3.6: Řetězení 1D uliček. Převzato z [6].

Z hlediska instalace je nutné zajistit LOS mezi oběma kotvami uličky pro zajištění synchronizace. Dvě paralelní uličky by měly být alespoň 6 metrů od sebe vzdálené v případě zastavěné fyzické překážky mezi uličkami. Na volném prostoru, pro který 1D systém není uzpůsobený, je tato vzdálenost mnohem větší, minimálně okolo 30 metrů.

3.3.3 0D

Poslední úroveň určování polohy v systému Sewio je nazvaná 0D - Presence detection, tedy detekce přítomnosti. V této úrovni lokalizace je každá pozice tagu počítána jako pozice kotvy, která zachytila blink tagu nejsilněji, tedy kotvy s největším RSSI příchozího blinku.

Jedná se o nejjednodušší typ lokalizace a nevyžaduje žádné vzájemné zaměření kotev. Pozice kotvy není důležitá pro výpočet pozice. Poloha tagu je rozhodnutá na základě RSSI jedné kotvy, což může být velmi nestabilní, proto RTLS Server nabízí možnost nastavení limitu RSSI pro změnu pozice, rozdíl v RSSI potřebný pro změnu a také čas mezi změnami pozice tagu.

Kapitola 4

Využití metody TWR pro měření vzdáleností

Tato kapitola pojednává o měření TWR dat napříč kotvami reálné instalace. Metoda lokalizace pomocí TWR byla k tomuto účelu předurčena svými charakteristickými vlastnostmi a aktuálními funkcemi Sewio RTLS. V této úloze neměříme vzdálenost mezi tagem a kotvou, nýbrž mezi kotvami navzájem. Všechny kotvy jsou externě napájeny a komunikují napříč pomocí Ethernetu anebo Wi-Fi. Z toho důvodu je TWR pro tuto úlohu ideální, jelikož je přesná, jednoduchá a v případě kotev je energetická náročnost procesu nepodstatná, stejně jako i časová složitost. Proces rozmístění kotev na plán se provádí typicky jednorázově pouze při instalaci anebo modifikaci RTLS instalace.

Pro TWR měření kotvami systému Sewio RTLS byly vybrány dvě instalace. První z nich je Intemac, který reprezentuje výrobní haly, dílny a kanceláře pro 2D lokalizaci. Důležitým poznatkem má být znalost chování TWR v takových prostředích a také informace o možnosti filtrace měření mezi kotvami, které nemají mezi sebou přímou viditelnost. Druhým měřícím prostředím je logistický sklad. Toto místo je typickým příkladem 1D lokalizace a je přesně zmapované, takže výsledky z tohoto místa mohou být relevantní pro další vývoj. Cílem měření bylo zejména zjistit přesnost TWR na DirectFive kotvách a dále charakteristiku TWR mezi kotvami sousedních uliček.

4.1 Výsledky TWR z 2D instalace

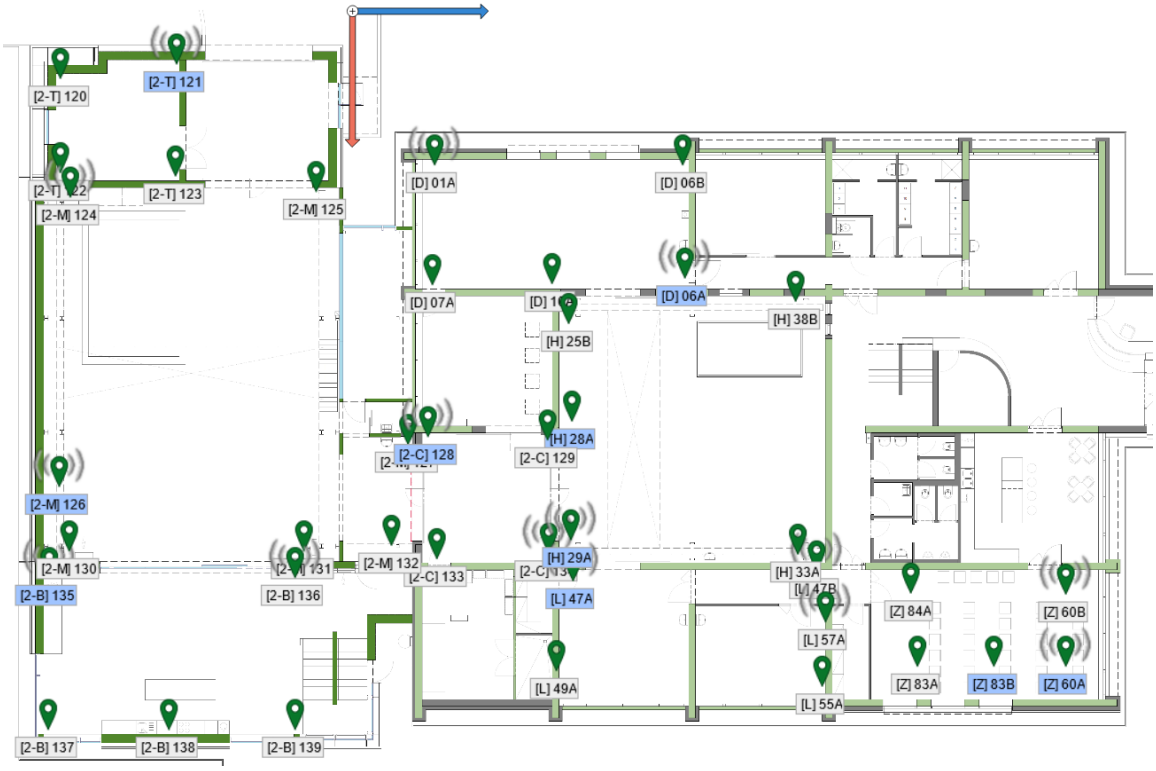
Motivací pro toto měření bylo zejména získání TWR dat z inicializace RTLS systému pro další analýzu a definici možných problémů spjatých s vykonáváním algoritmu pro automatické určování pozic kotev. Cílem prvotní analýzy dat je ze získaných dat vydolovat informaci o tom, která měření neměla mezi kotvami ideální prostředí a LOS. Absence LOS totiž přidává k měření chybu. Jelikož se v TWR lokalizaci měří ToF a rychlost šíření signálu v pevných překážkách je jiná než ve vakuu, proto tady obvyklý vzorec pro výpočet vzdálenosti není přesný. Detekce takto nepřesných měření může vést k rozhodnutí, například zda některá měření budou filtrována anebo určitým způsobem váhována, aby se zvýšila přesnost celkového výpočtu.

4.1.1 Prostředí

Pro měření byla firmou poskytnuta testovací instalace Intemac[1] v Kuřimi. Jedná se o výzkumné a inovační centrum a obsahuje přehled nejrůznějších výrobních technologií. Lokalizační platformou Sewio je pokryta velká část přízemí budovy. Lokalizace funguje v nejrůznějších prostředích, například v konferenční místnosti, chodby, velké haly s CNC stroji, ale i otevřené prostory s lokalizační buňkou oddělenou skleněnou stěnou.

Toto prostředí je z hlediska vzájemné viditelnosti kotev náročné a jednotlivé lokalizační buňky jsou často fyzicky oddělené. Z pohledu automatického zaměřování kotev budou největším problémem právě tyto přechody, které můžou potenciálně zanášet chybu v případě určování pozic kotev celé instalace.

Přestože je tato instalace z pohledu zaměřování kotev náročná, měření z ní může lépe natrénovat případný algoritmus. Prostředí bude také použito pro výsledné testování a vyhodnocení vyvinutého řešení. Půdorys lokalizovaného patra se zaznamenanými kotvami je ukázán na obrázku 4.1.



Obrázek 4.1: Půdorys lokalizovaných oblastí v instalaci Intemac.

4.1.2 Příprava měření

První fáze měření závisí na již implementované funkcionalitě v firmwaru kotvy a serverové aplikace RTLS Studio. V systému kotvy je potřebná funkcionalita dostupná od verze FW 3.005 a vyšší. TWR funkce kotvy obsahuje následující části:

- Inicializace měření pomocí REST rozhraní kotvy.

[1] https://www.intemac.cz/

- Možnost vytvořit požadavek na TWR měření pro několik oponentů, omezeno na 19 adres.
- Možnost nastavit pevný časový limit TWR procesu na jednotky milisekund.
- Konfigurovatelný počet iterací TWR měření.

TWR měření se na kotvě spouští pomocí GET požadavku na HTTP server kotvy. Příkladem URL adresy s potřebnými parametry je: `http://192.168.225.201/twr.cgi?addr=5410ecf4bdac&rep=200&rdly=1000&wait=0/`. Parametr *addr* očekává MAC adresy kotev, se kterými se TWR provádí. Jednotlivé adresy jsou oddělené čárkou. Parametr *rep* definuje počet opakování měření. V tomto případě se používá hodnota 200. Hodnota s názvem *rdly* reprezentuje reply delay a určuje časový rozestup UWB rámců Poll a Response v TWR měření. Parametr *wait* nastavuje dobu čekání mezi opakováními TWR měření pro jednu kotvu. Hodnota *rdly* a *wait* je celočíselná hodnota času v mikrosekundách.

V aplikaci RTLS Studio není nutné pro účely získání TWR dat z instalace provádět výrazné úpravy. Proces inicializace kotev RTLS Manageru, který automaticky určuje master kotvy pro TDOA synchronizaci, spouští také TWR na všech inicializovaných kotvách. Aby se redukovala výrazná časová složitost procesu TWR, která by během provádění měření se všemi kotvami znamenala kvadratickou složitost $\mathcal{O}(n^2)$, se TWR spouští pouze pro ty kotvy, které byly schopny přijmout v předchozí fázi inicializace alespoň nějaké synchronizační pakety. V procesu inicializace se provádí TWR s 200 iteracemi pro každou kotvu viz příklad URL adresy výše.

Pro účely měření bylo nutné pouze seskupit data z TWR a vhodně je serializovat do souboru. Pro tyto účely byl vybrán formát JSON, který obsahuje informace o identifikátoru kotvy v roli iniciátora měření, MAC adrese druhé kotvy a nakonec výsledku a statistikám z provedených 200 iterací TWR. Mezi tyto detailní výsledky patří průměrná vzdálenost mezi kotvami, střední hodnota vzdáleností, maximum, minimum, směrodatná odchylka a procentuální a číselné vyjádření přijatých TWR měření viz výpis 4.1.

```
{
    "initiator": "d880396254df",
    "opponent": "d88039627705",
    "distance": {
        "median": "9.87",
        "mean": "9.88",
        "min": "9.82",
        "max": "9.94",
        "std": "0.02",
        "percentage": "99.50",
        "recv": 199,
        "total": 200
    }
}
```

Výpis 4.1: Příklad JSON objektu s daty z TWR

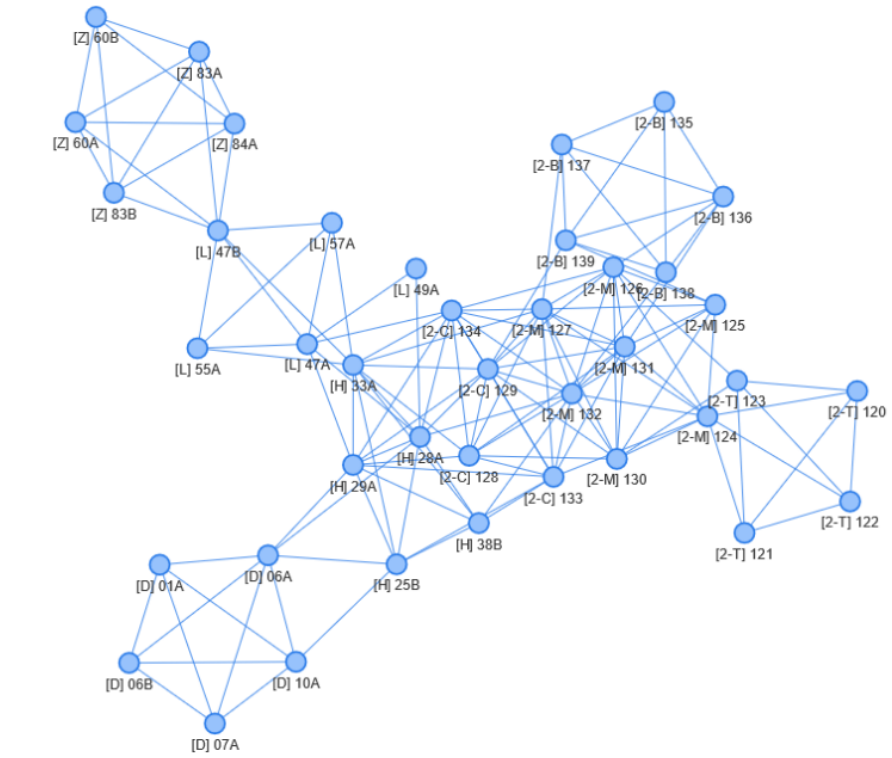
4.1.3 Analýza výsledků

Inicializace se záznamem TWR dat byla prováděna v instalaci Intemac pro všech dostupných 40 kotev najednou. Celková doba inicializace byla 16 minut, během kterých se provádělo

TWR měření mezi 289 kombinacemi kotev. Z těchto kombinací bylo možné změřit vzdálenost u 268 z nich, tedy 92,7 %. Avšak zbylá měření stále obsahovala spojení kotev, která mají ve skutečnosti velmi špatnou nebo žádnou viditelnost.

K vizualizaci výsledků z měření byla využita knihovna *pyvis* pro *Python3*. Vybraný způsob pro získání rychlé představy o naměřených vazbách mezi kotvami, byla vizualizace TWR dat jako síť využívající modul *Network* z uvedené knihovny. Tento modul na základě informací o uzlech a hranách vytváří interaktivní mapu spojení. V tomto případě uzly tvoří kotvy a hrany jsou jednotlivá TWR měření. Před vytvořením mapy bylo možné tato měření různě filtrovat a na základě vygenerované mapy ohodnocovat použitelnost jednotlivých filtrů. Pro zpřehlednění mapy jsou MAC adresy kotev přeloženy na aliasy, které jsou použity v systému RTLS Studio. Každý alias disponuje prefixem, který označuje místnost, kde se kotva nachází. Díky tomu lze jednoduše určit, které kotvy by měly být vzájemně viditelné.

Na obrázku 4.2 lze vidět síť kotev, pro které se podařilo alespoň jednou vypočítat vzájemnou vzdálenost. Síť je v této verzi hustě propojena. Pokud oddělíme jednotlivé lokalizační buňky, tedy místnosti, pak každá z nich tvoří úplný graf. Propojení i mezi místnostmi je značné, zejména pak v oblasti označené prefixy [2-M], [2-C] a [H]. Jedná se o přímo navazující místnosti. Tyto místnosti jsou však odděleny pouze tenkou zvedací bránou určenou pro průjezd vysokozdvižným vozíkem, tedy materiálem, který výrazně neblokuje UWB signál. Propojení místností [L] a [Z] je pouze pomocí kotvy *[L] 47B*, což je nedostatečné pro automatické polohování všech kotev instalace najednou.



Obrázek 4.2: Mapa všech spojení, která byla alespoň jednou měřitelná.

S daty naměřenými v instalaci Intemac byly dále prováděny různé experimenty, které využívaly statistické parametry vzdáleností obsažené v JSON objektu. Z pozorování vý-

sledků vyplynulo, že nejspolehlivější parametry pro filtrování nevhodných vzdáleností, je počet přijatých měření a směrodatná odchylka.

V případě množství přijatých měření, nižší hodnoty znamenají, že rádiové spojení mezi kotvami není spolehlivé. Celý proces inicializace i měření TWR probíhá na samostatném synchronizačním UWB profilu, kterým je ve výchozím nastavení kanál 5 profil RF1. Z toho důvodu nemůže být nižší spolehlivost zdůvodněna rušením kanálu pomocí aktivních tagů. V tomto případě je nižší spolehlivost zdůvodněna významnou překážkou mezi oběma kotvami. V běžném prostředí při zachování doporučených vzdáleností pro jednotlivé typy kotev a za předpokladu zajištěni LOS mezi kotvami by se počet přijatých měření měl blížit 100 %. Přestože v testu byla použita hodnota opakování 200 a TWR měření vyžaduje výměnu 3 zpráv pro zjištěni délky letu signálu, akceptovatelná ztrátovost zpráv se může pohybovat maximálně v jednotkách přijatých měření.

Směrodatná odchylka je druhý užitečný parametr pro filtrování Non-Line of sight (NLOS) spojení. Přestože nižší radiová spolehlivost a tedy ztracená TWR měření znamenají fyzickou překážku spojení, nemusí tomu být i naopak. UWB radio je schopné přijímat i odražený signál, což může znamenat i při NLOS spojení vysokou spolehlivost. Avšak signál se může různě odrážet a tím způsobovat různé vzdálenosti vypočtené pomocí TWR, což také zvyšuje směrodatnou odchylku. Z naměřených dat vyplynulo, že u kotev, které mají mezi sebou ideální LOS, se směrodatná odchylku pohybuje v jednotkách setin. Hodnoty větší 0,1 obvykle znamenají kotvy, které například spojují dvě lokalizační buňky, ale mezi sebou nemají ideální přímou viditelnost.

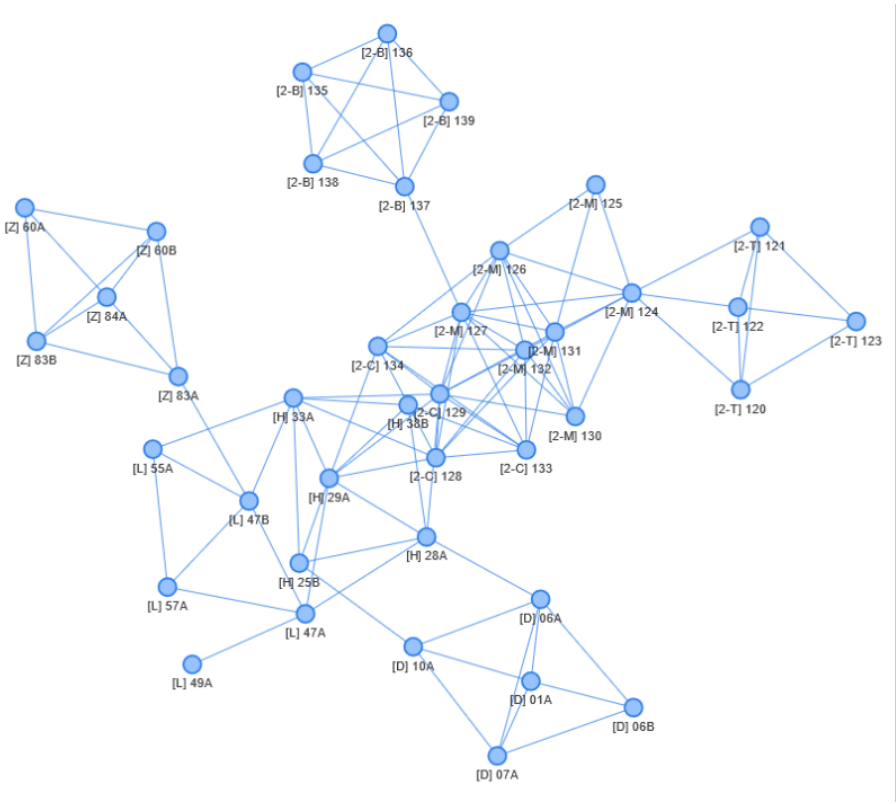
Při filtrování vazeb, které by mohly zanášet chybu měření, bylo v případě počtu přijatých měření z jedné kotvy limitní počet 198, tedy 99 % všech měření. Po zvýšení hranice pro tento filtr na 199 byly už viditelně znát ztráty vazeb uvnitř lokalizačních buněk. Nejlepšího nastavení směrodatné odchylky bylo dosaženo při limitu 0,05. Díky této hodnotě pak bylo možné snížit požadavky počtu přijatých hodnot na 98 %, čímž se zachovaly vazby uvnitř buněk, ale přechody zůstaly ve většině případů omezeny správně. Výsledek s tímto nastavením filtru lze vidět na obrázku 4.3.

Jedna kotva z celého měření vykázala s výše uvedeným nastavením velmi špatné výsledky a to konkrétně kotva *[L] 49A*. Tato kotva má pouze jednu stabilní vazbu a to na kotvu *[L] 47A*. Viditelnost této kotvy na ostatní odpovídá realitě, jelikož tato lokalizační buňka pokrývá dvě místnosti oddělené sádrokartonem, který by však neměl UWB silně blokovat. Je také možné, že kotva mohla být v době měření nevhodně natočená, čímž se zhoršila zisková charakteristika této všesměrové kotvy.

4.1.4 Zhodnocení

Provedené měření splnilo svůj účel. Podařilo se získat surová data TWR měření z reálné a známé instalace. Tato data byla dále zpracována pro vizualizaci reprezentovanou jako síť spojení. Tato síť ukázala, jaká jsou reálná radiová spojení mezi jednotlivými kotvami. Tato informace je velmi důležitá pro další postup.

Díky těmto informacím je možné lépe nastavit cíle a požadavky pro výslednou funkcionalitu. Je jisté, že výsledné řešení nebude schopné fungovat v takto členitém a uzavřeném prostředí plně automaticky. Přechody mezi buňkami budou pravděpodobně znamenat výraznou nepřesnost. Přesnou odpověď na tuto otázku ukážou až další experimenty s navrženým algoritmem. V tuto chvíli je však možné pomocí parametrů, s nimiž se experimentovalo během měření, alespoň detekovat lokalizační oblasti, které mohou tyto nepřesnosti vytvářet.



Obrázek 4.3: Výsledná síť s aplikovanými filtry počtu měření a směrodatné odchylky.

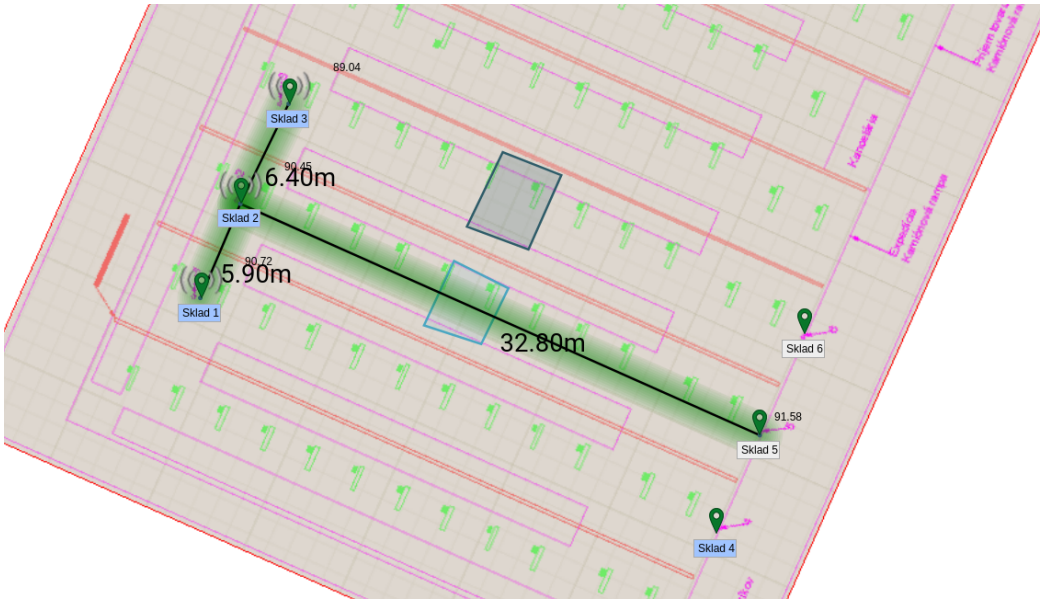
4.2 Výsledky TWR z 1D instalace

Druhé měření bylo provedeno v prostředí logistického skladu. Tato instalace byla vybrána proto, že nabízí vhodný příklad 1D lokalizace a tudíž umožňuje otestovat a naměřit výsledky TWR měření pro daný typ lokalizace. Cílem měření je zjistit charakteristiku TWR výsledku z podobných lokalizačních sestav. Z měření by mělo vzejít, zda použití TWR pro dlouhé vzdálenosti se směrovými kotvami neskrývá nějaký potenciální problém a zda se dá z výsledku TWR nepřímo vyčíst jiné důležité informace, které mohou pomoct následně při tvorbě algoritmu pro určování pozic kotev v 1D instalacích.

4.2.1 Prostředí

Logistický sklad má pokryté 3 uličky 1D lokalizací. Přestože jsou uličky pouze 3, toto prostředí je svou charakteristikou velmi náročné a testuje samotné limity 1D lokalizace. Jako v běžných skladech jsou uličky tvořené pomocí regálů, ve kterých jsou naskládané palety s výrobky. Uličky, které jsou pokryty systémem RTLS, jsou umístěné hned vedle sebe. Sousední kotvy jsou vzdáleny od sebe 6 - 7 m, což je udávané minimum pro správnou funkci RTLS a detekci správné uličky při vstupu do dané oblasti. Také délky uliček jsou limitní a lehce překračují doporučené maximum 30 m. Konkrétně vzdálenosti mezi kotvami jsou v tomto případě 32 - 33 m. Pohled na půdorys skladu s kotvami a vzdálenostmi mezi kotvami lze vidět na obrázku 4.4.

Kotvy jsou také umístěné v relativně vysoké výšce 8,1 m, což je náročnější spíše z hlediska lokalizace. Co se týče vzdáleností a umisťování kotev, je důležitým parametrem rozdíl



Obrázek 4.4: Půdorys logistického skladu se třemi 1D uličkami.

výšek kotev, které jsou v tomto případě také značné, jelikož se jedná o výškový rozdíl 2,7 m. Kotvy na druhé straně uličky jsou totiž ve výšce 5,5 m. Důležitým faktem je také skutečnost, že kotvy jsou montovány v místě, které již nejsou oddělené regály, takže viditelnost sousedních kotev je také velmi dobrá, což může být problém s hlediska automatického rozlišování uliček.

4.2.2 Analýza výsledků

V instalaci popsané v předchozí sekci byla provedena inicializace s TWR měřením pro každou kotvou v dosahu, stejně jako v případě měření v Intemacu. Pro každou kotvu bylo provedeno 200 měření, z nichž se poté agregoval konečný výsledek vzdálenosti dvou kotev. Inicializace byla spuštěna pro všechny kotvy v systému.

Jelikož kotvy jsou v systému manuálně zaměřené velice přesně, je možné tyto vzdálenosti porovnat s výsledky TWR, abychom zjistili, jak moc se liší oproti skutečnosti. Pro kotvy Vista DirectFive totiž funkce TWR nebyla laděna. Problém spočívá v odlišné anténě, která má odlišnou charakteristiku a přidává jiný Antenna Delay. TWR měření je také velmi variabilní s různým natočením kotvy, což může vést ke špatnému výsledku. V případě 1D lokalizace jsou DirectFive kotvy natočené vždy přímo proti sobě, takže jev natočení by v tomto případě neměl být podstatný.

Rozdílná výška kotev na obou stranách uličky znamená, že výsledná délka letu signálu bude delší, tedy i hodnota změřená pomocí TWR. Vzdálenost TWR lze reprezentovat jako přeponu pravoúhlého trojúhelníku a rozdíl výšek je jeho odvěsna. Pro výpočet skutečné vzdálenosti kotev v půdorysu plánu je potřeba vypočítat druhou odvěsnu jednoduše pomocí Pythagorovy věty. Výsledky a porovnání skutečných vzdáleností a TWR měření lze vidět v tabulce 4.1.

První zajímavost, kterou lze vyčíst z výsledků, je vysoká podobnost nepřepočtených TWR se skutečnou vzdáleností ve dvou ze tří případů. V posledním případě ulička tvořená kotvami Sklad 3 a Sklad 6 disponuje navýšením o 0,1 m, což po přepočtu výškového rozdílu

Kotva A	Kotva B	Vzdálenost	TWR	TWR přepoč.
Sklad 1	Sklad 4	32,70 m	32,69 m	32,58 m
Sklad 2	Sklad 5	32,80 m	32,78 m	32.67 m
Sklad 3	Sklad 6	32,60 m	32,70 m	32,59 m

Tabulka 4.1: Tabulka ukazuje rozdíl mezi skutečnými vzdálenostmi mezi kotvami 1D uliček v logistickém skladu a naměřenými TWR daty. Ty jsou dále redukované o výškový rozdíl kotev.

dává hodnotu, která se velmi blíží skutečnosti. Konkretně se liší o 0,01 m. Zbylé dvě uličky po redukování TWR vzdálenosti kvůli výškovému rozdílu se odchylují o 12 - 13 cm oproti vzdáleností manuálně změřené a zanesené do plánu v aplikaci Sensmap.

Měření ukázalo i možnou kvalitu vazeb nejen mezi kotvami v rámci jedné uličky, ale i se sousedními kotvami. Jedním z problému automatického rozmisťování kotev pro 1D je metoda, jakým způsobem lze řešit definici uliček. Pro automatickou kategorizaci je nutné najít nějaký vzor, podle kterého lze klasifikovat z TWR výsledků skutečnost, zda daná kotva je s měřenou v jedné uličce. K tomuto úkolu nelze použít filtraci NLOS spojení jako v případě 2D instalace v Intemacu. V případě 1D bývají sousední kotvy montovány obdobně, a proto je dost pravděpodobné, že mají mezi sebou LOS. Jelikož 1D lokalizaci lze použít pouze s DirectFive kotvami, pak směrovost kotvy může pomoct odfiltrovat kotvy nepatřící do stejné uličky.

V tomto jsou výsledky TWR z logistického skladu velmi rozporuplné. Tabulka všech měření kotev ve skladu s naměřenou vzdáleností, množstvím úspěšných měření a celkovým skóre inicializace lze vidět na obrázku 4.5. U kotev, které společně tvoří uličku se očekávaly stejné hodnoty jako v případě jakéhokoliv jiného LOS spojení, tedy zejména TWR úspěšnost větší než 98 %. Tento předpoklad nebyl splněn u žádné z uliček. Dané porušení se však objevilo vždy jen u jedné z kotev uličky. Pokud byla iniciátorem TWR měření druhá kotva, pak hodnoty inicializace byly správné. Nelze s přesností určit z jakého důvodu se snížila úspěšnost, jelikož například ani výška kotev iniciátora měření není shodná pro všechny 3 případy.

Výsledky také ukazují, že kotva DirectFive dokáže spolehlivě synchronizovat a provádět TWR se sousedními kotvami vzdálenými několik jednotek metrů. Hodnoty inicializace a TWR jsou natolik dobré, že překonávají i hodnoty s protější kotvou, se kterou se tvoří ulička. Toto lze vidět například u inicializace kotvy *Sklad 1*, jejíž výsledky jsou pro sousední kotvu *Sklad 3* ve všech ohledech lepší než u kotvy *Sklad 4*, s níž uzavírá uličku.

4.2.3 Zhodnocení výsledků

Měření 1D části v logistického skladu splnilo pouze částečně svůj cíl a účel, avšak i přesto pomohlo navést na správné zaměření při návrhu algoritmu pro automatické umisťování kotev v podobných instalacích a zúžilo počet možných řešení problémů, které úloha vykazuje.

Z nezodpovězených otázek zbývá zejména určení úrovně přesnosti TWR na kotvách DirectFive. Výsledky naměřených vzdáleností byly velmi rozporuplné, v některých případech výsledky skoro přesně odpovídaly skutečnosti, jindy se odchylka pohybovala kolem 10 cm. Předpokladem správného měření byla znalost přesných skutečných vzdáleností mezi kotvami. Ta byla převzata z konfigurace serverové aplikace RTLS Studio, avšak tyto hodnoty nebyly osobně ověřené, takže odchylka 10 centimetrů mohla být také způsobená například i lehkou nepřesností v zaměření kotev. Pokud vycházíme z předpokladu, že známá vzdále-

		Sklad 1 049162C94924	Sklad 2 049162C93188	Sklad 3 049162C9871F	Sklad 4 049162C99FBB	Sklad 5 049162C9109C	Sklad 6 049162C98B02
Iniciátor	Sklad 1 049162C94924		6.42 m 190/200 Score: 91.08	N/A	32.69 m 186/200 Score: 88.60	N/A	N/A
	Sklad 2 049162C93188	6.47 m 199/200 Score: 90.72		6.27 m 198/200 Score: 89.04	N/A	32.78 m 199/200 Score: 91.58	N/A
	Sklad 3 049162C9871F	N/A	6.27 m 200/200 Score: 89.29		N/A	N/A 0/200 Score: 15.01	32.70 m 199/200 Score: 91.41
	Sklad 4 049162C99FBB	32.69 m 200/200 Score: 88.31	N/A	N/A		6.35 m 199/200 Score: 90.10	25.92 m 1/200 Score:18.70
	Sklad 5 049162C9109C	N/A	32.78 m 174/200 Score: 91.77	N/A 0/200 Score: 14.68	6.35 m 198/200 Score: 90.91		6.34 m 199/200 Score: 89.12
	Sklad 6 049162C98B02	N/A	N/A	32.71 m 187/200 Score: 91.93	25.63 m 1/200 Score: 31.50	6.35 m 198/200 Score: 88.66	

Obrázek 4.5: Tabulka výsledků inicializace a TWR s kotvami v logistickém skladu.

nost odpovídala skutečné, pak i odchylka 10 cm je stále přijatelná. Přestože odpověď na otázku přesnosti není úplně známá, stále lze výsledky interpretovat tak, že nebyl shledám kritický problém v přesnosti TWR, který by zamezil reálné použitelnosti chystané funkce.

Cílem měření bylo rovněž zjistit, zda lze v TWR datech najít vhodnou korelaci pro automatickou detekci uliček pro 1D. Výsledky ukázaly, že v případě LOS mezi sousedními uličkami je u podobně krátkých vzdáleností, jako v tomto logistickém skladu, složité najít vhodný parametr, jelikož UWB spojení i mezi sousedy bylo dostatečně kvalitní. Bez sofistikovanějšího klasifikátoru je obtížné specifikovat informaci ohledně stavby uliček.

Kapitola 5

Koncepce automatického rozmisťování referenčních jednotek

Cílem funkce automatického rozmisťování referenčních jednotek je zejména zjednodušit a zefektivnit instalaci kotev Sewio RTLS systému. Podobné funkce, které usnadňují použití systému v různých prostředích, mohou být významnou konkurenční výhodou na trhu. Tato kapitola představí části funkce zaštiťované názvem Auto-Deployment a popíše jejich účel a způsob použití.

Mezi tyto podfunkce patří kontrola a detekce chyb v nastaveném rozložení kotev a automatické rozmístění kotev pro 1D i 2D lokalizaci. Rozmisťování kotev pro oba typy lokalizace je principiálně velmi odlišné, proto je potřeba k těmto funkcím přistupovat odděleně.

5.1 Detekce nesprávného zaměření kotev

Jelikož zaměřování kotev v Sewio RTLS bude v budoucnu nutné provádět současně i manuálně pro velkou část prostředí a instalací, je tato funkce určena zejména pro snadnou detekci chyb zanesených postupným měřením vzdáleností mezi jednotlivými kotvami. Vzdálenosti mohou být také změřeny správně, ale lidská chyba často nastává i během procesu, kdy se zaměřené kotvy vkládají do plánu aplikace Sensmap, a proto mohou být některé kotvy také pouze přehozené.

Tato funkce nemá za cíl uživatele od tohoto procesu zcela oprostit, ale pomůže v případě, kdy po zanesení kotev systém RTLS nefunguje anebo je velmi nepřesný. Detekce chyb v zaměření kotev je post-processing[1] funkce, která využívá TWR data získaná během procesu inicializace kotev.

Z hlediska chybných zaměření kotev, které by měly být detekovány tímto systémem, byly vybrány dvě nejčastější varianty. Mezi tyto chyby patří nepřesnost měření vzdáleností mezi kotvami a přehození kotev. Existují i další způsoby, kterými lze poškodit instalaci, ale většina z nich vyžaduje znalostní přesah požadavků na lokalizaci a prostředí. Takové chyby je nutné kontrolovat většinou vizuálně.

Funkce by měla být použita následovně. Uživatel systému musí nejprve zavést všechny kotvy na předem plánované místo. Všechny kotvy připojí do sítě se serverem, na kterém běží RTLS Studio. Dále provede inicializaci kotev za účelem nalezení master kotev. V této fázi se naměří TWR pro kontrolu instalace v pozdější fázi. Jelikož je instalace nevhodná pro automatické rozmístění kotev, uživatel kotvy zaměří sám a zanese je do plánu v aplikaci

[1] zpracování dat nezávisle po jejich uložení

Sensmap. Pokud RTLS nefunguje správně, spustí funkci pro kontrolu instalace. Ta porovná aktuální stav s poslední inicializací a vrátí výsledek, ve kterém uživatel nalezne informace o nepřesném zaměření anebo o prohozených kotvách. Uživatel upraví plán podle výstupu a následně může znovu dodatečně provést kontrolu instalace po aplikaci změn.

5.1.1 Nepřesné vzdáleností mezi kotvami

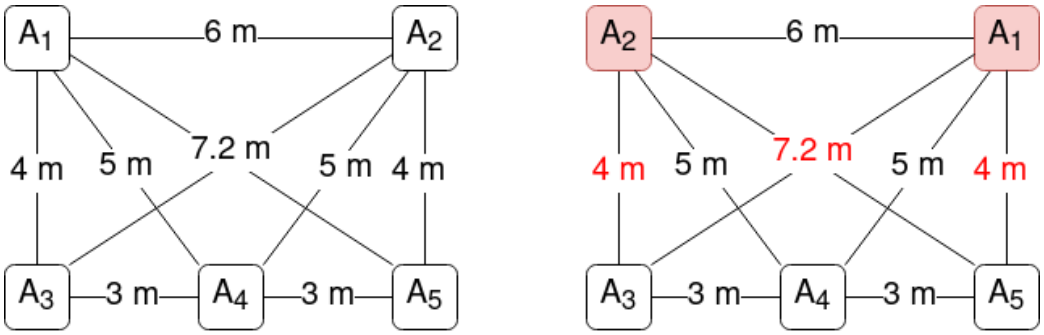
Smyslem této detekce je upozornit uživatele na nepřesné vzdáleností mezi kotvami. Informace je získána z porovnání skutečných vzdáleností zanesených v aplikaci Sensmap a naměřených pomocí metody TWR z posledního procesu inicializace.

Před porovnáním vzdáleností je nutné odstranit zavádějící TWR měření, která jsou sama o sobě chybná. K filtraci těchto dat lze využít závěr měření z kapitoly 4, tedy poměr počtu úspěšných měření z celkových a směrodatná odchylka všech dvou set naměřených vzdáleností mezi dvěma kotvami. Tolerance mezi měřením a aktuální konfiguraci je nastavena na 20 cm.

5.1.2 Prohozené kotvy

Kotvy mohou být prohozené z důvodu špatné identifikace jednotlivých zařízení. V systému jsou kotvy identifikovány pomocí MAC adresy, ale dále k nim může být přiřazen alias, který vylepšuje rozlišovací schopnost identity kotev. Samotná MAC adresa není vhodná, jelikož se může lišit například pouze v jednom oktetu, což může být lehce přehlédnuto.

Princip detekce tohoto problému spočívá ve vztahu mezi prohozených kotvami a s kotvami v jejich okolí. Prerekvizitou správné detekce je, aby všechny kotvy v systému byly zaměřené správně i v případě prohození. Poté ve většině reálných případů by mělo platit, že TWR vzdálenost mezi prohozenými kotvami je podle předpokladu v toleranci s aktuální konfigurací kotev v plánu. Naopak vzdálenosti s dalšími sousedními kotvami budou v případě dvou prohozených kotev nesprávné. Toto pravidlo platit nemusí s kotvou, která má shodnou vzdálenost s prohozenými kotvami. Vyobrazení správných a nesprávných vzdáleností v případě prohození dvou kotev lze vidět na obrázku 5.1.



Obrázek 5.1: Porovnání vzdáleností v případě prohození dvou kotev.

Tato funkce slouží k pomocným účelům, a proto není nutné, aby detekovala 100 % pozitivních případů prohození kotev. Důležité je, aby nenastala situace, která vrátí false positive stav, který by nesprávně označil dvě kotvy jako prohozené, což by však nebyla pravda a uživatel by na popud systému provedl změnu, která by vytvořila chybu. Během testování bude nutné se zaměřit na reálnou použitelnost této detekce a v případě neuspokojivých výsledků se zamyslet nad jejím vylepšením.

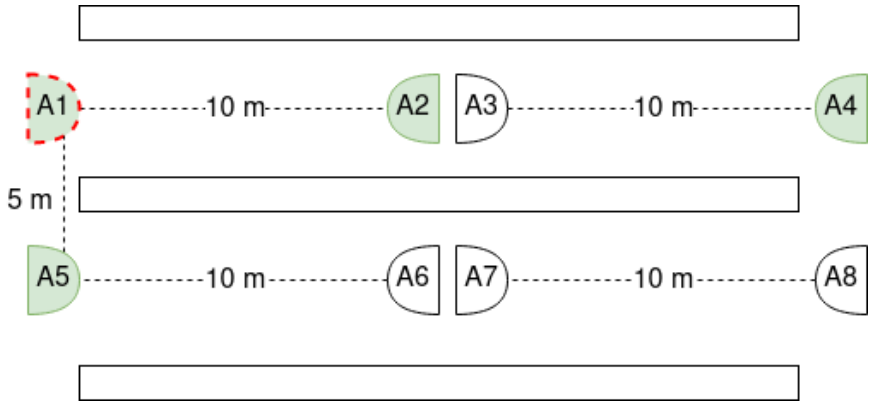
5.2 Automatické rozmisťování kotev pro 1D

Funkce automatického rozmisťování kotev pro 1D lokalizaci má sloužit zejména pro zjednodušení a zefektivnění Sewio RTLS systému. Lokalizace typu 1D funguje na principu TDOA, ale pouze pro jednu dimenzi určenou prostorem mezi dvojicí kotev, které definují takzvanou 1D uličku. Jak už i z názvu vypovídá, tento způsob určování pozic je určený pro úzké koridory, uličky v budovách anebo uličky mezi regály ve skladech.

Zaměřování 1D buněk je jednodušší než v případě plné 2D lokalizace. Pokud i půdorys plánu budovy obsahuje dostatečně detailní informace o uličkách, pak technik, který instaluje kotvy systému, musí změřit zejména vzdálenost mezi kotvami a poté ukotvit do plánu počáteční bod uličky. Pokud jsou však kotvy instalovány například ve více než 10 metrech, pak proces měření vzdáleností mezi kotvami bez vhodné techniky může být nebezpečný a bez asistence i obtížně proveditelný.

Tuto fázi zaměřování systému může nahradit TWR měření mezi kotvami. Naměřená vzdálenost, kterou kotva vrátí, určuje vzdušnou vzdálenost, ale ta promítne-li se do 2D plátna, nebude přesná, pokud kotvy nebyly nainstalovány ve stejné výšce. Pro získání skutečných vzdáleností ve 2D půdorysu, musíme znát výšky obou kotev a jejich rozdíl.

Dalším problémem, který vyvstává pro správné použití funkce, je otázka, jakým způsobem definovat, které dvě kotvy tvoří uličku. Z toho důvodu bylo provedeno TWR měření v reálné 1D instalaci se třemi uličkami ve skladu oddělenými regály. TWR měření ukázalo, že z rádiových výsledků inicializace a TWR není možné jednoznačně určit, které dvě kotvy tvoří uličku. Kotvy, které vedle sebe uzavírají jiné paralelní uličky, mohou mít také velmi dobré výsledky inicializace a mohou být i lepší než v případě párové kotvy z uličky viz obrázek 5.2.



Obrázek 5.2: Příklad problému automatické definice uliček. Kotva A1 má dobré radiové parametry pro kotvy A2, A4 i A5.

Definice uliček i výšky kotev a jejich rozdíly je náročné definovat automaticky, takže v navrhované verzi Auto-Deploymnetu bude potřeba tyto informace dodat ručně uživatelským zásahem s možností rozšíření v budoucnu o plně automatický způsob.

5.2.1 Návrh použití v RTLS Studio

Běžný postup zprovozněni 1D lokalizace vyžaduje následující kroky: instalace a zapojení kotev, konfigurace kotev v RTLS Manager, inicializace kotev, vytvoření plánu v aplikaci

Sensmap, změření zanesení kotev do plánu, nastavení výšek kotev a definice Aisles v aplikaci Sensmap. Poté byli-li všechny kroky provedeny správně, lokalizace typu 1D začne fungovat.

Jelikož samotná konfigurace 1D lokalizace vyžaduje manuálně definovat uličky a výšky kotev, pak by bylo vhodné, aby uživatel nebyl nucen tuto činnost provádět hned dvakrát, jednou pro auto-deployment a podruhé pro samotnou lokalizaci. I z důvodu implementace a architektury systému je žádoucí, aby operace definující jednotlivé elementy v aplikaci Sensmap, byly nastavovány právě tam využitím již hotového GUI.

Aby bylo možné využít již dostupné rozhraní, ať už webové či aplikační, je nutné změnit pořadí prováděných operací uživatelem pro zprovoznění systému 1D. Součástí inicializace kotev je i TWR měření a první verze Auto-Deploymentu, která ukládá výsledky inicializace a vypočtených pozic kotev do databáze v tabulkách *initialization* a *initialization_details*. Inicializace musí být rozšířena o deployment typu 1D. Aby inicializace měla všechny dostupné informace potřebné pro deployment, musí uživatel předtím, než spustí tento proces, jít do aplikace Sensmap a definovat obvyklým způsobem uličky a výšky kotev. Uživatel poté spustí inicializaci, která na základě TWR vytvoří rozložení kotev použitelné v Sensmap.

5.3 Automatické rozmisťování kotev pro 2D

True Location neboli 2D lokalizace je nejčastějším využitím systému Sewio RTLS. Metoda určování pozice pomocí TDOA vyžaduje přesnou synchronizaci hodin na kotvách, ale také je velmi citlivá na přesnost zaměření kotev v systému, jež by mělo být s přesností v jednotkách centimetrů. Pokud je některá z kotev v systému výrazně vychýlena, pak netrpí pouze přesnost a správnost lokalizace tagů, ale výrazně se snižuje i počet vypočtených pozic. Nesprávně zaměřená kotva zanáší do výpočtu větší chybu, která pak může přesáhnout limitní hodnotu a taková pozice je nakonec zahozena.

Chyb, které může technik při instalaci RTLS provést, je několik, ale většina z nich vyžaduje nutnost vidět fyzicky prostředí a znát přesný use case dané instalace. Přestože existují profesionální zaměřovací laserové nástroje (Total Station), které umožní rychlé, přesné a pohodlné zaměření kotev do plánu budovy, samotná schopnost systému toto provést i bez této techniky je obrovským uživatelským i marketingovým přínosem.

Pro uživatele seznamující se Sewio RTLS anebo pro ty, kteří demonstruji schopnosti takového systému, je zde funkce Auto-Deployment. Ta se pomocí TWR měření, která vrátí vzdálenosti mezi kotvami, snaží rozvrhnout kotvy tak, aby jejich rozložení odpovídalo realitě a toto rozložení jednoduše zanést kamkoli do plánu a zprovoznit lokalizaci. Avšak jak již bylo zmíněno v sekci 3.2.1, tato funkce má několik limitací.

5.3.1 Limitace Auto-Deployment gen1

Funkce Auto-Deployment je integrovaná do procesu inicializace. Pokud je povolena, pak během inicializace je každá kotva přepnuta do režimu TWR, kde provede 200 měření se svými sousedy. Poté se zavolá funkce rozmisťování kotev na základě těchto TWR měření. Funkce pracuje následovně:

1. Filtrace duplicitních TWR měření tak, aby každý TWR pár se vyskytoval pouze jednou.

2. Zpracování TWR dat do struktury se vzdálenostmi mezi párem pro efektivnější vyhledávání.

3. Generování počátečních kombinací kotev (trojic).
4. Filtrovaní počátečních kombinací s cílem najít takovou kombinaci, která dokáže se svými vzdálenostmi vytvořit trojúhelník, který neobsahuje úhly větší než 95°.
5. Výpočet fixních pozic počáteční kombinace kotev.
6. Hledání vhodné neumístěné kotvy pro výpočet. Pro tuto kotvu musí být k dipozici nejméně 3 TWR měření - rádiusy od umístěných kotev a vybere se ta kotva, která má nejbližší průměrný rádius.
   - Pokud žádná kotva splňující tyto podmínky nebyla nalezena, pak skok na bod 10.
   - Jinak pokračovat bodem 7.
7. Výpočet pozice vybrané kotvy.
8. Odstranění vypočtené kotvy se seznamu "neumístěných"kotev.
9. Pokračování od bodu 6.
10. Zpracování výsledku výpočtu všech kotev

Tento algoritmus je zobrazen aktivity digramem na obrázku A.1. Prvním problémem, který zanáší chybu do výpočtu pozice kotvy v reálných instalacích, je chybějící filtrace NLOS párů. Nepřímá viditelnost u TWR způsobuje kromě snížené stability měření i větší neměřenou vzdálenost, jelikož i čas letu signálu v takovém prostředí je delší. To způsobí, že taková kotva je často velmi odskočena oproti ostatním.

Další limitací pro použití v reálné instalaci je počítání dvourozměrné pozice pomocí vzdáleností, které jsou ovlivněny výškami kotev. Informace o vzdálenosti mezi kotvami se využívá v různých fázích běhu algoritmu, a tudíž by bylo vhodné předzpracovat vzdálenosti, které by odpovídaly rozdílu ve výškách. Stejný problém vzniká i v případě 1D umisťování kotev, takže může být řešen obdobně.

První generace Auto-Deploymentu umožňuje rozmístit pouze malou část kotev. To je způsobeno zejména tím, že výpočet se provádí pouze pro jednu z počátečních kombinací kotev. To znamená, že zvolením jedné trojice kotev, která splňuje výše uvedené podmínky, omezíme výpočet rozložení pouze pro ty kotvy, které mají viditelnost na právě jednu vygenerovanou kombinaci počátečních kotev a případně dalších již vypočtených kotev v rámci aktuálního běhu algoritmu Auto-Deployment. V případě, že lokalizované prostředí je velmi otevřené s dostatečnou viditelností a propojeností kotev, pak je i teoreticky možné vypočítat celou instalaci v jednom běhu. K tomu je nutné vybrat i ideální počáteční kombinaci, která je v tomto případě vybrána pouze náhodně, tedy první výsledek splňující geometrické podmínky. Pokud algoritmus běží ve velmi členitém prostředí, kde jednotlivé lokalizační buňky jsou uzavřené a oddělené, pak výsledkem Auto-Deploymentu je rozložení pouze pro některou z lokalizačních buněk.

Samotný výpočet v první generaci algoritmu se snaží vypočítat pozici neznámé kotvy pomocí všech dostupných rádiusů (TWR vzdáleností) od kotev, které mají svou pozici v plánu již určenou. V případě pokročilé fáze výpočtu, kdy počítaná oblast obsahuje větší počet kotev, například 10 a více, může být tento způsob kontraproduktivní. Pokud například 2 z 10 TWR rádiusů jsou určitým způsobem chybné, pak i výsledek může být ovlivněn touto chybou. Z hlediska výpočtu pozice je pro přesnost výsledku stěžejní výběr menšího počtu

kotev, které však dávají správnější data. V ideálním světě dává trilaterace jeden konkrétní výsledek a tedy souřadnice neznámého bodu. V reálném světě toto nenastane, ale snažíme se najít výsledek, který si nejméně rozporuje a vykazuje nejmenší rozptyl a chybu.

5.3.2 Návrh změn pro Auto-Deployment gen2

Cílem navrhovaných změn je výše uvedené limitace co nejvýrazněji zredukovat tak, aby se z funkce určené pro demonstrativní účely stala funkce použitelná ve velkém množství instalací Sewio RTLS.

Problém s nevhodnými TWR měřeními je stejný jako v případě funkce pro kontrolu umístění kotev a 1D Auto-Deployment. Řešení je proto také stejné a je založeno na provedeném měření ve výzkumném centru Intemac. To ukázalo spojitost mezi parametry výsledku TWR měření a NLOS spojeními, kterým se chceme vyhnout. Tato spojení je nutné v aktuálním návrhu filtrovat, jelikož na přepočet těchto dat nemáme dostatečné informace o prostředí a tedy způsobu, jakým bylo měření ovlivněno. Spojitost byla nalezena v parametrech počtu přijatých měření a variaci vzdáleností reprezentující směrodatnou odchylku. Byly stanoveny limitní hodnoty pro NLOS: pro počet úspěšných iterací bylo určeno minimum 98 % a pro směrodatnou odchylku bylo stanoveno maximum 0,1. Data o vzdálenostech porušující tyto parametry musí být odstraněna.

Rozdílné výšky kotev jsou druhým společným problémem nových funkcí pro vytváření a kontrolu rozmístění kotev. Vzdálenost kotev ve 2D plánu lze jednoduše vypočítat pomocí aplikace Pythagorovy věty, stejně jako v předešlých případech. Je potřeba také vyřešit, jakým způsobem budou systému informace ohledně výšek kotev předány. Stejný problém byl řešen i v případě 1D verze algoritmu. Ta vyžaduje, aby uživatel nejprve v aplikaci Sensmap nakonfiguroval výšky všech kotev a až poté přešel v RTLS Manager k inicializaci kotev. Aby se sjednotil pracovní a technologický postup obou variant algoritmu, vyžaduje 2D Auto-deployment stejným způsobem ruční nastavení Z souřadnic kotev v systému. Pokud některá z kotev nemá tuto informaci, pak vzdálenost s takovou kotvou nebude nijak upravována a zachová se hodnota z TWR.

Co se týče uživatelského hlediska, je takové řešení přijatelné. Výška všech kotev by měla být přesně definovaná již před plánovou instalaci kotev v areálu a to ve fázi návrhu projektu. Bohužel se stále jedná o uživatelskou interakci, která může být chybná.

Aby bylo možné vytvořit rozmístění kotev pro celou instalaci, je nutné vyřešit několik dílčích problémů, které způsobí to, že běh algoritmu nelze provést pro všechny kotvy v systému. Řešením je rozdělit instalaci do několika částí a pro každou z nich pak vypočítat samostatný Auto-Deployment. Způsobů jimiž může být instalace rozdělena, je několik. Jedním z nich by mohlo být rozdělení na základě TWR stability a variability použité pro filtrování NLOS spojení. Z měření v Intemacu vyplynulo, že v mnoha případech došlo k oddělení vazeb mezi dvěma různými buňkami. Tento způsob rozdělení buněk však nemusí být dostatečný, jelikož v obecném pojetí může dojít k vytvoření buněk, které sice dobrou viditelnost mezi sebou za určitých podmínek mají, ale výrazně se liší v geometrii, výškách a lokalizovaném prostředí. Tyto informace má pouze uživatel, a proto je schopný logicky rozdělit lokalizované oblasti tak, aby dávaly smysl nejen pro samotnou lokalizaci, ale i pro funkci Auto-Deploymentu.

Uživatel ohraničuje lokalizační buňky pomocí funkce Anchor Restrictions. Ta je dostupná v aplikaci Sensmap a umožňuje pro každého mastera vymezit kotvy, které synchronizuje a využívá pro lokalizaci. Tímto způsobem se zabraňuje tomu, aby se pro výpočet pozice nepoužily ty kotvy, které jsou například oddělené nějakou fyzickou překážkou a nebylo

s nimi počítáno pro určování pozice v daném místě. Tato funkce je velmi vhodná, protože její motivace a účel je shodný s tím, co potřebujeme pro Auto-Deployment. Docílíme tím logické oddělení kotev, které mají vzájemnou viditelnost pro synchronizaci i lokalizaci a je tedy vhodné i pro měření vzdáleností mezi kotvami a hledaní jejich vzájemné polohy.

Z toho vyplývá další vylepšení, což je rozdělení instalace do takzvaných buněk a počítání Auto-Deploymentu pro každou buňku zvlášť. Tím docílíme výsledku, že v několika iteracích běhu algoritmu získáme pozice ve většině případech v celé instalaci anebo v její valné většině.

Použití Anchors Restriction pro Auto-Deployment má další vliv na pracovní postup při instalaci lokalizačního systému. Kromě ruční konfigurace výšek kotev před inicializaci kotev se musí také nadefinovat restrikce. Ty se nastavují na stejném místě jako výšky, takže v části Sensmap. Restrikce se nastavují tak, že pro každou master kotvu se vyberou sousední kotvy. To lze provést buď kliknutím myši na jednotlivé kotvy nebo výběrem ze seznamu.

Restrikce jednoho mastera může obsahovat i jiné mastery anebo kotvy, které patří do jiných restrikcí. Aby se předcházelo tomu, že jedna kotva bude mít vypočítanou pozici pro více Auto-Deploymentů, je nutné s touto možností počítat. Řešením je spojování restrikcí do jedné buňky. To znamená, že pokud dvě restrikce obsahují dvě společné kotvy, pak z principu trilaterace jsme schopni tyto buňky spojit do jedné. Z uživatelského hledisku tato funkce může být výhodná, jelikož uživatel pak nemusí umisťovat hodně malých buněk, ale méně větších.

V původní verzi se k výpočtu samotné pozice využívají všechny dostupné radiusy, tedy vzdálenosti, od známých kotev. Jak bylo zmíněno v předchozí sekci, takový přístup od určitého počtu kotev nepřináší větší přesnost a správnost pozice, ale spíše vede k zanesení chyby od jedné čí více kotev, které již nenabízí nejsprávnější data. Nachází-li se v buňce například 15 kotev, z nichž je k dispozici 14 TWR rádiusů pro pro výpočet 15. kotvy, původní verze algoritmu Auto-Deployment hledá jeden společný bod, ve kterém se všechny dostupné rádiusy protínají. Z toho důvodu je potřeba najít způsob, jak definovat menší počet kvalitnějších vstupních dat.

Kvalitu výstupu určuje v první verzi průměrná vzdálenost všech radiusů od vypočteného bodu. Během hledání nejvhodnější kombinace vstupních dat je právě hodnota průměrné odchylky od výsledku klíčový parametr. Podobný princip je uplatněn i v RTLS Serveru, který počítá pozice tagů na základě TDOA. Ve výchozím nastavení se počítá pozice tagu s maximálně šesti kotvami a minimálně se čtyřmi. Minimum 4 je nastaveno z důvodu výrazně vyšší přesnosti pozic, než už teoretického minima 3. Pokud je v oblasti s tagem dostatečný počet kotev, pak RTLS server vybere 6 kotev s nejlepšími hodnotami parametrů RSSI a First Path a vypočte pozici tagu. Pokud je pozice vypočtena s větší chybou, než je parametr Optimum Error Threshold, pak se server pokusí vypočítat pozici znovu s jinou kombinací kotev. Nepodaří-li se výpočet s dostatečnou přesností v další iteraci, je výpočet pozice ukončen. Je to z toho důvodu, že se jedná real-time lokalizační systém a hledání dalších možných kombinací by zabralo mnoho času, pokud ještě uvážíme, že v systému běží najednou stovky tagů.

Delší doba výpočtu není v případě operace typu Auto-Deployment tolik kritická jako v případě RTLS. Jedná se o jednorázovou operaci, která se obvykle využije pouze jednou při instalaci systému, případně dále při jejím rozšiřování. Tudíž si v algoritmu můžeme dovolit strávit více času hledáním optimálnějšího řešení.

Nový vylepšený algoritmus bude proto omezovat počet radiusů pro výpočet pozic kotev od 3 do 6. Minimum tří kotev není možné stejným způsobem navýšit jako v případě RTLS Serveru, jelikož na začátku výpočtu jsme schopni geometricky přesně určit pozice pouze 3 kotev. Čtvrtá kotva se tedy bude počítat běžným způsobem a vyžívá pozice počátečních

3. Pokud bude k výpočtu neumístěné kotvy k dispozici vice než 6 TWR radiusů, pak algortimus vygeneruje všechny kombinace možných šestic. Pro každou šestici se vypočítá pozice a vybere se ten výpočet, který vykázal nejmenší chybu, tedy průměrnou odchylku rádiusů od vypočteného bodu. Pokud by žádná z šestic nevykazovala dostatečně nízký error, menší než nastavený limit, pak se algoritmus pokusí udělat totéž, ale nyní se všemi možnými pěticemi.

Algoritmus pokračuje do té doby, dokud nenajde vhodné řešení anebo neotestuje všechny možné trojice kotev pro výpočet neumístěné kotvy. Avšak pokud nastane situace, že pro kotvu nebylo nalezeno vhodné řešení, ale jsme schopní vypočítat pozici jiné kotvy, pak poté se výpočet pozice vrátí zpět k předchozí neúspěšně umístěné kotvě a zkusí ji vypočítat znovu s novými rádiusy, které mohou dostatečně zpřesnit výpočet.

Kapitola 6

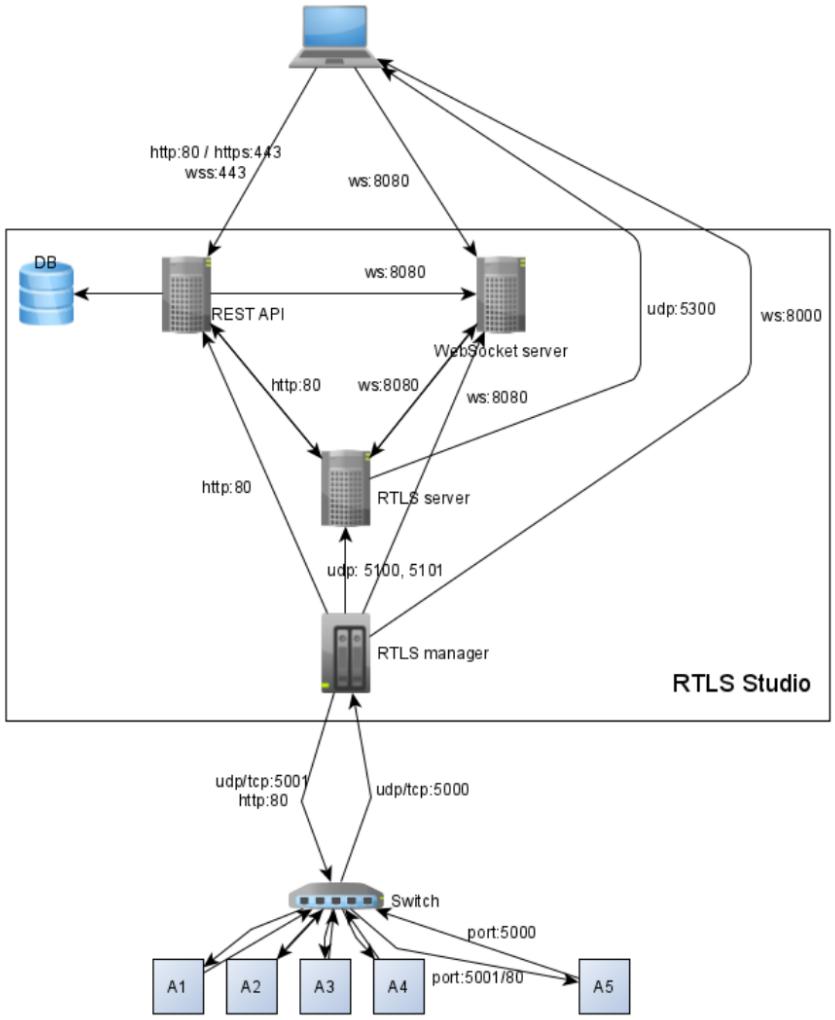
Implementace

V této kapitole je popsáno, jakým způsobem byly implementovany návrhy algoritmů pro detekci nesprávného zaměření kotev a automatické rozmisťování kotev pro 1D a 2D lokalizaci. První část kapitoly se zaměří na systém RTLS Studio, zejména z pohledu architektury a technologií, ze kterých je tento systém tvořen. Tyto informace budou potřebné pro pochopení, kde a jak jsou jednotlivé dílčí funkce implementovány a integrovány do celého systému. Následující část se věnuje implementaci detekce chybných umístění kotev. Je zde popsáno, jakým způsobem je funkce implementovaná, jaké přijímá vstupy a jaké vrací výstupy. Další sekce se zabývají samotnou funkcí Auto-Deploymentu. Ten je rozdělen do částí pro 1D a 2D lokalizaci, jelikož obě varianty vyžadují odlišný způsob zanášení kotev do plánu. Stále ale obě funkce mají společný účel a jsou také integrovány společně.

Serverová aplikace RTLS Studio se skládá z několika služeb, kde každá z nich je postavená na odlišné technologii. RTLS Studio je serverová aplikace s webovým rozhraním. Jako HTTP server je využit Apache2. Základ webové aplikace, která běží na tomto serveru, je postavená na PHP frameworku Laravel. Tato část je zaštítěna službou *sensmapserver*. Ta se stará o autentizaci, logování a REST API pro práci s MySQL databázi *sensmapserver*. Dále obsahuje i WebSocket server pro notifikaci o změnách ve reálném čase. Z pohledu funkce RTLS jsou stěžejní aplikace a služby *rtlsmanager* a *rtlsserver*. Aplikace *rtlsmanager* je node aplikace napsaná v Typescriptu a její hlavní funkcí je spravování kotev, řízení synchronizace, inicializace a kontrola a konfigurace tagů. Přes RTLS Manager prochází v podstatě veškerá komunikace na kotvy i z kotev. S kotvou komunikuje na portu 80 pro její konfiguraci a dále na portech 5000 a 5001 pro přijímaní zpráv reportů a synchronizace kotev. Reporty z kotev RTLS Manager přijímá, vytváří z nich statistiky a dále odesílá na RTLS server pro výpočet pozic tagů. Služba *rtlsserver* zajišťuje tedy lokalizaci. Z důvodu potřeby optimálního výkonu aplikace je napsaná v jazyce C. RTLS server předává vypočtené pozice pomocí WebSocket serveru a v určitých dávkách ukládá pozice do MySQL databáze. Architekturu RTLS Studio a Sewio RTLS celkově z pohledu síťových portů a vzájemných komunikací lze vidět na obrázku 6.1.

6.1 Detekce nesprávného zaměření kotev

Princip a smysl funkce pro detekci chybných zaměření kotev byla popsána v předchozí kapitole. Jedná se o oddělenou funkci, která nepatří přímo k funkci Auto-Deployment, ale využívá TWR data z inicializací, aby detekovala chyby aktuálního zaměření kotev způsobené často uživatelskou chybou. Funkce má na základě vybraných dat ohodnotit nepřesnost



Obrázek 6.1: Architektura RTLS Studio z pohledu síťové komunikace. Převzato z [7].

vzdáleností a pokud je to možné i detekovat přehozené kotvy. To využijí jak uživatelé, ale i Sewio Customer Experience tým při vzdáleném hledání problémů v lokalizaci instalace zákazníka.

Jedná se tedy o operaci, která zpracovává data post-procesově. Jelikož funkce není nutná pro samostatný Auto-Deployment a jejím účelem je dodatečná kontrola nastavení dat v databázi *sensmapserver*, bylo vhodné zakomponovat tuto operaci do PHP Laravel aplikace, který pohání core webové aplikace RTLS Studio.

Aby bylo možné funkci jednoduše spouštět z různými parametry odkudkoli, musí funkce, která implementuje logiku detekce chyb zaměření kotev, být napojená endpointy na REST API serveru. Díky tomu lze funkci jednoduše integrovat anebo spouštět pomocí jiných REST API klientů, například aplikace Postman.

Ve webovém aplikačním frameworku Laravel se REST API definuje metodami třídy *Route*. GET požadavek na kontrolu inicializace v aplikaci RTLS Studio lze definovat jako v příkladu 6.1. Aby se definice route správně načetly po startu serveru, je nutné umístit do adresáře PHP souboru *routes/api.php*. Soubor *api.php* může být rozdělen do více částí a každá část oddělená pomocí vnořeného podadresáře. Případě RTLS Studo jsou endpointy

pro REST API sensmapserveru umístěny v souboru *routes/api/sensmapserver/api.php*. V tomto souboru jsou definované výstupní body pro práci s inicializací a jsou zde také přidány body pro kontrolu rozložení kotev.

```
Route::middleware("permission:get from api,api")->get(
    "/initializations/{id}/check",
    "InitializationController@checkDeployment");
```

Výpis 6.1: Příklad definice GET endpointu v laravel route

Funkce pro kontrolu a detekci chyb v rozmístění kotev v instalaci pomocí TWR funguje následovně.

1. Načíst datastreamy kotev z databáze.
2. Extrahovat pozice kotev.
3. Vypočítat vzdálenosti párů kotev.
4. Načíst data z inicializace.
5. Extrahovat TWR páry z výsledků inicializace.
6. Porovnat seznamy párů vzdáleností s TWR páry a vytvořit seznam se správnými a špatnými vzdálenostmi.
7. Vyhledat přehozené kotvy na základě seznamu s dobrými a špatnými vzdálenostmi párů kotev.
8. Připravit JSON výstup.

Kotvy a tagy jsou v databázi sensmapserver aplikace RTLS Studio uloženy do společné tabulky *feeds*. Kotvy a tagy mají některé společné vlastnosti, ale například tagy poskytují mnohem více informací do aplikace Sensmap, například hodnoty senzorů. Všechny tyto parametry, jako jsou souřadnice X, Y, Z anebo hodnoty senzorů, jsou uložené v tabulce *datastreams*. Ta obsahuje pro každou hodnotu každého feedu záznam, který je referencován pomocí feed ID.

Pro načtení aktuálních informací o kotvách, je nutné udělat složený SELECT dotaz na tabulky *feeds* a *datastreams* a vybrat ty datastreamy, které patří feedům obsahujícím *title* záznam s MAC adresou některé z kotev poslední inicializace.

Výpočet vzdálenosti nastavených kotev je pouze jednoduchým výpočtem vektoru dvou bodů a jeho délky. V tomto případě již počítáme s trojrozměrným vektorem a tedy i výškou kotev. V dalších fázích algoritmu detekce chyb porovnáváme aktuálně nastavené vzdálenosti s TWR daty, které udávají přímou vzdušnou vzdálenost mezi kotvami, proto vzdálenosti kotev musí být také ve 3D.

Funkce pro extrakci TWR hodnot do párů kotev s jejími vzdálenostmi obsahuje také filtr nepřesných NLOS měření. Ten je nastaven podobně jako v jiných částech systému způsobem, který odstraňuje vzdálenosti určené méně než 98 % úspěšných měření a vzdálenosti se směrodatnou odchylkou větší než 0,05.

Po zpracování dat z databáze sensmapserver a dat z TWR inicializace se tyto dvě skupiny dat porovnávají pro vytvoření takzvaných správných a špatných párů kotev. Pár kotev je označen za správný, pokud vzdálenosti mezi dvěma kotvami odpovídá naměřené hodnotě

z TWR s tolerancí do 20 cm. Zbylé páry jsou zařazeny jako špatné. Seznam špatných párů se dále ukládá do JSON objektu, který vrací REST API.

Seznamy správných a špatných kotev se využívají i pro evaluaci prohozených kotev. To řeší funkce *evaluateSwappedAnchors*. Tato funkce prochází všechny správné páry. Pokud se nalezne kotva, která je v správném páru pouze s jednou kotvou a vyskytuje se ve více než v jednom špatném páru, pak se totéž ověří pro onoho jednoho správného oponenta. Pokud se i on vyskytuje pouze v jednom správném páru a v jednom a více špatných párech, pak tyto dvě kotvy jsou označeny za prohozené. Funkce nakonec vrací pole všech prohozených párů.

Výsledky kontroly chyb zaměření kotev jsou uloženy do objektu, který je serializován do JSON struktury, vrácené GET požadavkem na REST API. Objekt obsahuje ID inicializace použité pro kontroly správnosti rozmístění kotev, dále pak datum provedené inicializace, seznam špatných párů kotev a seznam prohozených kotev. Příklad výstupu lze vidět ve výpisu 6.2. Tento příklad ukazuje zhodnocení instalace s pěti kotvami, kde kotvy *6827198ECC40* a *6827198F060C* jsou prohozené. V části *wrongMeasures* jsou páry se všemi pěti kotvami, avšak prohozené kotvy vykazují chybu měření se zbylými třemi kotvami. Tři zbylé kotvy, které jsou umístěné správně, vykazují chybu s prohozenými kotvami.

```
{
    "id": "5",
    "at": "2022-03-06 17:27:38",
    "wrongMeasures": {
        "6827198F2528": ["6827198F060C","6827198ECC40"],
        "6827198ECAB5": ["6827198ECC40","6827198F060C"],
        "6827198ECC40": ["6827198ECAB5","6827198F2528","6827198F5EEF"],
        "6827198F060C": ["6827198F2528","6827198ECAB5","6827198F5EEF"],
        "6827198F5EEF": ["6827198F060C","6827198ECC40"]
    },
    "swappedAnchors": [
        ["6827198ECC40","6827198F060C"]
    ]
}
```

Výpis 6.2: Příklad JSON objektu vracené funkcí na kontrolu rozmístění kotev.

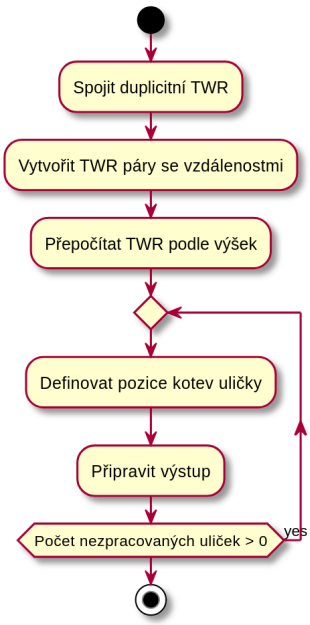
6.2 Automatické rozmisťování kotev pro 1D

Princip automatického rozmisťování kotev pro 1D lokalizaci je popsán v kapitole 5.2 pojednávající o návrhu a koncepci funkcí. Cílem je tuto funkci zakomponovat do procesu inicializace tak, aby fungovala podobně jako první generace Auto-Deploymentu. Motivací toho je využití aktuálního GUI v Sensmap pro zanášení kotev do plánu s minimálními úpravami.

Z toho důvodu se implementace 1D Auto-Deploymentu nachází v části *rtlsmanager* a je napsaná v Typescriptu. Funkce, která řídí celý proces rozmisťování kotev, je nazvaná *calculateDeployment1D* a je volaná z části inicializace, která vyhodnocuje pro každou kotvu její výsledky. Během inicializace se každá kotva přepne po určitý čas do role mastera. Výsledkem toho je informace, jak dobře dokáže každá kotva zasynchronizovat své sousedy. Pokud je zapnuta funkce Auto-Deployment, pak k informaci schopnosti kotvy jako mastera synchronizovat se přidá informace o pozici vypočtené pomocí TWR.

V původní verzi Auto-Deploymentu se vypočtená pozice přiřazuje k detailům inicializace pro daného mastera. Pozice je uložena jako hodnota jednoho sloupce v tabulce *initialization_details* a je ve formátu "*x;y*". Pokud pozice tohoto daného mastera není vypočtena, pak hodnota uložená v tabulce je "*N/A*". Takovou inicializaci dokáže načíst i funkce Auto-Deployment v Sensmap.

Funkce *calculateDeployment1D* přijímá v parametrech funkce inicializační data, seznam uliček definovaných v Sensmap a feed informace o kotvách z databáze *sensmapserver*. Pro účely automatického rozložení kotev je důležitá pouze informace ohledně výšky kotvy. Diagram aktivity funkce pro 1D rozmisťování kotev lze vidět na obrázku 6.2.



Obrázek 6.2: Aktivity diagram 1D Auto-Deploymentu

Jelikož 1D lokalizace vyžaduje ke své činnosti LOS mezi dvěma směrovými kotvami DirectFive, získat dostatek dat pro umísťování několika uliček najednou je velmi náročné. Proto i z návrhu funkce vyplynulo, že 1D instalace se musí rozkládat jednotlivě po uličkách. Uličky musí být definované v Sensmap před spouštěním inicializace s Auto-Deploymentem.

Každá 1D ulička tvoří samostatné rozvržení se dvěma kotvami. Při sestavování rozložení kotev se první kotva vždy vkládá na pozici [0;0], tedy do počátku plánu. Druhá kotva se posune od počáteční pozice na ose X ve vzdálenosti od prvního bodu. V tomto případě se tedy použije hodnota z TWR přepočítaná na základě rozdílu výšek kotev.

Každé z těchto malých rozložení se musí kvůli zpětné kompatibilitě se starým Auto-Deploymentem jevit jako samostatná inicializace. Takto to je uloženo i v databázi. Pro každou uličku je vytvořena inicializace, která je referencovaná v detailech inicializace pro obě kotvy uličky. Takové inicializace lze poté postupně načítat v Sensmap a zanést do plánu pomocí funkcí přesun, rotace a přehození. Příklad výsledku 1D Auto-Deploymentu pro 3 uličky lze vidět v tabulce 6.1.

Webové rozhraní pro zanášeni kotev v Sensmap do plánu vyžadovalo malé úpravy, aby umožňovalo umisťovat inicializaci se dvěma kotvami. Samotná funkce byla omezena pro rozložení o třech kotvách. Při pokusu o načtení inicializace s menším počtem, byla vy-

id	initialization_reference	master	overallscore	position
49	5	049162c94924	89.84	0;0
50	5	049162c99fbb	65.70	32.57830719973031;0
51	6	049162c9109c	71.62	32.66861490788981;0
52	6	049162c93188	90.45	0;0
53	7	049162c9871f	65.24	0;0
54	7	049162c98b02	70.70	32.58834147360065;0

Tabulka 6.1: Tabulka *initialization_detail* po provedení 1D Auto-Deploymentu v instalaci logistický sklad. V tabulce chybí sloupec *details*, který obsahuje JSON strukturu s výsledky inicializace a TWR.

psaná chybová hláška. Po úpravě této podmínky však proces havaroval a ukončil fungování aplikace Sensmap.

Po výběru inicializace pro Auto-Deployment se v plánu objeví obdélník ohraničující rozložení kotev. To vyžaduje nejprve výpočet konvexní obálky a poté výpočet Oriented Minimum Bounding Boxu. Obě metody byly v původní implementaci použitelné pro rozloženi dvou kotev. Pro použití tohoto nástroje bylo potřeba upravit vykreslování obdélníku pro bounding box a výpočet pozice středu obdélníku pro ovládací prvky posouvání.

6.3 Automatické rozmisťování kotev pro 2D

True Location neboli 2D lokalizace, představuje z pohledu umisťování kotev mnohem obtížnější problém, než ve variantě 1D. Princip funkce Auto-Deploymentu je vysvětlen v sekci 5.3. Tato funkce staví na původní implementaci funkce Auto-Deployment, která byla určena zejména pro rychlé sestavení funkce RTLS pro demo účely s pěti kotvami. Nová implementace má za cíl zmírnit anebo odstranit limitace původní verze, aby byla použitelná pro libovolný počet kotev.

Přestože v návrhu 1D a 2D verze funkcí pro rozmisťování kotev se postupovalo odděleně, jsou obě implementace integrované do procesu inicializace, tedy v službě *rtlsmanager*. I přes algoritmické oddělení obou funkcí je možné sdílet některé části s implementací 1D algoritmu. Mezi společně využívané funkce patří:

- *removeTwrDuplicates* - funkce pro sloučení duplicitních měření
- *createTwtPairs* - vytvoření párů kotev z TWR měření
- *recalculateTwrPairsByHeightDiff* - funkce pro přepočet vzdáleností na základě rozdílu výšek kotev

Celý proces umisťování kotev do plánů je řízen funkcí *calculateDeployment2D*. Ta v parametrech funkce přijímá data inicializace, aktuálně nastavené restrikce a feed informace o kotvách. Stejně jako v případě 1D jsou z feed informacích vzaty pouze údaje o výškách kotev. Funkce pro 2D je volaná ve stejné fázi inicializace jako varianta 1D a vrací i stejný datový typ. Tím je zajištěna integrace obou funkcí do procesu inicializace.

První věc, v čem se varianta 2D liší od 1D, je v načítání restrikcí místo uliček. Restrikce slouží k tomu, aby oddělily proces umisťování kotev do plánu pro kotvy, které mají mezi sebou potřebnou viditelnost. Restrikcemi se kotvy rozdělují do různých lokalizačních buněk.

Přesně to dělá i funkce pro 2D Auto-Deployment, a proto se na základě restrikcí vytvoří buňky kotev. Jelikož buňky mohou sdílet společné kotvy, pak pokud dvě restrikce sdílí dvě a více kotev, je možné tyto dvě buňky pozičně spojit.

Spojování buněk je implementováno ve funkci *mergeConnectedCells*. Jedná se o funkci, která pomocí rekurze postupně spojuje buňky kotev. Funguje tak, že v cyklu hledá první dvě buňky, které odpovídají podmínkám spojení. poté tuto dvojici spojí a vloží do nového pole buněk. Všechny ostatní buňky se dále vloží do nového seznamu a funkce *mergeConnectedCells* se znovu zavolá nad tímto novým seznamem. Pokud funkce nenalezla dvě buňky vhodné pro spojení, pak se vrátí aktuální seznam.

Poté se na základě definovaných buněk spouští proces výpočtu pozic kotev. Ten probíhá iterativně pro každou buňku. V první fázi výpočtu se vytvoří všechny kombinace trojic kotev dané buňky. Poté se pro každou trojici vyhodnotí, zda z dostupných dat o trojici kotev jsme schopni vytvořit trojúhelník, což je provedeno podmínkou, že dvě strany trojúhelníku musí být delší než 3. strana. Dále se kontrolují úhly daného trojúhelníku. Ty musí splňovat to, že žádný z nich není větší než 95°. Úhly vnitřních stran trojúhelníku jsou vypočteny podle Kosinovy věty. Po verifikaci všech kombinací počátečních kotev pomocí funkce *filterThreePointsByAngles* se vybere nakonec ta kombinace kotev, která má pro všechny 3 kotvy nejmenší rozdíl výšek.

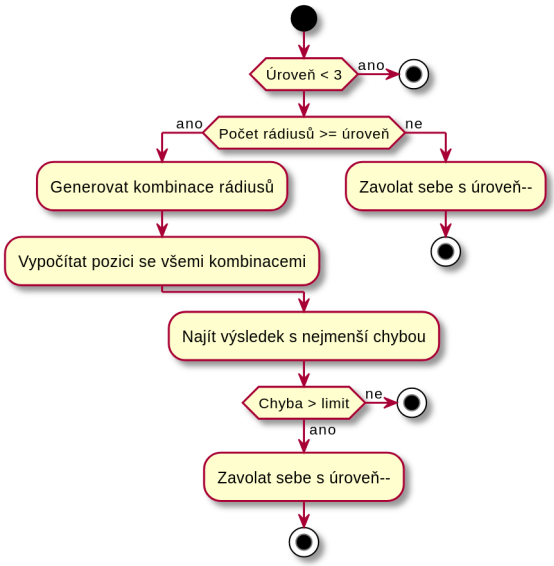
Pro vybranou počáteční trojici se vypočte fixní pozice. První kotva se umístí do počátku souřadného systému, tedy do bodu [0;0]. Druhá kotva se ve vzdálenosti od první kotvy posune na ose X. Pozice na ose Y je u druhé kotvy stále 0. Výpočet třetího bodu je implementován zjednodušením dvou rovnic Pythagorovy věty o dvou neznámých, kterými funkce vypočte pozici 3. bodu na ose X a z té pak dále dosazením do Pythagorovy věty získáme souřadnici Y.

Po zafixování počátečních kotev výpočtu se iterativně vybírá ze seznamu nezařazených kotev buňky a vybere se ta, která má nejmenší průměrnou vzdálenost k již zaneseným kotvám.

Po výběru vhodné kotvy volá funkce pro výpočet pozice dané kotvy, která se jmenuje *calcWithElimination*. V parametrech funkce se předávají rádiusy, MAC adresa neznámé kotvy, úroveň výpočtu pozice a maximální limit akceptovatelné chyby. Rádiusy v trilateraci reprezentují kružnici okolo známého bodu, na které leží neznámý bod a hledáme průnik několika radiusů, abychom našli přesnou pozici bodu. Rádiusem v systému pro automatické rozložení kotev je pouze TWR vzdálenost mezi známou a neznámou kotvou. Úroveň výpočtu pozice určuje s kolika kotvami se má pozice vypočítat. Smyslem toho je mít konfigurovatelný výpočet kotev v celém průběhu výpočtu. Jak už z návrhu vyplynulo, je výhodnější počítat pozice s menším počtem kvalitních radiusů. V aktuální implementaci je maximum kotev nastaveno na 6, avšak tento parametr je jednoduše upravitelný konstantou *maxCirclesToCalc* v *autoDeployment.ts*.

Funkce tedy nejprve zkouší výpočet pozice kotvy s počtem známých kotev daný úrovní aktuálního volání funkce. Pokud není k dispozici dostatek známých kotev, rekurzivně se znovu zavolá s dekrementovanou úrovní výpočtu. Pokud je úroveň menší než 3, což je nedostatečný počet vzdáleností pro výpočet pozice trilaterací, funkce vrátí nedefinovaný výsledek. Pokud výsledek výpočtu pozice má úroveň chyby větší než limit, pak se funkce znovu rekurzivně volá se sníženou úrovní. Nakonec pokud se výpočet vykoná a úroveň chyby je ve správných mezích, pak funkce vrátí danou pozici. Proces výpočtu pozice jedné kotvy je znázorněn diagramem na obrázku 6.3.

Jelikož trilaterace s daty z TWR nevrátí jedno přesné řešení, pro samotný výpočet pozice se využívá algoritmus Levenberg–Marquardt. Jedná se o metodu, která řeší problémy



Obrázek 6.3: Aktivity diagram funkce *calcWithElimination*.

nelineárních nejmenších čtverců a v případě Auto-Deploymentu hledá optimální řešení trilaterace. Algoritmus je využit z knihovny *ml-curve-fitting* dostupné z npm online repozitáře[1]. Funkce, kterou algoritmus Levenberg–Marquardt optimalizuje, je výpočet euklidovské vzdálenosti mezi umístěnou a neumístěnou kotvou. Vzdálenost a pozice umístěné kotvy je známá, takže výsledkem výpočtu je pozice neznámé kotvy.

Po výpočtu pozice pomocí algoritmu Levenberg–Marquardt se kontrolují vzdálenosti TWR rádiusu se vzdáleností oproti vypočtenému bodu. Tyto rozdíly se agregují do hodnoty průměru, který je využit jako indikátor chyby výpočtu.

Pokud výpočet pozice vrátil nedefinovanou hodnotu, pak daná neznámá kotva se zařadí do seznamu nerozhodnutelných kotev. Pokud v dalších iteracích se výpočet kotvy podaří, pak obsah seznamu nerozhodnutelných kotev se přesune zpět do seznamu kotev pro výpočet pozice.

Iterace výpočtu pozic pro buňku končí, když seznam kotev k výpočtu je prázdný. V případě celkového procesu Auto-Deploymentu se čeká, dokud se neprovede proces výpočtu pro všechny definované buňky. Funkce *calculateDeployment2D* vrací pole objektů *Positions*, který reprezentuje asociativní pole s klíčem MAC adresy kotvy a hodnotou pozice kotvy. Výstupní typ je shodný s 1D verzi Auto-Deploymentu, proto výstupní seznamy obou funkcí se dále spojí a uloží do databáze pro aplikaci rozložení kotev v Sensmap.

[1] https://www.npmjs.com/package/ml-curve-fitting

Kapitola 7

Testování a zhodnocení funkce

Tato kapitola pojednává o testování a ověřování schopnosti vyvíjeného systému implementovaného podle popisu v kapitole 6. V první části této sekce je ověřena funkce detekce chyb v rozmístění kotev a automatické rozmisťování kotev pro 1D a 2D lokalizaci. V případě detekce chyb se zaměřujeme zejména na algoritmus detekce prohozených kotev, kde je potřeba ověřit známé limitace řešení. Automatické rozmísťování kotev pro 1D je jednoduchá funkce, která neskrývá žádná funkcionální rizika. Metoda je přesná tak, jak je přesná metoda TWR. Z toho důvodu část o 1D variantě funkce zejména ověří navrhovaný uživatelský postup a to, jak zapadá do celé koncepce systému RTLS. V případě 2D varianty se testování zaměří na konkrétní vylepšení a jejich viditelný dopad na výsledek rozložení kotev v systému. V druhé části této kapitoly jsou shrnuty limitace aktuálního řešení a návrh možných změn a vylepšení do budoucna.

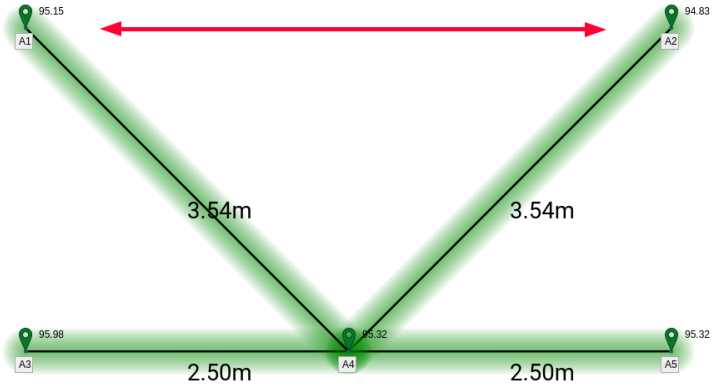
7.1 Ověření detekce prohozených kotev

Hlavní logika funkce pro detekci špatných zaměření kotev je součástí detekce prohozených kotev. Tato funkce hledá danou chybu na základě seznamu párů kotev, které odpovídají svou vzdáleností hodnotě z TWR měření a seznamu těch páru, které svou vzdáleností neodpovídají.

Z principu funkce, která je popsána v kapitole 5.1.2, vyplývá situace, kdy funkce nebude fungovat správně. Jedná se o situaci, kdy lze prohozené kotvy zaměnit s jinou dvojicí kotev se zachováním stejných vstupních dat. Podobné situace byly testovány pomocí simulací testovacích vstupů na funkci *evaluateSwappedAnchors()*. První takový nejednoznačný případ lze vidět na obrázku 7.1. Přestože jsou kotvy A1 a A2 prohozené, stále zaujímají správnou vzdálenost vůči kotvě A4, což způsobí, že tyto prohozené kotvy nebudou detekovány.

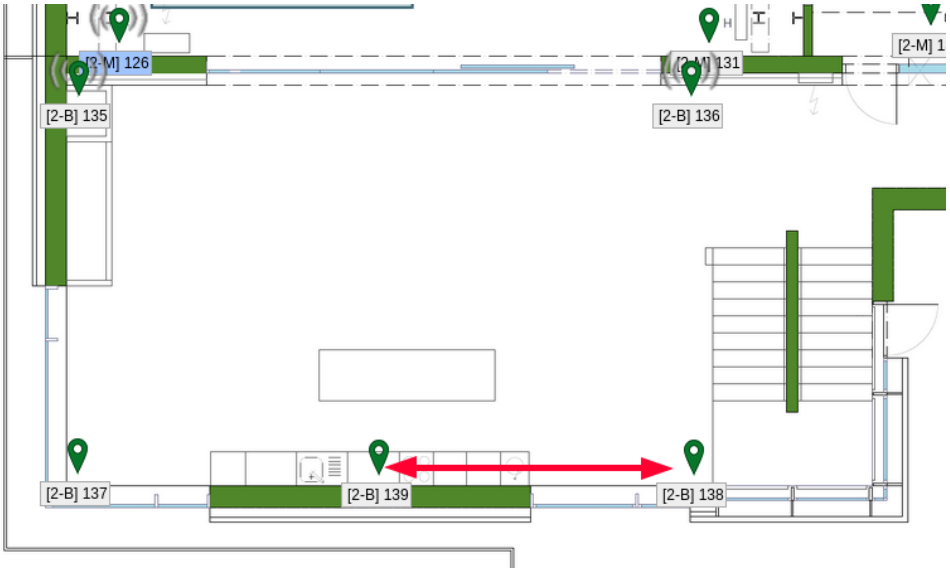
Když se však narušila symetrie mezi prohozenými kotvami posunem kotvy A4, pak systém už byl schopný správně detekovat prohození kotev A1 a A2.

V instalaci Intemac se nachází mnoho lokalizačních buněk podobné koncepce, tedy 4 kotvy v rozích a pátá kotva ve střední části delší stěny. V jedné z nejpřesněji zaměřených buněk byla otestována tato funkce. V prvním testu byly prohozeny dvě sousední kotvy, které nemají mezi sebou výše popsanou symetrii viz obrázek 7.2. Systém pro detekci přehozených kotev by měl spolehlivě detekovat tento pár. Vyměněny byly kotvy *[2-B] 138* a *[2-B] 139*. S výchozí tolerancí 0,2 m pro kontrolu vzdáleností se očekávaný výsledek nevrátil. Prohozené kotvy byly detekovány až s tolerancí 0,3 m. Zvýšení tolerance bylo nutné z důvodu většího rozdílu TWR dat oproti skutečnosti v této lokaci.



Obrázek 7.1: Rozvržení kotev, které po prohození A1 a A2 vrací false negative výsledek.

Poté bylo vyzkoušeno prohození kotev *[2-B] 138* a *[2-B] 139*. V tomto případě nebylo dosaženo správného výsledku vůbec, jelikož nepřesnosti ve vzdálenosti vyžadovaly větší toleranci, která však v tomto případě způsobila to, že jemné rozdíly ve vzdálenostech mezi kotvami *[2-B] 137*, *[2-B] 137* a *[2-B] 139* se smazaly a v této reálné situaci nastal problém symetrických vzdáleností stejně jako na obrázku 7.1.



Obrázek 7.2: Nesymetrická výměna kotev *[2-B] 138* a *[2-B] 139* nebyla správně detekována z důvodu velkých rozdílů TWR dat oproti skutečnosti.

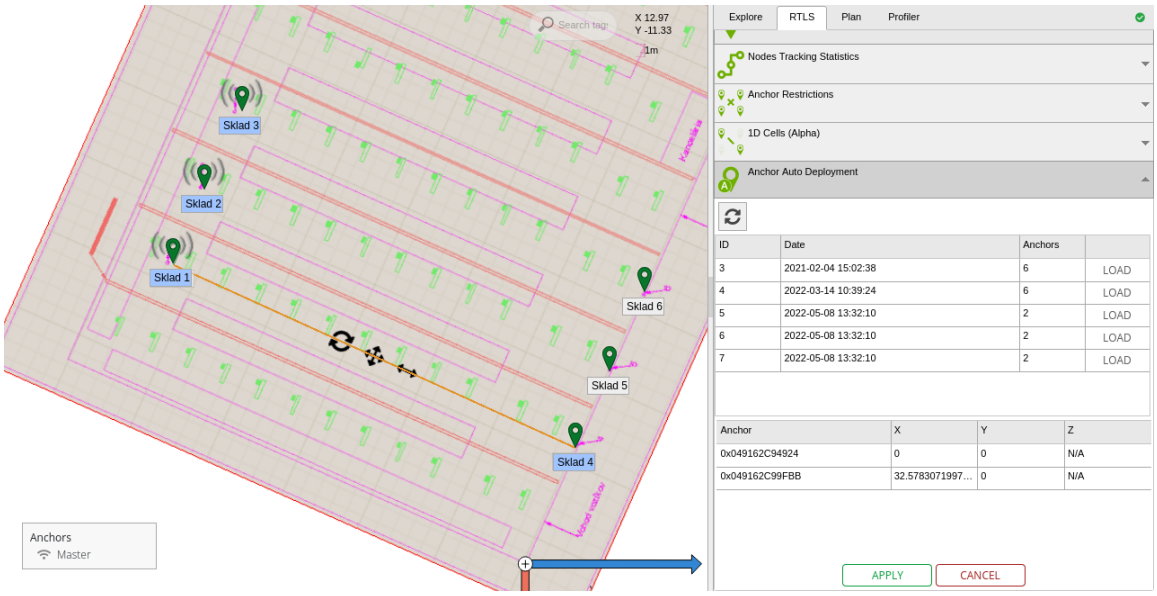
Testování navržené detekce prohozených kotev ukázalo, že ke správnému fungování vyžaduje co možná nejvyšší přesnost zaměření kotev, které hlavně odpovídá výsledkům TWR. V takovém prostředí je možné snížit toleranci rozdílů vzdáleností a tím také vylepšit schopnost detekce. Použitelnost detekce prohozených kotev v náročném a nepřesně zmapovaném prostředí je diskutabilní.

7.2 1D Auto-Deployment

Auto-Deployment ve variantě 1D byl testován vůči datům z instalace logistického skladu představené v kapitole 4.2, kde se TWR data z této instalace využila pro vyhodnocení použitelnosti metody pro Auto-Deployment. Díky tomu lze porovnat a ověřit výsledek programu s výsledky měření, jelikož byla použita stejná data.

Funkce Auto-Deploymentu správně vytvoří 3 inicializace se dvěma kotvami. V každé položce Auto-Deploymentu pro 1D je druhá kotva vždy posunutá na ose X o vzdálenost mezi kotvami. Po načtení takové inicializace se v plánu správně objeví obě kotvy a pomocí ovládacích prvků lze kotvy posunout na požadované místo. Čára propojující obě kotvy v režimu konfigurace pomáhá ke správnému natočení obou kotev. Toto lze vidět na obrázku 7.3.

Z 3 uliček, které jsou k dipozici, byl průměrný rozdíl Auto-Deploymentu oproti skutečnosti -0.088 m. Konkrétní hodnoty rozdílů TWR a skutečných pozic je shodný s tabulkou 4.1. Rozdíl u všech 1D buněk je maximálně 12 cm, což je na vzdálenost více než 30 m pro reálné využití naprosto dostatečné. Na obrázku 7.4 vypadá rozdíl délek oproti skutečnosti o 3 cm menší, což vzniklo z důvodu nepřesnosti při grafickém umisťování kotev do plánu. V tomto případě není umístění kotev z pohledu lokalizace úplně ideální, protože je mezi kotvami uličky výrazný výškový rozdíl. Je doporučeno, aby se kotvy instalovaly do stejné výšky. Proto se očekává, že v obvyklých případech by TWR mělo být ještě přesnější.



Obrázek 7.3: Pohled na Sensmap po načtení uličky Sklad1 - Sklad4.

7.3 2D Auto-Deployment

Tato funkce staví na předešlé implementaci, kterou vylepšuje a rozšiřuje tak, aby nabídla řešení, které bude použitelné pro nastavovaní nových RTLS instalací anebo rozšiřování těch původních. Testování 2D systému pro rozmisťování kotev využívá data z několika instalací, kde každá z nich nabízí jiný druh prostředí a rozmístění kotev. Testování se zaměřuje na jednotlivá vylepšení a jejich efekt na výsledek rozmístění kotev.

Obrázek 7.4: Detail rozdílu délky uličky změřené pomocí TWR metody aplikované v Auto-Deploymentu. Načtené rozložení kotev je zarovnané s kotvou Sklad 4. Rozdíl na kotvě Sklad 1 je zhruba 15 cm.

7.3.1 Kompenzace výškového rozdílu

Velké výškové rozdíly mohou způsobit výraznou nepřesnost zaměření, která může dosahovat i jednotek metrů. Takto nepřesně umístěná kotva může výrazně ovlivňovat lokalizaci, ale také proces Auto-Deploymentu, jelikož může ovlivnit další kotvy ve výpočtu a tedy zanést další chybu v rozložení kotev.

Funkce pro kompenzaci výšek vyžaduje, aby uživatel před spuštěním inicializace nastavil v Sensmap výšky kotev. Rozdílem výšek jsou poté přepočítané TWR vzdálenosti, které se použijí k výpočtu pozice kotvy. Výrazný vliv má rozdíl výšek v případě instalace s malými vzdálenostmi. Z toho důvodu test výškové kompenzace je proveden v malém testovacím prostředí kanceláře se vzdálenostmi mezi kotvami do 5 m.

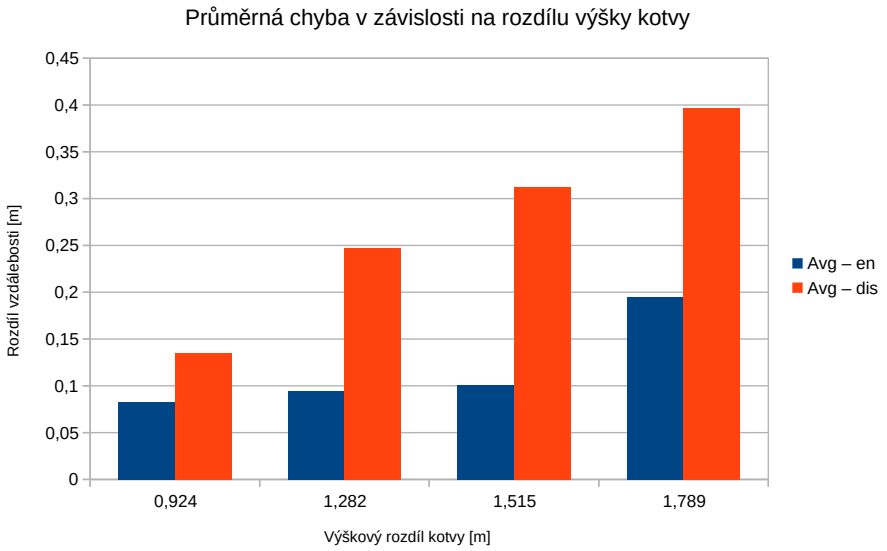
Kompenzace výšek byla testována v prostředí RTLS se 6 kotvami. Místnost, ve které byly kotvy umístěny, byla čtvercového půdorysu, kde jedna trojice kotev byla umístěna na jedné straně místnosti a druhá polovina šestice na straně opačné. Jedna z kotev byla připevněna na stativu, který umožnil jednoduše měnit výšku kotvy. Během měření se provedlo několik inicializací, kde pro každou z nich se umístila daná kotva do jiné výšky.

Pro referenční umístění kotev se vyžilo laserové zaměřovací zařízení, které vrátilo DXF soubor s rozestavěním kotev. Toto rozestavění bylo aplikované do Sensmap a referenční data ze sensmapserver databáze byla použita pro ověřování výsledků Auto-Deploymentu. Výsledky rozmístění kotev pro tyto inicialializace s rozdílnou výškou jedné kotvy jsou vyobrazeny grafem na obrázku 7.5.

Tyto výsledky ukazují, že kompenzace výšek udržuje určitou hladinu přesnosti na podobné hodnotě, většinou do 10 cm. Průměrná chyba je reprezentována jako rozdíl vzdáleností dvou kotev na základě skutečných pozic a pozic vypočteným Auto-Deploymentem. Vyšší průměrná chyba nastává v případě výškového rozdílu 1,789 m. To je způsobeno tím, že tak velký výškový rozdíl u blízko umístěných kotev, jako bylo v případě testovacího prostředí, způsobí, že úhel natočení kotvy a její antény již není optimální a tedy i metoda TWR není dostatečně přesná. V případě vypnuté kompenzace výšek kotev se chyba rozložení kotev zvyšuje s každým navýšením rozdílu výšek kotev.

7.3.2 Filtr nepřesných TWR měření

Způsob filtrování NLOS spojení byl určen na základě TWR měření provedeném v instalaci Intemac. Toto měření je popsáno v sekci 4.1. Prostředí instalace Intemac je velmi členité



Obrázek 7.5: Průměrný rozdíl vzdáleností oproti kotvě s rozdílnou výškou. Hodnota Avg - en označuje závislost při použití kompenzace výšek. Pro závislost Avg - dis je kompenzace vypnutá.

a často rozdělené do samostatných místností, které jsou pokryty pomocí 5 až 6 kotev. Místnost označená kotvami s prefixem *[2-M]* je hala pokrytá 7 kotvami, které však neposkytují přímou viditelnost ke všem ostatním kotvám, jelikož lokalizovaná oblast má charakter písmene L.

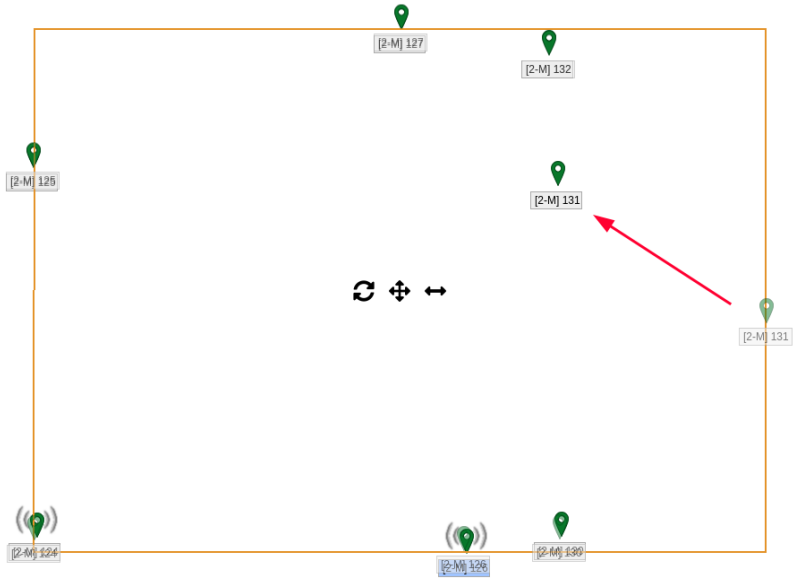
Toto testovací prostředí bylo vhodné k testování nepřímých spojení kotev, protože původní verze funkce Auto-Deployment nedokázala v této místnosti spolehlivě rozmístit kotvy přesně z tohoto důvodu. Roh, který vede do jiné části budovy, skrývá kotvy, které byly v původní verzi výpočtu pozic kotev výrazně odskočené, což často ovlivnilo i pozice jiných kotev.

Pro porovnání reálného fungování filtrů NLOS spojení je využita inicializace kotev právě této místnosti. Vyhodnocení užitečnosti filtru NLOS je provedeno pomocí porovnání výsledků funkce Auto-Deployment se zapnutým a poté vypnutým filtrem. Filtrem se myslí mechanismus, který odstraňuje všechna TWR se směrodatnou odchylkou větší než 0,1 m a s procentuální úspěšností menší než 90 % měření.

Výsledek testu je znázorněn na obrázku 7.6. První poznatek z ověřování funkčnosti nové verze Auto-Deploymentu je, že systém byl schopný spolehlivě a přesně umístit všechny kotvy v této místnost. Tento výsledek nastal v případě běhu algoritmu se všemi navrhovanými funkcemi. Po vyřazení funkce filtru nedůvěryhodných TWR vzdáleností bylo výsledné rozložení, kromě jedné kotvy, také přesné. Výjimkou byla kotva *[2-M] 131*, která byla posunuta od správné pozice o 7,4 m. Výsledek testu ukázal pozitivní vliv vylepšení na výsledné rozložení kotev.

7.3.3 Umísťování kotev rozsáhlých instalací

K testování automatického rozmisťování kotev v rozsáhlých instalacích byla použita data z instalace, která nabízí otevřené prostředí se 40 kotvami, které jsou v celé instalaci vzájemně



Obrázek 7.6: Podkladové kotvy reprezentují rozložení s použitím filtru NLOS. Druhá řada transparentních kotev je Auto-Deployment bez použití filtru. Kotva *[2-M] 131* je výrazně posunuta od správné pozice.

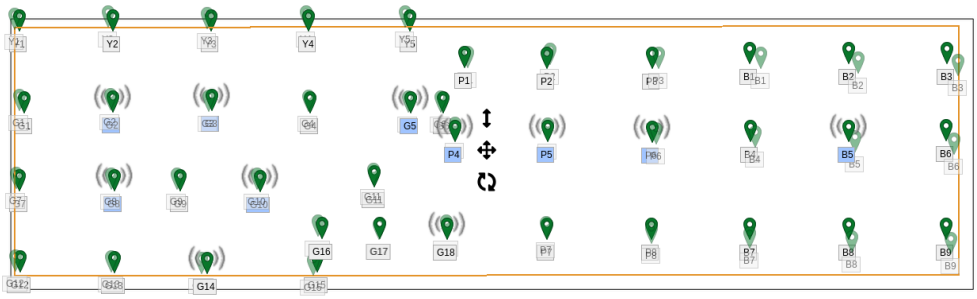
propojené. Teoreticky v tomto prostředí je možné udělat Auto-Deploymment pro celou instalaci najednou, kvůli vhodné vzájemné propojenosti kotev. Všechny kotvy jsou ve stejné výšce 3,46 m.

Referenční pozice kotev v této instalaci jsou změřeny přesně pomocí laserového zaměřovacího zařízení. Vůči těmto pozicím se v rámci testování a ověřování výsledku běhu rozmisťování kotev počítá rozdíl vzdáleností mezi všemi kombinacemi kotev a porovnávají se vzdálenosti s referenčními hodnotami.

Pokud chceme vypočítat rozložení kotev v jedné iteraci algoritmu, tedy jako jednu buňku, je nutné buď odstranit všechny restrikce anebo vytvořit restrikce tak, aby se prolínaly pomocí nejméně dvou kotev. To zajistí, že restrikce se propojí do jedné velké. Poskytnutá data již obsahovala i restrikce pro funkční lokalizační systém. Jelikož lokalizační buňky jsou v celé této instalaci vhodně propojené, vhodně vytvořené byly i restrikce pro Auto-Deployment.

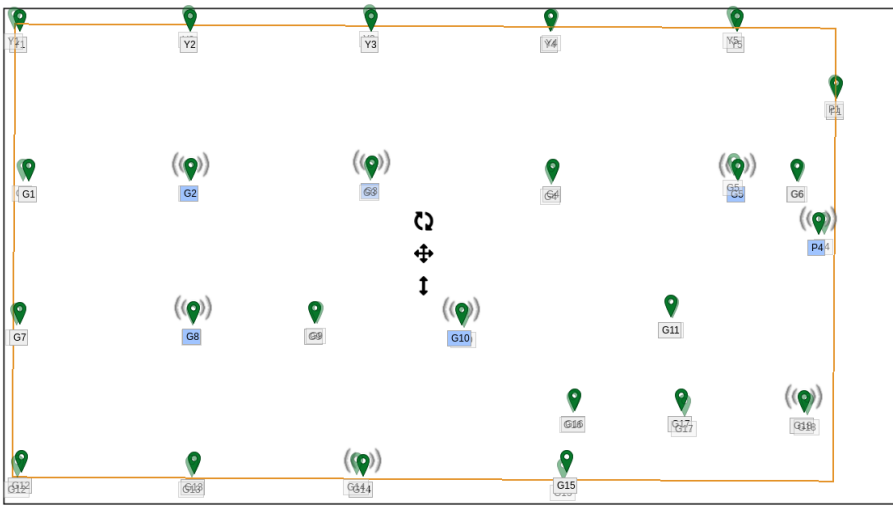
Výsledek rozmisťování kotev pomocí TWR dat v případě konfigurace restrikcí pro jednu iteraci Auto-Deploymentu nejprve přinesl velmi rozporuplné výsledky. Průměrný rozdíl vzdáleností mezi kotvami oproti skutečnosti byl vyhodnocen na 14,80 m, což jednoznačné vysvětloval grafický pohled rozložení vypočtených pozic kotev. První část instalace se podařilo rozložit správně a dostatečně přesně, avšak v polovině nastává výrazný problém, který následně otočí celý zbytek instalace kotev. Tuto situaci způsobila kotva G6, která měla chybně nastavenou výšku. Po korekci výšky této kotvy výsledek Auto-Deploymentu již odpovídal skutečnému rozložení kotev této instalace. Výsledek lze vidět na obrázku 7.7.

Výpočet Auto-Deploymentu pro 40 kotev v tomto prostředí s dostupnými daty se však stále ukázal být nedostatečně přesný, jelikož průměrná odchylka vzdáleností mezi všemi kotvami byla 0,75 m. Již v návrhu se s touto situací počítalo a jejím řešením by ve většině případů mělo být rozdělení výpočtu do několika samostatných rozložení kotev.



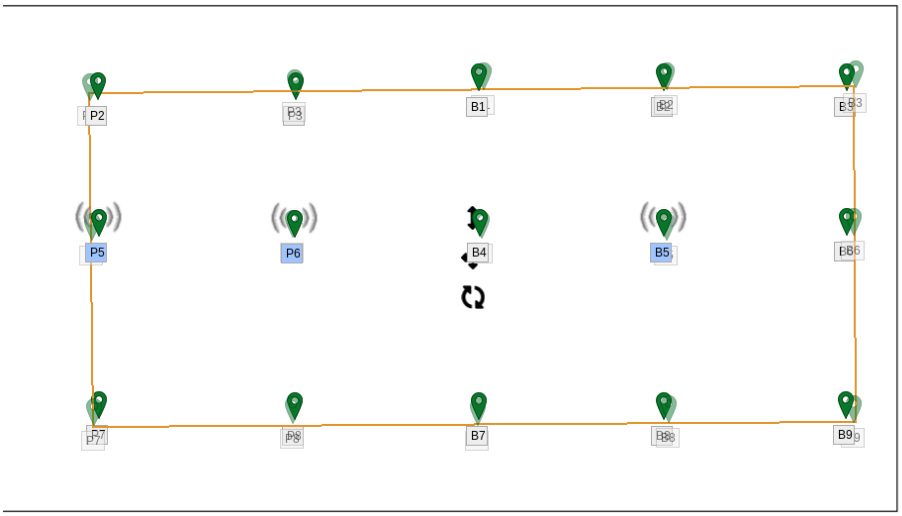
Obrázek 7.7: Výsledek Auto-Deploymentu výpočtu 1 iterací algoritmu.

V druhém testu byly použity restrikce, které rozdělily prostředí do dvou buněk, první s 25 kotvami a druhá s 15 kotvami. Výsledek Auto-Deploymentu vykázal pro obě rozložení průměrnou odchylku od skutečnosti v hodnotě 0,39 m, což je akceptovatelná hodnota. Z pohledu lokalizace je důležitější faktor celkové správnosti pozic, což znamená, že žádná kotva se nebude výrazně vychylovat od ostatních. Přesně taková kotva je pak z pohledu lokalizace problém. Levá část instalace s 25 kotvami je zobrazena na obrázku 7.8. Pravou část z 15 kotvami lze vidět na obrázku 7.9



Obrázek 7.8: Výsledek levé části rozložení kotev.

Přestože obě výsledná umístění kotev ukazují viditelné chyby, stále však nejsou tak významné, aby porušily i lokalizaci. Ve větších RTLS prostředích s podobně naskládanými kotvami nastává problém s kumulováním TWR chyby. Každé TWR měření totiž vykazuje určitou chybu, která může být mezi jednou dvojicí kotev například 5 cm. V celé instalaci se tato chyba však kumuluje a může narůstat až do jednotek metrů. Což je i případ této instalace. V některých situacích může být nutné rozdělit Auto-Deployment do více menších částí, aby výsledné rozložení lépe zapadalo do půdorysu.



Obrázek 7.9: Výsledek pravé části rozložení kotev.

7.4 Limitace a možná vylepšení

Detekce chybných umístění kotev a primárně detekce prohozených kotev je zejména limitována existencí stavů, kdy funkce nedokáže v některých případech přesně určit, které páry jsou prohozené, či nikoli. V takovém případě by funkce neměla takové kotvy označit za prohozené. False negative informace je pro uživatele vhodnější, protože v opačném případě to znamená, že uživatel, pokud by byl informován o falešně prohozených kotvách, učiní kroky na popud systému, které vedou k vytvořeni chyby v instalaci.

Prvním vylepšením tohoto systému by bylo zdokonalit detekci a snížit množství falešných stavů. Tomu by pomohla zpětná kontrola vyhodnocených kotev. Ta by mohla fungovat tak, že v první fázi se aktuálním systémem určí kritické kotvy a v další fázi se systém pokusí vyzkoušet všechny kombinace změn, které vedou ke zlepšení stavu. Pokud takovou kombinaci najde, pak je i nalezená chyba ověřena a uživatel může provést správné změny.

Z hlediska funkce automatického rozmisťování kotev pro 1D lokalizaci, jsou nejvíce limitující manuální kroky, které uživatel musí před inicializací provést. Například automatická detekce uliček by výrazně zpříjemnila uživatelskou zkušenost s tímto systémem. Taková funkce by vyžadovala velmi spolehlivý klasifikátor, který by na základě inicializace toto rozhodoval. Podle mého názoru, by pro tuto úlohu byla nejvhodnější dobře natrénovaná neuronová síť, která by dokázala z výsledku TWR a inicializace najít drobné nuance, které by tuto informaci potvrdily. Systém pro nastavování rozložení kotev do plánu by potřeboval také jisté GUI vylepšení, které by umožnilo uživateli umístit jednu kotvu do plánu a dále manipulovat pouze s druhou kotvou tak, aby uživatelsky zpříjemnil a hlavně zpřesnil proces vkládání uliček do plánu.

Varianta funkce pro automatické rozmístění kotev pro 2D je limitovaná podobně jako systém pro 1D. Namísto uliček jsou zde využívány restrikce. Pokud by i v tomto případě bylo možné spolehlivě oddělovat lokalizační buňky, pak by se také zvýšila uživatelská přívětivost aplikace RTLS Studio. Stejně jako v případě 1D uliček i zde by mohl fungovat klasifikátor z neuronové sítě, který by tuto úlohu rozhodoval.

Kapitola 8

Závěr

Tato diplomová práce začínala průzkumem principů určování polohy v senzorových sítích. Část byla zaměřená na porovnání různých bezdrátových technologií používaných pro lokalizaci, tedy zejména Wi-Fi, BLE a UWB. Dále se téma přesunulo na výhody technologie UWB v oblasti lokalizace ve vnitřních prostorech pro pochopení bližších radiových principů dané bezdrátové technologie. Následovala část, která sumarizovala různé algoritmy určování pozice v čele s multilaterací. Rozebrány byly vlastnosti a rozdíly mezi dvěma typy lokalizace TWR a TDOA.

Poté následovalo seznámení se s platformou Sewio RTLS. Funkcionalita pro automatické rozmisťování kotev v RTLS systému byla tvořena ve spolupráci s firmou Sewio, a proto byl tento systém vyvíjen přímo pro tuto lokalizační platformu. Prozkoumány byly všechny elementy této platformy se zaměřením zejména na druhy lokalizace, které tato platforma podporuje. Způsob, jakým se kotvy v systému zaměřují, je přímo spojen s druhem lokalizace, a proto vyvíjené řešení muselo akceptovat tyto požadavky.

Dále bylo provedeno měření reálné instalace platformy Sewio RTLS ve výzkumném centru Intemac. Z tohoto měření byla získána data vzdáleností mezi jednotlivými kotvami změřených pomocí TWR. Na základě těchto vzdáleností byly nalezeny vazby pro filtrování nevhodných dat, které by mohly potencionálně znehodnotit výsledek automatického polohování kotev. Podobné měření proběhlo i v logistickém skladu, který přinesl výsledky TWR z 1D instalace.

Na základě získaných informací o platformě Sewio RTLS, technikách lokalizace a reálných výsledků metody TWR bylo možné analyzovat problém a navrhnout řešení, které by vykonalo potřebnou funkci s požadovaným výsledkem. Z návrhu vyplynula funkce pro detekci chyb zaměření kotev v instalaci, funkce pro automatické zaměřování 1D kotev pomocí TWR a vylepšená implementace zaměřování 2D kotev zvané Auto-Deployment. Tyto navržené funkce byly úspěšně implementovány a integrovány do celého systému.

Výsledky testů těchto řešení potvrdily reálný přínos pro platformu Sewio RTLS, ale také ukázaly na limitace, kterými systém pro automatické rozmisťování kotev pomocí TWR disponuje. Mezi tyto limitace patří zejména malá citlivost a vysoké požadavky na přesnost zaměření kotev pro detekci přehozených kotev. Dále pak kumulace TWR chyby v rozsáhlých instalacích. Potenciálním problémem může být navržený uživatelský postup pro využití vyvíjených funkcí a také nutnost uživatelské interakce pro dodání potřebných informací. Tyto interakce můžou vést k zanesení uživatelské chyby. Budoucí vylepšení tohoto systému by mohlo některé uživatelské vstupy vytvářet automaticky bez nutnosti uživatelského zásahu.

Zkratky

AoA Angle of Arrival. 4, 6, 7

BLE Bluetooth Low Energy. 3, 4, 13, 52

BPM Burst position modulation. 5

BPSK Binary phase-shift keying. 5

GNSS Global Navigation Satellite System. 4

GPS Global Positioning System. 4

IMU Inertial measurement unit. 13

IPS Indoor positioning system. 4

LOS Line of sight. 4, 12, 15, 16, 17, 19, 23, 26, 27, 40

NLOS Non-Line of sight. 23, 26, 32, 33, 38, 47, 48

RF Radio Frequency. 4

RFID Radio-frequency identification. 4

RSSI Received Signal Strength Indication. 4, 5, 6, 8, 14, 18

TDOA Time Difference of Arrival. 4, 8, 9, 11, 12, 13, 16, 21, 31, 34, 52

ToA Time of Arrival. 6, 8

ToF Time of Flight. 6, 9, 19

TWR Two-way Ranging. 9, 11, 15, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52

UWB Ultra-wideband. 3, 4, 5, 6, 11, 12, 14, 15, 16, 21, 22, 23, 52

Literatura

[1] Biswas, I. *Triangulation vs Trilateration vs Multilateration – for Indoor Positioning Systems* [online]. 2019 [cit. 2021-11-15]. Dostupné z: https://www.pathpartnertech.com/triangulation-vs-trilateration-vs-multilateration-for-indoor-positioning-systems/.

[2] Connell, C. *What's The Difference Between Measuring Location By UWB, Wi-Fi, and Bluetooth?* [online]. 2015 [cit. 2021-11-05]. Dostupné z: https://www.electronicdesign.com/technologies/communications/article/21800581/whats-the-difference-between-measuring-location-by-uwb-wifi-and-bluetooth.

[3] Duru, A., Şehirli, E. a Kabalci, I. Ultra-Wideband Positioning System Using TWR and Lateration Methods. In:. červen 2018, s. 1–4. DOI: 10.1145/3234698.3234756.

[4] Hsu, E. *An Overview of the IEEE 802.15.4 HRP UWB Standard* [online]. 2021 [cit. 2021-11-05]. Dostupné z: https://blogs.keysight.com/blogs/tech/rfmw.entry.html/2021/07/28/an_overview_of_ieee-J7ac.html.

[5] Mickael Viot, J. S. et al. *Ultra-Wideband for dummies*. 1. vyd. John Wiley & Sons, Inc., 2021. ISBN 978-1-119-80959-3.

[6] Sewio. *1D Deployment Rules* [online]. [cit. 2022-05-10]. Dostupné z: https://docs.sewio.net/docs/1d-deployment-rules-38535880.html.

[7] Sewio. *Network Architecture and Required Ports* [online]. [cit. 2022-05-10]. Dostupné z: https://docs.sewio.net/docs/network-architecture-and-required-ports-25593406.html.

[8] Sewio. *Radiation Pattern - Vista DirectFive* [online]. [cit. 2021-12-23]. Dostupné z: https://docs.sewio.net/docs/radiation-pattern-vista-directfive-3244597.html.

[9] Sewio. *Tags Overview* [online]. [cit. 2021-12-23]. Dostupné z: https://docs.sewio.net/docs/tags-overview-3244803.html.

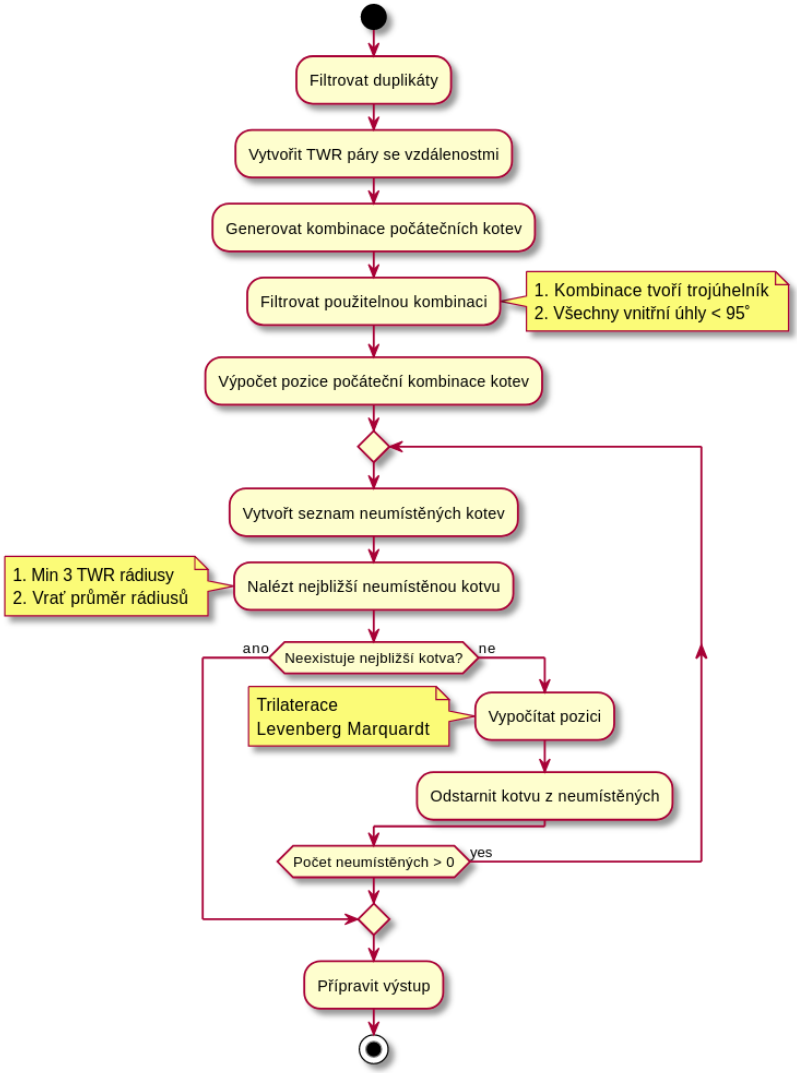
[10] Sewio. *Time Difference of Arrival* [online]. [cit. 2021-11-15]. Dostupné z: https://www.sewio.net/uwb-technology/time-difference-of-arrival/.

[11] Sewio. *Two Way Ranging* [online]. [cit. 2021-11-15]. Dostupné z: https://www.sewio.net/uwb-technology/two-way-ranging/.

[12] Staff, C. C. *Triangulation, Trilateration, or Multilateration?* [online]. 2014 [cit. 2021-11-15]. Dostupné z: https://circuitcellar.com/resources/ee-tips/triangulation-trilateration-or-multilateration-ee-tip-125/.

Příloha A

Auto-Deployment aktivity diagram



Obrázek A.1: Diagram aktivity funkce Auto-Deployment

Příloha B

Obsah přiloženého paměťového média

- text/ — Diplomová práce.
  - src/ — LaTeX zdrojové kódy tohoto dokumentu.
  - xprude02-dip.pdf — PDF soubor s diplomovou prací.
- rtls-studio/ — Kořenový adresář projektu rtls-studio
  - rtls-manager/ — Zdrojové soubory a testy funkce Auto-Deployment
  - rtls-web/ — Zdrojové soubory a testy funkce pro kontrolu rozložení kotev
- tests/ — Zbylé testovací soubory
  - twr-connections/ — Vizualizér TWR vazeb
- README.md — Soubor s popisem obsahu CD