



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék



TDoA algoritmusok összehasonlítása UWB alapú beltéri helymeghatározó rendszerben

TDK DOLGOZAT

Szerző:

Kovács Zoltán Márk

Konzulens:

Matolcsy Balázs

2021

Tartalomjegyzék

1. Bevezetés	4
1.1. Lehetséges technológiák	4
1.2. UWB technológiai áttekintés	6
1.3. A pozíció meghatározása	9
1.3.1. ToA (Time of Arrival)	10
1.3.2. TDoA (Time Difference of Arrival)	13
1.4. A pozicionálás pontosságát befolyásoló jelenségek	13
1.5. A technológia előnyei és hátrányai	15
2. TDoA algoritmusok	17
2.1. Legkisebb négyzetek módszere	18
2.2. Foy módszere	20
2.3. Fang módszere	22
2.4. Friedlander módszere	25
3. Összehasonlító vizsgálat	27
3.1. Algoritmusok működési tulajdonságai	27
3.2. Szimulációs eredmények	28
4. Összefoglalás	36

Kivonat

Az ultraszélessávú (Ultra-Wideband, továbbiakban UWB) technológia hatékonyan alkalmazható beltéri helymeghatározás esetén. Ez a beltéri helymeghatározás pontossággal szembeni igényére vezethető vissza, amely igényt a globális helymeghatározó rendszerek általában nem tudnak kellő megbízhatósággal kielégíteni. Ezzel szemben az UWB technológián alapuló módszerekkel képesek vagyunk 5...15 cm pontossággal beltéri pozíciót becsülni, ráadásul ezt adott esetben alacsony késleltetéssel. UWB esetén ezek a módszerek alapvetően jelterjedési idő, illetve beérkezési időkülönbség alapján becsülik a keresett eszköz pozícióját. A TDK dolgozatom témáját képezi a beérkezési időkülönbségből pozíciót becsülő módszerek analízise és összehasonlítása. Az egyes módszerek összehasonlításának főbb szempontjai a pontosság és hibatűrés. Ezen szempontok mutatják meg a különböző hibakomponensek ráhatását a becsült pozícióra, illetve jelenítik meg a módszerek előnyeit és hátrányait. A kapott eredményeket szimulációs környezetben támasztom alá MATLAB segítségével.

Abstract

Ultra-Wideband (UWB) technology can be used effectively for indoor positioning. This is due to the need for indoor positioning accuracy, which global positioning systems are generally unable to meet with sufficient reliability. In contrast, with methods based on UWB technology, we are able to estimate the indoor position with an accuracy of 5...15 cm, and in addition, this may be done with low delay. In the case of UWB, these methods basically estimate the position of the searched device based on the time of arrival (ToA) and the time difference of arrival (TDoA). The topic of my TDK dissertation is the analysis and comparison of methods for estimating position from the time difference of arrival. The main aspects show the effect of different error components on the estimated position and the advantages and disadvantages of the methods. I support the obtained results in a simulation environment using MATLAB.

1. fejezet

Bevezetés

Napjainkra a helyalapú szolgáltatások iránti igény kimagaslóvá vált. Ezen szolgáltatások több kategóriában játszhatnak központi szerepet. Például a mindennapok során az okos otthon, személyre szabott kirakatok, forgalom figyelő rendszerek említhetők meg, mint hely alapú szolgáltatás. Azonban nemcsak a mindennapok során, hanem ipari felhasználásban is szót ejthetünk ilyen szolgáltatásokról. Az Ipar 4.0 fejlődésével ipari létesítményekben elengedhetetlen a beltéri pozicionálás, mely egyrészt az autonóm robotok közlekedésének alapvető feltétele, másrészt a pozíció ismerete alapján lehetőség van a logisztikai feladatok lefutási idejének optimalizálására [1]. A hely alapú szolgáltatások technológia szintjén 4 területből épülnek fel, úgy mint mobil eszközök (lényegében miattuk lett igény a szolgáltatásra), kommunikációs hálózat (mely biztosítja a szereplők közti kommunikációs csatornákat), GIS (Geographical Information System, téradatbázis, mely helyfüggő adatokat nyújt) és pozicionálási technológia. Utóbbi területhez fűződik a TDK dolgozatom. A pozicionálásra különböző feltételek mellett számos technológia alkalmas. TDK dolgozatomban az imént említett feltételeket a következők alkotják: pontosság (minimum 5...10 cm), beltéri környezet, legalább 10 Hz-es pozíciófrissítés. Ezen feltételeket az ipari felhasználás teszi kiemelten fontossá. Robotok vagy drónok épületen belüli közlekedésénél elengedhetetlen a pozíciójuk pontos ismerete. Ipari környezetben jellemzően épületen belül kerülnek elhelyezésre a különböző berendezések, innen ered a beltéri környezet feltétel. A pozíciófrissítés igénye pedig a navigálhatóságot segíti, ugyanis a navigációs algoritmus hatékonyabb, ha időben és térben elegendően sűrű a mintavételezett pozíciósorozat mely rendelkezésre áll.

1.1. Lehetséges technológiák

A következőkben számba veszem jelenleg elérhető technológiákat, amelyekkel képesek vagyunk pozíciót meghatározni.

Elterjedt megoldás a GPS (Global Positioning System), GPS esetén a kezdeti

szinkronizációs idő nagyságrendje az egyik probléma, de arról sem szabad megfeledkezni, hogy a szinkron beállása után is a polgári felhasználás esetén a pontosság a méter nagyságrendjébe esik, tehát önmagában alkalmatlan kis (pár centis) megmozdulások észlelésére. GPS esetén egy további probléma, hogy beltérben a jelentős jelerősség csökkenés miatt alapvetően leromlik a pontosság. Viszont ehhez még hozzájön a beltéri többutas terjedés hatása, amely a gyakorlatban használhatatlanná teszi a GPS-t beltérben [2] [3].

Egy másik lehetőség a mobil hálózat használata helymeghatározásra. A problémát ez esetben az okozza, hogy alapvetően nem ez technológia célja, hanem a kommunikáció létrehozása, illetve adatátvitel. A mobil hálózat problémája a helymeghatározást tekintve az, hogy főként csak cella/szektor szintű pozicionálásra alkalmas. Emiatt csak nagyon kis méretű cellákkal lefedett területen lehet önmagában helymeghatározásra használni, általánosságban azonban kis mozgások érzékelésére alkalmatlan. Más tekintetben a cellaszintű meghatározás kifejezetten előnyös, ugyanis ilyen szinten rendkívül gyorsan megtalálható egy mobil készülék. A kis- és középvállalatok néhányszor 10 gépjárműből álló flottát tartanak fent. Ez esetben elegendő, ha a gépjárművek pozícióját cellaszinten ismerik, ezáltal hatékonyabbá tudják tenni a logisztikai folyamataikat [4].

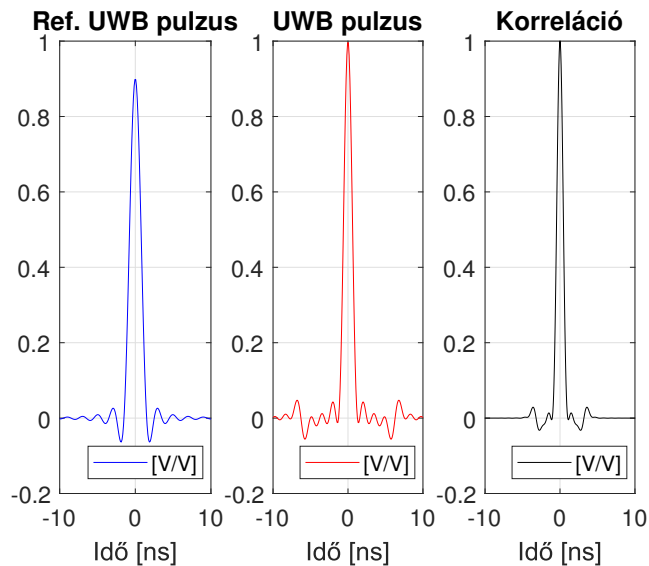
A Wi-Fi (Wireless Fidelity) is felhasználható helymeghatározásra [5]. Általánosan a vett jel erősségét (RSSI-Received Signal Strength Indicator) veszi alapul. A módszer előkészülete, hogy nagy méretű táblázatban eltárolja az egyes pozíciókhoz tartozó RSSI értéket, majd a pozicionálás úgy történik, hogy a mért RSSI értéket összeveti a táblázattal, és a mért értékhez legközelebb eső táblázati rekord tartalmazza azt a pozíciót, amely a legvalószínűbb (ez az úgynevezett "fingerprinting" technika). Ennek a módszernek az a hátránya, hogy a környezeti változásokra érzékeny, a táblázat értékeit frissíteni kell, ha a rekordok felvételének idejéhez képest a belső tér objektumainak helyzete változott. Beltérben ez a hátrány jelentős is lehet, ha nagy objektumok folyamatosan mozognak, mert ez folyamatosan azt igényelné, hogy minden egyes elmozdulás előtt újra fel kellene térképezni a területet, így frissítve a táblázatunkat. Akkor célszerű használni ezt a módszert, ha követelmény az alacsony beruházási költség, mivel a Wi-Fi hálózat elterjedtsége épületeken belül szignifikáns, viszont a pontossággal szemben nem támaszt nagy igényt a megrendelő. A módszer nem csak beltérben üzemképes, hanem kültéren is, viszont nem nagy távolságban. Ez főként azért alakult így mert a kültéri Wi-Fi lefedettség inkább csak konkrét üzlethelységeket, vendéglátáshoz fűződő teraszok esetén van jelen és nem egy összefüggő kiterjedt hálózatként, így inkább mobilhálózatra csatlakoznak a mobilkészülékek.

A Bluetooth alapvetően a Wi-Fi-hez hasonlóan adatátvitelre lett kifejlesztve, de szintén alkalmas RSSI alapon történő helymeghatározásra [6] [7]. A Bluetooth alkalmazásának előnye, hogy az épületek falain jelentős csillapítást szenved, mivel kis

kitöltési tényezőt használ, így a jel erőssége elhanyagolhatóvá válik [8]. Ez előnyös, mert így a Wi-Fi-vel ellentétben az egyes helységekből kialakuló jelerősségek eloszlása kevésbé lesz egyenletes, így a jelerősség mintákat tartalmazó táblázat sokkal nagyobb dinamikával fog rendelkezni, így kisebb a valószínűsége annak, hogy két területhez azonos RSSI fog tartozni. A látható fényű kommunikációt (VLC-Visible Light Communication), infravöröst, és RFID-t (Radio-frequency identification) használó módszereket a TDK dolgozat szempontjából egy kategóriába sorolom. Ilyen tekintetben hátrányuk, hogy nagy pontosságú helymeghatározásra nem használhatók, ezzel szemben nagyterületű beltérben (pl.: áruházban) statisztikus adatgyűjtést tekintve nagy potenciál van ezekben a technológiákban. Egyetlen kivételt képez, az RFID "szőnyeg", ekkor akár pár centiméterenként telepíthetők a padlózat felszíne alatt RFID kártyák. Ezeknek a jelét egy talajon mozgó robot képes érzékelni, és tovább küldeni akár Wi-Fi-t, akár Bluetooth-t használva a vett jeleket, így egy központi adatbázis alapján a robot orientációja, és pozíciója is azonosítható. Természetesen ez akkor működik, ha a keresett objektum csak 2 dimenzióban végez haladó mozgást, és ez a sík közel van a kitelepitett RFID szőnyeghez. A felsorolt technológiák és módszerek széles körben nem alkalmasak az összes említett feltételt maradéktalanul kielégíteni. Azonban létezik egy technológia amely igen, ez pedig az UWB (Ultra-Wideband). A következőkben röviden bemutatásra kerül a technológia, illetve azok a módszerek amelyek UWB alapon realizálhatók.

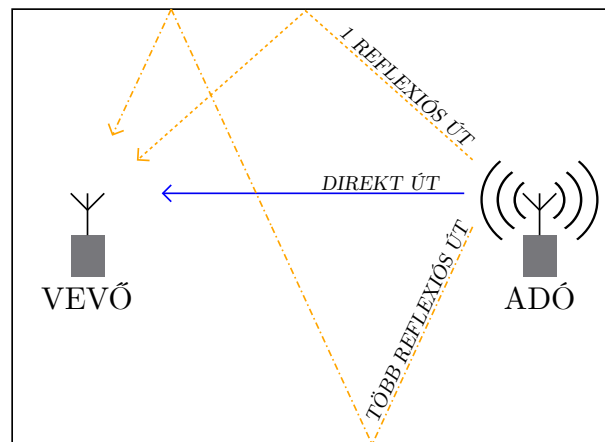
1.2. UWB technológiai áttekintés

Az UWB-t használó eszközök szabályozása különleges esetként fogalmazható meg. Ennek oka az, hogy az UWB technológia által lefedett spektrum átlapolódik olyan frekvenciasávokkal, amelyeket valójában más (jellemzően keskeny sávú) szolgáltatók is használnak. Tekintettel arra, hogy számos üzemeltető hatalmas összegeket fizetett a frekvenciahasználatért érthetővé válik, hogy az UWB szabályozási folyamata érdekellentéteket von maga után. Az UWB eszközökre vonatkozólag először az FCC (Federal Communications Commission - Amerikai Szövetségi Távközlési Bizottság) tett rádiószabályozási követelményeket, ezzel lehetővé téve a kis hatótávolságú UWB technológiát alkalmazó termékek forgalmazását, és külön engedély nélküli működését [9]. Ezen dokumentum definiál először széles körben alkalmazandó specifikációkat, amelyek alapján egy jel UWB kategóriába sorolható. Definíció szerint az UWB jel legalább 20 %-os relatív sáv szélességű (az elfoglalt sáv szélesség a sáv középi frekvenciának legalább 20 %-a), vagy sáv szélessége nagyobb mint 500 MHz. Az IEEE 802.15.4 szabvány rögzíti az UWB jelek időtartománybeli tulajdonságait, tehát ezeknek kell megfelelni a szabvány szerinti működés érdekében. Az 1.1 ábrán látható az elemi jel definíció szerinti alakja (ábrán: Ref. UWB pulzus), középen pedig egy kompatibilis UWB jel, jobb oldalt pedig a kettő közti korreláció. Pontosan rögzítve



1.1. ábra. Elemi UWB jellel való korreláció kimutatása [10]

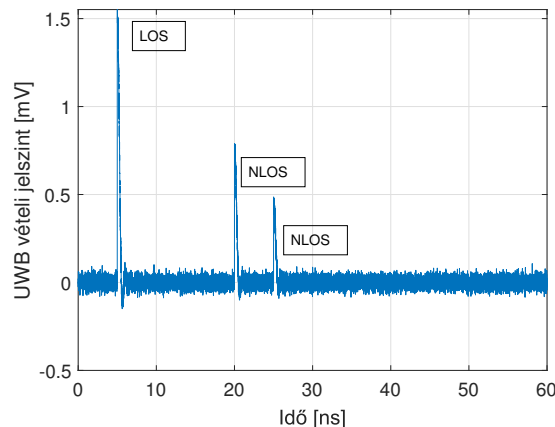
van, hogy a korrelációnak milyen mértékűnek kell lenni, hogy kompatibilis legyen a szabvánnyal [11]. A szabályozás alapján látható, hogy az UWB egy olyan rádiós technológia, mely időtartományban keskeny pulzusokat használ adattovábbításra. Az időben keskeny pulzusok spektrális szélessége szignifikáns, innen eredeztethető a technológia elnevezése. Az 1.2 ábrán látható az, ahogy beltérben nem csak egy



1.2. ábra. Többutas terjedés beltérben

közvetlen (direkt) útvonalon érzékelik meg az adó által kisugárzott pulzus, hanem vannak reflexiós útvonalak is, amelyeken keresztül a pulzus terjedni tud. Először minden esetben a direkt útvonalon érkező pulzust detektálja a vevő, majd egyre kisebb amplitúdóval és egyre nagyobb késleltetésekkel a nem közvetlen úton érkező komponenseket. Ez egy egyszerűsítő ábra, a valóságban akár nagyságrendekkel több reflektált komponens is eljuthat a vevőbe, de nagy részük már oly mértékben csillapodik, hogy ezeket elhanyagolhatjuk. Az UWB technológia nagy előnye a többi technológiához képest, hogy az időben rendkívül keskeny pulzusoknak köszönhetően fokozott tűréssel rendelkezik a többutas terjedés hatása ellen. Ennek

a túrésnek a határát akkor érjük el, ha két pulzus olyan gyorsan érkezik be egymás után, hogy a vevő autokorrelációs algoritmus már képtelen különbséget tenni köztük. Ez a gyakorlatban gyártótól függ, jellemzően néhányszor 10. . . 100 ps nagyságrendbe esik. Az 1.3 ábrán látható, hogy a közvetlen rálátásból érkező pulzustól jól elkülöníthető a reflektált pulzusok érkezési időpontja, azaz a többutas terjedés nem okoz számottevő káros hatást.



1.3. ábra. UWB jel vételi időfüggvénye [12]

Az UWB pulzusok időbeli szélessége (és a vevőben alkalmazott autokorrelátor) alapvetően meghatározza azt, hogy milyen időbeli felbontással vagyunk képesek detektálni a beérkező pulzusokat. Szabadterben az elektromágneses hullámok terjedési sebessége jó közelítéssel fénysebesség. Ha például 1 ns pulzusszélességet tekintünk, akkor mivel ennyi idő elteltével a hullám önmagában 0,3 m-t halad, a távolságbeli felbontóképességnek felső korlátjára legrosszabb esetben 0,3 m választással szintén élhetünk. Ezt a korlátot tovább finomítja a vevőben futó autokorrelátor, így a gyakorlatban 5. . . 10 cm pontosságra lehet számítani.

Az UWB jeleket különböző megoldásokkal állítják elő az egyes gyártók. Ez alapján lehet úgynevezett carrier-less (vivő nélküli) illetve carrier-based (vivőn alapuló) jelről beszélni. Utóbbi esetben a jelgenerálás folyamata az, hogy pulzusokat generálnak, amelyeket utána formálásnak vetnek alá, majd egy lokális oszcillátor által felkeverik a vivő környékére, majd sáváteresztő szűrő után kerül antennával kisugárássra. Előbbi (vivő nélküli) esetben pedig kimarad a lokáloszcillátorral beállított keverő fokozat, és a pulzusokat eleve úgy hozzák létre készletelés útján digitális kapukkal, majd formálják meg, hogy az mint alapsávi jel kisugározható legyen.

Az UWB-vel való kommunikálás esetén fontos, hogy alacsonyan tartsuk a kisugárzott spektrális teljesítmény sűrűséget, ugyanis az UWB spektrálisan átlapolódik más keskenysávú technológiákkal (pl.: Wi-Fi, Bluetooth). Magyarországon az NMHH (Nemzeti Média és Hírközlési Hatóság) honlapján található táblázatok mutatják be a Magyarországon betartandó előírásokat a kisugárzott teljesítményre vonatkozóan. Alkalmazástól függ, hogy adott felhasználó számára melyik tartalmaz-

za a releváns adatokat [13]. A honlapon látható 10.2.2 táblázat alapján például az általános felhasználásra vonatkozó paraméterek olvashatók ki, melyekre akkor van szükség, ha az adott alkalmazásra egyik további speciális adatokat tartalmazó táblázat sem hivatkozik. Ha ezt az első táblázatot vesszük alapul, akkor 6...8,5 GHz-es sávban például 0 dBm (Effective Isotropic Radiated Power - Izotropikusan Sugárzott Ekvivalens Teljesítmény) a megengedett maximális kisugárzott csúcsteljesítmény (50 MHz-re vonatkoztatva). A maximális átlagos spektrális teljesítménysűrűség pedig -41,3 dBm/MHz. Ez utóbbi 1000 MHz sávszélességű jel esetén -11,3 dBm átlagos spektrális teljesítményt jelent. (ábra az átlapolódásról még kell)

A technológia hátrányaként említhető, hogy például a Wi-Fi, vagy Bluetooth-hoz képest kevésbé volt ezidáig elterjedt, azonban az Apple saját fejlesztésű U1 integrált áramköri megoldását beépítette 2019-ben az Iphone 11 készülékbe, majd utána az Iphone 12, Iphone 13 mobilokba, illetve az Apple Tag-be is [14], [15], [16]. Emellett egy másik világszerte ismert márka, a Samsung is beépítette a Samsung Galaxy Note20 Ultra, ZFold2, és S21 készülékekbe, valamint a SmartTag+ eszközbe [17]. Jelenleg az UWB-t arra használják, hogy együttműködve a BLE (Bluetooth Low Energy) technológiával, pontosabb távolságmeghatározást kínáljanak az eszközök, illetve a távolságot felhasználva már meglévő funkciókat fejlesszenek tovább. Ezzel összefüggésben egy feltételezett cél a kiterjesztett valóság (Augmented Reality-AR) elterjesztése a mobil eszközök által [18]. A kiterjesztett valóság pedig nagyban könnyíti bizonyos térbeli tervezési feladatok közben fellépő problémák áthidalását. Ez alapján kijelenthető, hogy az UWB technológiára kíván építeni a jelenleg és a közeljövőben több jelentős világpiaci szereplő, más szóval a területnek aktualitása van.

1.3. A pozíció meghatározása

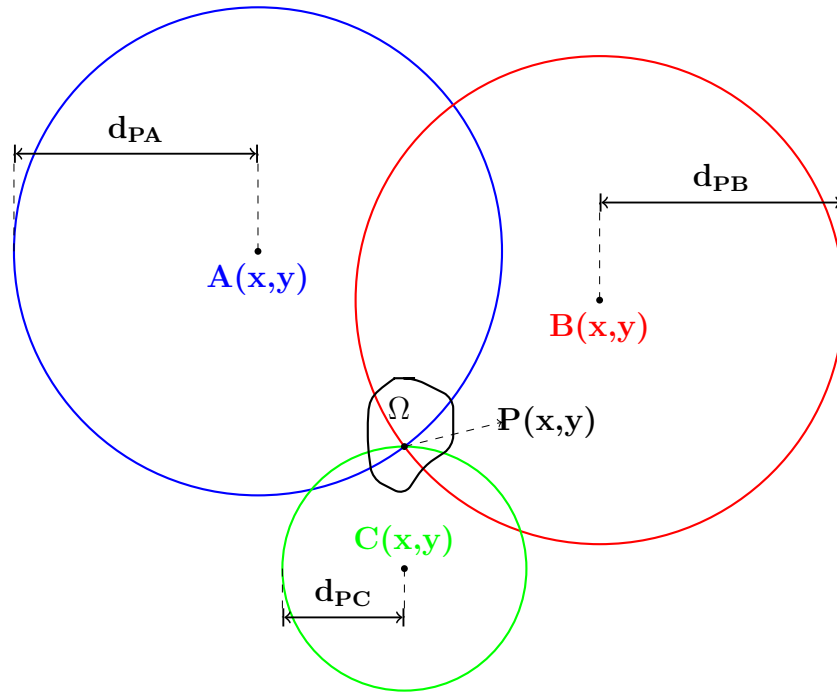
A technológia által alkalmazható módszerek 2 jellemző eszközt különböztetnek meg. Az egyik amelynek a pozícióját meg szeretnénk becsülni (ennek neve tag), a másik pedig az, amely rendelkezésünkre áll a pozicionáláshoz (ennek neve anchor). Helymeghatározás esetén beszélhetünk abszolút illetve relatív pozícióról. Előbbi akkor becsülhető, ha ki van hozzá építve egy infrastruktúra (azaz ismerjük az anchorok pontos helyzetét), utóbbi pedig akkor ha ilyen nem áll rendelkezésre. Utóbbira szemléletes példa, amikor gyors reagálású katonai bevetés zajlik valahol beltéri terepen, és nincs idő infrastruktúrát kiépíteni. Ilyenkor a katonáknál lévő készülékek ad-hoc hálózatot (készülékek közvetítő nélkül kommunikálnak) alakítanak ki, így tudják egymáshoz képesti helyzetüket. Ez azonban egy fix, ismert helyzetű eszköz nélkül csak a bevetés résztvevői számára ad információt. Ha ki van építve egy infrastruktúra, akkor a lokális koordináta-rendszer összefésülhető a globális koordinátákkal. A továbbiakban relatív pozíció számításával nem foglalkozom.

Az alábbiakban bemutatom a gyakran használt UWB alapú, időmérést alkalmazó pozicionálási módszereket. [19]

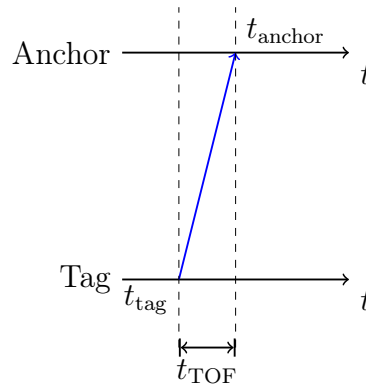
1.3.1. ToA (Time of Arrival)

A Time of Arrival egy időmérésen alapuló távolságmérés, a mért paraméter pedig az eszközök közötti jelterjedési idő. Ha ismerjük a jel terjedési sebességét, akkor kiszámítható a két eszköz közötti távolság. A pozíció meghatározása 3 dimenzióban ez esetben a következő módon történik. Adott egy meghatározandó $P(x, y, z)$ pont, melyet tagnak nevezünk. Kérdés, hogy hány anchor kell $P(x, y, z)$ meghatározásához. 1 anchor (későbbiekben $A(x, y, z)$) esetén mindössze egy d távolságot tudunk meghatározni, amely kirajzol $A(x, y, z)$ körül egy gömböt, és ezen gömb bármely pontján lehet $P(x, y, z)$. Ha $A(x, y, z)$ mellett elhelyezünk egy másik anchor (továbbiakban $B(x, y, z)$), akkor $A(x, y, z)$ és $B(x, y, z)$ által kifeszített gömbök metszeni fogják egymást, a metszet alakja pedig egy esetet leszámítva (ha a $P(x, y, z)$ pont a két anchor között található) mindig egy szabályos kör lesz. Ezen helyezkedik el $P(x, y, z)$, és ez még mindig végtelen sok pontot jelent, tehát újabb anchor kell elhelyezni. Észrevehető, hogy bár a lehetséges pontok száma továbbra is végtelen, mégis érezhetően javul a helyzet ahhoz képest mintha csak $A(x, y, z)$ működne. $A(x, y, z)$, és $B(x, y, z)$ mellé egy további $C(x, y, z)$ anchort helyezünk el. Az előbbi helyzethez képest a két gömb metszetét egy újabb gömb fogja metszeni a térben, amely azzal egyenértékű, mintha egy kört metszenénk egy gömbbel. Ennek eredménye esetünkben 2 metszéspont és az egyikben található $P(x, y, z)$. A korábbi gondolatmenetet folytatva a 2 metszéspont közül úgy tudjuk eldönteni, hogy melyik a helyes, ha felveszünk egy negyedik $D(x, y, z)$ anchor. Következésképpen a pozíció kiszámításához legalább 4 anchor szükséges és ez matematikailag egyértelmű megoldást ad eredményül. Azonban mérési hibák következtében a kiszámított pozíció nem pontos. Eme hibák eredménye az, hogy keresett tag $P(x, y, z)$ az öt körülvevő Ω térrészben van. A cél a mérési hibák csökkentése egy olyan szint alá, amely szint alatt Ω térfogata meghatározott érték alá esik. Az 1.4 ábrán látható a folyamat 2 dimenzióban. A ToF kulcsa az, hogy az anchorok és tagek közötti üzenetek küldési, és fogadási időpontját jegyezzük fel, majd ezek alapján számítjuk a jel terjedési idejét. Több üzenetváltási mód létezik, melyeket felsorolásszerűen mutatok be [20]:

- OWR (One Way Ranging) A legegyszerűbb séma az egyutas távolságmérés, ekkor 1 db üzenetet küld a tag az anchornak, így az 1.5 ábrán látható módon számítható a jelterjedési idő. Egyszerűségéből következik, hogy üzenetváltásokkal ekkor lehet leghamarabb mérési eredményt kapni, illetve leggyorsabb helyzetfrissítést elérni. Azonban a módszer egyszerűsége magában hordozza a fő hátrányát is, melyet magába rejt az 1.5 ábra. Ez az időbeli szinkronizáció a két eszköz között. Szinkronizáció nélkül a módszer használhatatlanná válik.



1.4. ábra. Pozicionálás folyamata 2 dimenzióban

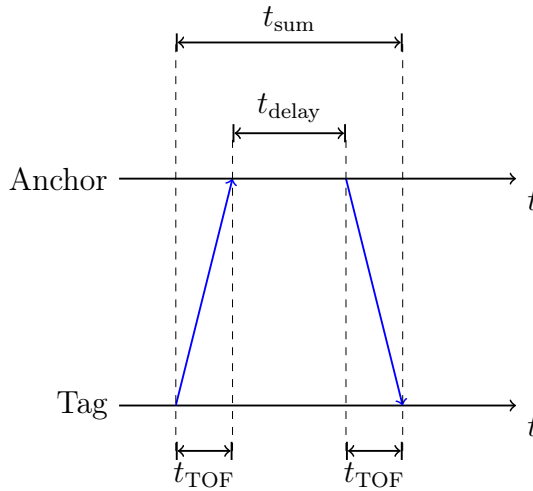


1.5. ábra. Egyutas üzenetváltási séma

$$t_{\text{TOF}} = t_{\text{anchor}} - t_{\text{tag}} \quad (1.1)$$

- TWR (Two Way Ranging) Az OWR szinkronizációs feltételét ki lehet küszöbölni egy plusz üzenet segítségével, melyet az 1.6 ábra reprezentál. Az 1.6 ábrán világosan látható, hogy a jelterjedési időre vonatkozó összefüggésből az abszolút időpillanatok (t_{tag} , t_{anchor}) kiesnek, mindössze a késleltetések kapnak szerepet. Természetesen mindkét eszköz saját órajelének frekvenciája meg kell egyezzen, különben skálatényezőt kell alkalmazni (szoftveres kompenzáció), vagy egy bonyolultabb üzenetváltási sémát kell fontolóra venni, esetlegesen hardveres módosítást kell eszközölni (kristályoszillátor finomhangolása trimmelő kondenzátorral, kisebb frekvenciahibával rendelkező oszcillátor választása). Létezik még a TWR sémának egy módosított változata, amely két

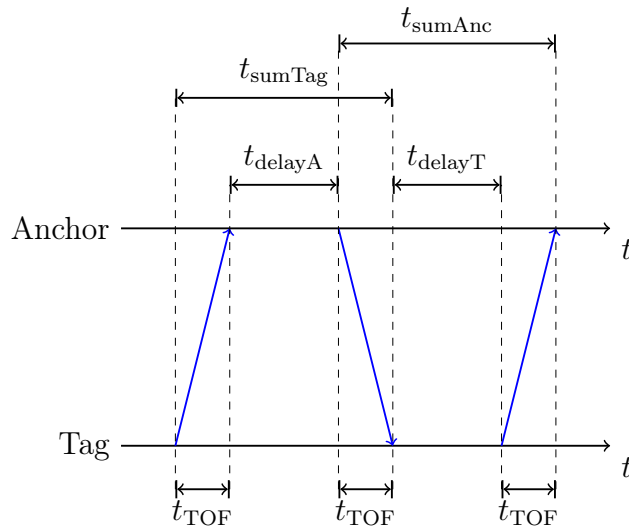
üzenettel kezdődik, és pozitív hatással van a mérés hibájára [21].



1.6. ábra. Kétutas üzenetváltási séma

$$t_{\text{TOF}} = \frac{t_{\text{sum}} - t_{\text{delay}}}{2} \quad (1.2)$$

- SDS-TWR (Symmetrical Double-Sided Two Way Ranging) módszer jó tulajdonsága az, hogy a tag és anchor kristályoszillátor frekvenciakülönbsége által okozott hibát a TWR módszerhez képest jelentősen csökkenti. Mindezt úgy teszi, hogy lényegében összefésül szimmetrikusan 2 TWR sémát, ez pedig a frekvenciahiba okozta jelterjedési idő mérésének hibáját csökkenti nagy mértékben. Az üzenetváltási séma az 1.7 ábrán látható. Hátránya, hogy cserébe ez hosszabb üzenetváltási séma, tehát a helyzetfrissítési gyakoriság jelentősen lecsökken.

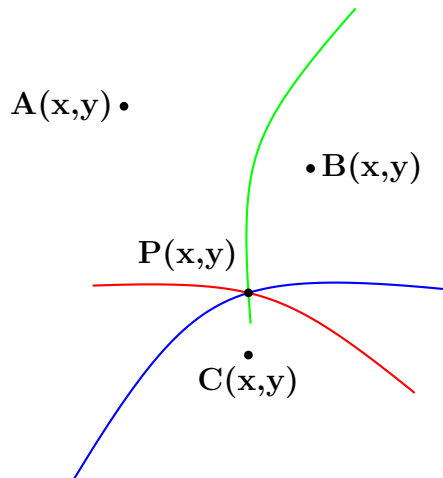


1.7. ábra. Szimmetrikus kétutas üzenetváltási séma

$$t_{\text{TOF}} = \frac{t_{\text{sumTag}} + t_{\text{sumAnc}} - t_{\text{delayA}} + t_{\text{delayT}}}{4} \quad (1.3)$$

1.3.2. TDoA (Time Difference of Arrival)

A módszer alapvetése, hogy az anchorok időben pontosan legyenek szinkronizálva, illetve fix helyre legyenek telepítve. Az eljárás beérkezési időkülönbség mérésén alapul. Ennek szemléletes magyarázata a következő. A tag elküld egy üzenetet minden anchor számára, azok pedig megnézik mikor érkezik meg az üzenet. Válasszunk ki két tetszőleges anchort, és végezzük el a feljegyzett időpontok kivonását. Az így kapott érték több időpont kivonásával létrejöhethet. Ezek a kombinációk 2 dimenzióban kijelölnek a két anchor között egy hiperbolát. A tag helyzete 2 anchor (mint megfigyelők) szempontjából bárhol lehet a hiperbolán. Az eddigiek alapján ha felveszünk (3 dimenziós esetet vizsgálva) még 2 anchort, akkor összesen 6 forgáshiperboloid (ez esetben a hiperbola megoldást adó köpenyének megforgatása) feszül ki a térben, és ezek metszéspontja lesz a tag pozíciója [22]. A szinkronizáció pontosságára rendkívül érzékeny, 120 ps nagyságú hiba már 3 cm pozíció meghatározási hibával járhat [23]. Az 1.8 ábra mutatja be 2 dimenzió esetében a TDoA alkalmazását.



1.8. ábra. Pozicionálás TDoA alkalmazása esetén

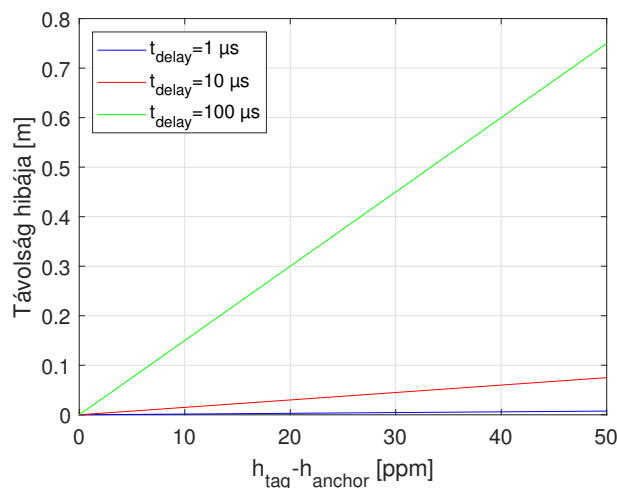
1.4. A pozicionálás pontosságát befolyásoló jelenségek

A bemutatott módszerek időmérésen alapulnak. Az egyes anchorok, és a tag időmérési pontosságát pedig a saját órajelük pontossága határozza meg. A modulokat általában kristályoszillátor működteti, frekvenciájának hibáját ppm-ben (part per million) szokás megadni, és a névleges frekvenciától maximum ekkora hibával tér el.

Amennyiben két anchor órajele különbözik, úgy a mérési eredményben is megjelenik a különbség mértékével kapcsolatban álló hiba. A kapcsolat maga attól függ, hogy milyen módszert alkalmazunk pozicionálásra. TWR esetén levezetés nélkül az 1.4 összefüggéshez jutunk [20]:

$$t_{\text{hiba}} = \frac{1}{2}t_{\text{delay}}(h_{\text{tag}} - h_{\text{anchor}}) \quad (1.4)$$

Ahol t_{delay} az anchor késleltetése, h_{tag} és h_{anchor} pedig a tag és anchor frekvenciájának hibája ppm-ben. Az 1.4 egyenlet alapján készített 1.9 ábrán látható az, hogy néhány konkrét t_{delay} érték mellett mekkora a távolságmérés hibája annak függvényében, hogy mennyire különböznek a ppm-ben mért frekvenciahibák. Jól látható, hogy például ha a hibák különbsége 20 ppm, akkor 100 μs késleltetés esetén a távolságmérés hibája 0,3 m. Ennél nagyságrendekkel jobb eredményt kapunk abban az



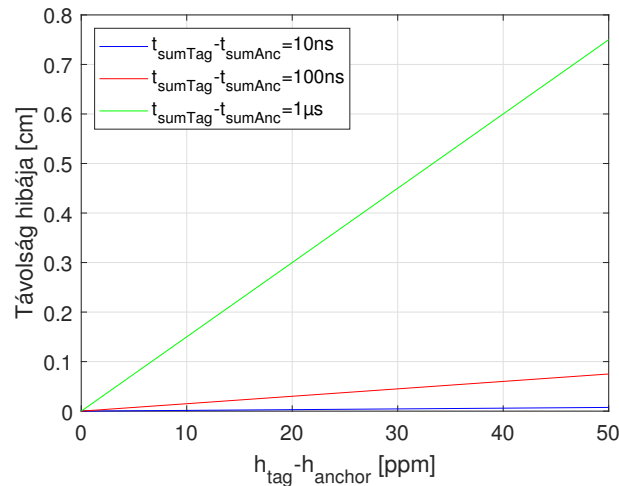
1.9. ábra. Mért távolság hibája a tag és anchor közti frekvenciakülönbség függvényében TWR esetén

esetben, ha SDS-TWR módszert alkalmazunk. Levezetés nélkül ekkor a következő összefüggést kapjuk [20]:

$$t_{\text{hiba}} = \frac{1}{2}(t_{\text{sumTag}} - t_{\text{sumAnc}})(h_{\text{tag}} - h_{\text{anchor}}) \quad (1.5)$$

Ahol h_{tag} és h_{anchor} továbbra a tag és anchor frekvenciájának hibája ppm-ben, t_{sumTag} pedig az az időtartam, amíg a tag várakozik az anchor válaszára. t_{sumAnc} pedig azt az időtartamot jelöli, amíg az anchor várakozik a tag válaszára. Az 1.10 ábra szemlélteti távolságmérés hibáját. Láthatóan sokkal jobb a helyzet, mivel feltételezhetően t_{sumAnc} és t_{sumTag} egymáshoz közel álló értékek.

Az 1.9 és az 1.10 ábrából kivehető, hogy legjobban akkor tudjuk csökkenteni a mérési hibát, ha a frekvencia különbség minimális. Ezt elérhetjük a korábban említett kristályoszillátor finomhangolással, melyet egy trimmelő kondenzátorral



1.10. ábra. Mért távolság hibája a tag és anchor közti frekvenciakülönbség függvényében SDS-TWR esetén

tudunk elérni. A másik megoldás a TCXO (Temperature Compensated Crystal Oscillator - hőmérséklet kompenzált kristályoszillátor). Az ilyen oszcillátorok tartalmaznak egy kis kiegészítő áramkört, amellyel akár 1 ppm alá is vihetjük a frekvenciahibát. Sajnos ez jelentősen megnövelheti a modul árát, így olcsóbb termékekben nem jellemző a TCXO használata. Ha nincs lehetőség TCXO használatára, és a modul belső kristályoszillátora nem trimmelhető, akkor szoftveres úton tudjuk kompenzálni valamelyest a frekvenciahibát.

A terjedési időn kívül az időpont feljegyzéséig eltelik néhány 100 ps, ezt nevezünk antenna késleltetésnek (noha ebbe bele kell érteni az analóg illetve digitális feldolgozási időt is). Ezt a késleltetést szoftveres úton tudjuk kompenzálni oly módon, hogy a mért értékhez hozzáadunk egy konstans, amelyet referenciamérések útján kaphatunk meg.

A közvetlen rálátás (Line Of Sight - LOS) esetében olyan útról beszélünk, ahol a jel reflexió nélkül megérkezik. A jel terjedési sebessége konstans és értéke közelítőleg a fény sebességével egyenlő. Abban az esetben ha ez nem teljesül, akkor beszélünk NLOS-ról (Non Line Of Sight). Ekkor a jel közvetlen úton nem érkezik be a vevőhöz és csak egy másik útvonalon érkező üzenet lesz csak feldolgozható. Az UWB kulcsfontosságú előnye éppen abban rejlik, hogy időtartományban a közvetlen úton érkező pulzustól a nem közvetlen úton érkező pulzusokat meg tudjuk különböztetni, ahogy az 1.3 ábrán látható.

1.5. A technológia előnyei és hátrányai

Az UWB technológia kulcs elemei pozicionálást figyelembe véve a következők:

- Nagy pontosságú helyzetmeghatározás (cm...dm nagyságrend), melynek kö-

szönhetően például akár drónok térbeli útvonalának nyomon követésére nyílik lehetőség, illetve beltéri helymeghatározás esetén a helyalapú szolgáltatások finomítása lehetséges.

- Nem zavar más technológiákat a spektrális átlapolódás (Wi-Fi, Bluetooth, stb.), illetve a zajtűrő képessége is kiemelkedő.
- Az időben keskeny pulzusok következtében a többutas terjedés hatása nem káros a jelfeldolgozásra.
- Kis teljesítményű adók segítségével működtethető a technológia, ezzel hosszú élettartam garantálható mobil eszközök esetén is.
- Az UWB modulok előállítási költsége alapján gazdaságilag kedvező megoldást kínál helyzetmeghatározásra.

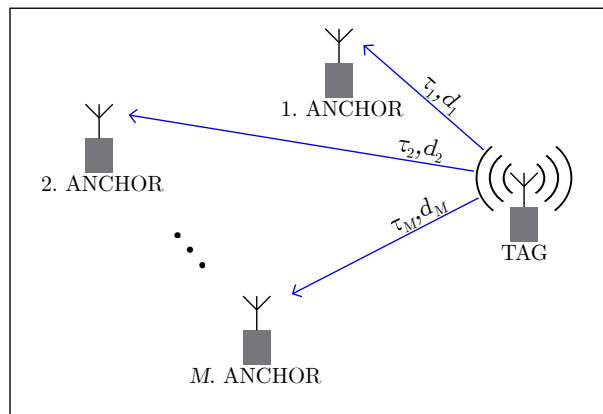
Az UWB technológia kedvezőtlen tulajdonságai pozicionálást figyelembe véve az alábbiak:

- Széles spektrumon történő működés miatt átlapolódik más technológiákkal, ezért szigorú előírások vonatkoznak a kibocsátott maximális teljesítménysűrűségre.
- Léteznek módszerek a közvetlen rálátás hiánya által okozott negatív hatás csökkentésére [24], viszont az UWB közvetlen rálátás esetén hatékony igazán [22]. Ugyanakkor a gyakorlatban ez elmondható minden olyan technológiáról amely nagy pontosságú helymeghatározást kínál.

2. fejezet

TDoA algoritmusok

A következőkben 4 olyan algoritmus kerül bemutatásra, amelyek beérkezési időkülönbség mérésén alapulnak. Mindegyik algoritmus abból indul ki, hogy az anchorok közös órajelet kapnak, ezáltal biztosított a nagy pontosságú szinkronizáció. A TDoA-n alapuló algoritmusok olyan esetekben jönnek számításba, amikor nagy pozíció frissítési sebességre van szükség. Az üzenetváltásokon alapuló módszerekkel (ranging) egy távolságszámítás több ciklusból tevődik össze. Ezzel szemben a TDoA esetén alapvetően a tag üzenetküldésének ismétlési sebessége, illetve az anchorok mögötti adatbázis és feldolgozóegység sebessége a meghatározó. Fontos különbség ezen a ponton az, hogy míg a ranging esetén az egyes késleltetések közvetlenül összeadódnak a teljes számítási folyamatot illetően, a TDoA-t tekintve ez nincs így, mivel amíg a tag újabb üzenetküldésre készíti fel magát, addig az előző kiküldött üzenetet az anchorok éppen feldolgozzák. A tag és az anchor/feldolgozóegység közül az egyik fog dominálni a pozíciófrissítési gyakoriság felső korlátját tekintve.



2.1. ábra. TDoA elrendezés

A 2.1 ábrán látható elrendezés érvényes mind a 4 algoritmus esetén. Az ábrán látható módon az adó szerepét csak és kizárólag a tag, a vevők szerepét pedig az anchorok töltik be. Természetesen több tag is alkalmazható, ebben az esetben minden tagnek külön azonosítóval ellátott üzenete lesz. Ekkor az anchorok meg tudják

állapítani, hogy az egyes beérkezett üzenetek melyik taghez tartoztak. Ez többlet terhelést jelent feldolgozási oldalon, így bizonyos darabszámú tag felett a pozíció-frissítési sebességet előtérbe helyezve mindenképp a feldolgozóegység köthető a felső korláthoz.

A következő néhány egyszerű összefüggésre szükség lesz a továbbiakban:

$$\tau_{1,i} = \tau_i - \tau_1 = \frac{d_i}{c} - \frac{d_1}{c} \quad (2.1)$$

$$d_i = \|\mathbf{p}_i - \mathbf{p}_T\| \quad (2.2)$$

$$d_i = d_1 + c \cdot \tau_{1,i} = d_1 + d_{1,i} \quad (2.3)$$

Ahol:

- τ_i a tag és az i -edik anchor közötti jelterjedési idő
- $\tau_{1,i}$ jelterjedési időkülönbség, amely reprezentálja azt, hogy az 1 anchorhoz képest mennyivel több idő alatt jut el a jel az i -edik anchorhoz
- $d_{1,i}$ a $\tau_{1,i}$ -hez tartozó távolságkülönbség
- c a fénysebesség vákuumban
- d_i tag és az i -edik anchor közötti távolság
- \mathbf{p}_i az i -edik anchor pozíciója 3 dimenzióban, amely felbontva: $[x_i, y_i, z_i]^T$
- \mathbf{p}_T a tag pozíciója 3 dimenzióban, amely felbontva: $[x_T, y_T, z_T]^T$

2.1. Legkisebb négyzetek módszere

Első bemutatott módszer kiindulása a 2.4 egyenlet, amely kifejezi az i -edik anchor és a tag közti távolságot azok koordinátáinak segítségével:

$$d_i = \sqrt{(x_i - x_T)^2 + (y_i - y_T)^2 + (z_i - z_T)^2} \quad i = 2 \dots M \quad (2.4)$$

Második lépésben az előzőhöz hasonlóan az első anchor távolsága is felírható a taghez képest, amelynek később kitüntetett szerepe lesz az algoritmusban, ahol kiszámolandó ismeretlenként fog szerepelni.

$$d_1 = \sqrt{(x_1 - x_T)^2 + (y_1 - y_T)^2 + (z_1 - z_T)^2} \quad (2.5)$$

Felhasználva a 2.3 egyenlet négyzetét jutunk a következő egyenletre, amelyben már közvetlenül látható a mérési eredmények megjelenése:

$$(x_i - x_T)^2 + (y_i - y_T)^2 + (z_i - z_T)^2 = d_1^2 + 2d_1c\tau_{1,i} + c^2 \cdot \tau_{1,i}^2 \quad i = 2 \dots M \quad (2.6)$$

A 2.6 egyenletből algebrai azonosságot felhasználva jutunk a 2.7 egyenlethez:

$$\begin{aligned} x_i^2 - 2x_ix_T + x_T^2 + y_i^2 - 2y_iy_T + y_T^2 + z_i^2 - 2z_iz_T + z_T^2 &= \\ &= d_1^2 + 2d_1c\tau_{1,i} + c^2 \cdot \tau_{1,i}^2 \quad i = 2 \dots M \end{aligned} \quad (2.7)$$

A 2.7 egyenlet már majdnem olyan alakú, amely megoldáshoz vezet, mindössze a tag koordinátáinak négyzeteit kell kiküszöbölni, amelynek ára egy új ismeretlen bevezetése lesz a tag koordinátái mellett. Ez pedig a korábban említett d_1 távolság:

$$d_1^2 = x_1^2 - 2x_1x_T + x_T^2 + y_1^2 - 2y_1y_T + y_T^2 + z_1^2 - 2z_1z_T + z_T^2 \quad (2.8)$$

Kivonva a 2.7-ből a 2.8 egyenletet, és rendezve azt, jól láthatóan a keresett tag koordinátái már csak elsőfokon szerepelnek, illetve a d_1 -en kívül minden más paraméter ismert, vagy mérési eredményként adódik:

$$\begin{aligned} 2(x_1 - x_i)x_T + 2(y_1 - y_i)y_T + 2(z_1 - z_i)z_T - 2d_1c\tau_{1,i} &= \\ &= x_1^2 - x_i^2 + y_1^2 - y_i^2 + z_1^2 - z_i^2 + c^2\tau_{1,i}^2 \quad i = 2 \dots M \end{aligned} \quad (2.9)$$

Mátrixos alakban:

$$\begin{aligned} \underbrace{\begin{bmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 & -c\tau_{12} \\ x_1 - x_3 & y_1 - y_3 & z_1 - z_3 & -c\tau_{13} \\ \vdots & \vdots & \vdots & \vdots \\ x_1 - x_M & y_1 - y_M & z_1 - z_M & -c\tau_{1M} \end{bmatrix}}_{\underline{A}} \cdot \underbrace{\begin{bmatrix} x_T \\ y_T \\ z_T \\ d_1 \end{bmatrix}}_{\underline{X}} &= \\ &= \frac{1}{2} \cdot \underbrace{\begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2 + c^2\tau_{1,2}^2 \\ x_1^2 - x_3^2 + y_1^2 - y_3^2 + z_1^2 - z_3^2 + c^2\tau_{1,3}^2 \\ \vdots \\ x_1^2 - x_M^2 + y_1^2 - y_M^2 + z_1^2 - z_M^2 + c^2\tau_{1,M}^2 \end{bmatrix}}_{\underline{B}} \end{aligned} \quad (2.10)$$

Egy lineáris, inhomogén egyenletrendszer (vagy mátrixegyenlet) pontosan akkor oldható meg, ha \underline{B} benne van az \underline{A} együtthatómátrix oszlopterében. A gyakorlati eseteket tekintve azonban rendkívül gyakran előfordul, hogy a teljes egészében mérések

által meghatározott a 2.10 mátrixegyenlet inkonzisztens, azaz ellentmondásos. Ilyen esetben egy működőképes megoldás, ha kiválasztjuk \underline{A} oszlopteréből a \underline{B} -t legjobban közelítő vektort, és azzal oldjuk meg a mátrixegyenletet. Az ilyen megoldásokat az eredeti egyenletrendszer optimális megoldásainak, vagy a legkisebb négyzetek elve szerinti megoldásainak nevezzük:

$$\hat{\underline{X}} = \underline{A}^+ \cdot \underline{B} \quad (2.11)$$

2.2. Foy módszere

Wade H. Foy publikálta először a Taylor-soros közelítő módszert TDoA pozícióbecsléshez 1976-ban [25]. A módszer azon alapszik, hogy linearizálja elsőrendű Taylor-sorával a nemlineáris távolságkülönbség függvényt (2.12), amelyet a 2.1 egyenlet átrendezésével írhatunk fel.

$$d_i - d_1 = c\tau_{1,i} = \|\mathbf{p}_i - \mathbf{p}_T\| - \|\mathbf{p}_1 - \mathbf{p}_T\| = f_i(\mathbf{p}_T) \quad i = 2 \dots M \quad (2.12)$$

Ha a 2.4 és a 2.5 egyenleteket behelyettesítjük a 2.12 összefüggésbe, akkor jutunk a linearizálandó függvény konkrét alakjához.

$$f_i(\mathbf{p}_T) = \sqrt{(x_i - x_T)^2 + (y_i - y_T)^2 + (z_i - z_T)^2} - \sqrt{(x_1 - x_T)^2 + (y_1 - y_T)^2 + (z_1 - z_T)^2} \quad i = 2 \dots M \quad (2.13)$$

Egy $f(x)$ függvény n -ed rendű Taylor-soros közelítése a pont körül a következőképpen írható fel:

$$f(x) = f(a) + \frac{f'(a)}{1!} \cdot (x - a)^1 + \frac{f''(a)}{2!} \cdot (x - a)^2 + \dots \dots + \frac{f^{(n)}(a)}{n!} \cdot (x - a)^n \quad n \rightarrow \infty \quad (2.14)$$

Felhasználva a 2.13, és a 2.14 egyenlet, és elsőrendű közelítést alkalmazva jutunk a 2.15 közelítésre. A közelítés a tag 0. úgynevezett kezdeti pozíciója környezetében történik. Ez egyben az algoritmus egyik hátrányos tulajdonsága, miszerint egy alkalmazott rendszerben ismerni kell a tag kezdeti pozícióját.

$$f_i(\mathbf{p}_T) \cong f_i(\mathbf{p}_{T,0}) + \frac{f'_i(\mathbf{p}_{T,0})}{1!} \cdot \underbrace{(\mathbf{p}_T - \mathbf{p}_{T,0})}_{\Delta \mathbf{p}_T} \quad (2.15)$$

A 2.15 közelítés átrendezésével, és a tag koordinátás alakjára való áttéréssel kapjuk a 2.16 egyenletet:

$$\begin{aligned}
f_i(\mathbf{p}_T) - f_i(\mathbf{p}_{T,0}) &\cong \frac{\partial f_i(x_{T,0})}{\partial x_T} \cdot (x_T - x_{T,0}) + \\
&+ \frac{\partial f_i(y_{T,0})}{\partial y_T} \cdot (y_T - y_{T,0}) + \\
&+ \frac{\partial f_i(z_{T,0})}{\partial z_T} \cdot (z_T - z_{T,0}) \quad i = 2 \dots M
\end{aligned} \tag{2.16}$$

A 2.16 egyenletet vektor mátrixos alakban felírva jutunk a 2.17 egyenletre, amelyben \underline{C} együtthatómátrix koordinátánkénti parciális deriváltakból áll (ismertebb nevén Jacobi-mátrix), a \underline{D} pedig a mérésekből ismert és adottnak feltételezett függvényérték különbségekből álló vektor.

$$\underbrace{\begin{bmatrix} \frac{\partial f_2(x_{T,0})}{\partial x_T} & \frac{\partial f_2(y_{T,0})}{\partial y_T} & \frac{\partial f_2(z_{T,0})}{\partial z_T} \\ \frac{\partial f_3(x_{T,0})}{\partial x_T} & \frac{\partial f_3(y_{T,0})}{\partial y_T} & \frac{\partial f_3(z_{T,0})}{\partial z_T} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_M(x_{T,0})}{\partial x_T} & \frac{\partial f_M(y_{T,0})}{\partial y_T} & \frac{\partial f_M(z_{T,0})}{\partial z_T} \end{bmatrix}}_{\underline{C}} \cdot \underbrace{\begin{bmatrix} \Delta x_T \\ \Delta y_T \\ \Delta z_T \end{bmatrix}}_{\Delta \mathbf{p}_T} \cong \underbrace{\begin{bmatrix} f_2(\mathbf{p}_T) - f_2(\mathbf{p}_{T,0}) \\ f_3(\mathbf{p}_T) - f_3(\mathbf{p}_{T,0}) \\ \vdots \\ f_M(\mathbf{p}_T) - f_M(\mathbf{p}_{T,0}) \end{bmatrix}}_{\underline{D}} \tag{2.17}$$

Lényegében ebben a formában már egy lineáris egyenletrendszerként kezelhető (eltekintve attól, hogy valójában közelítésen alapszik), amelynek megoldásával egy tag általi elmozdulás vektort ($\Delta \mathbf{p}_T$) kapunk eredményül.

A 2.17 egyenletben szereplő \underline{C} együtthatómátrix elemeinek számítása a következő:

$$\begin{aligned}
\left. \frac{\partial f_i}{\partial x_T} \right|_{\substack{x_T=x_{T,0} \\ y_T=y_{T,0} \\ z_T=z_{T,0}}} &= \frac{\partial}{\partial x_T} \cdot \left[\sqrt{(x_i - x_T)^2 + (y_i - y_T)^2 + (z_i - z_T)^2} - \right. \\
&\quad \left. + \sqrt{(x_1 - x_T)^2 + (y_1 - y_T)^2 + (z_1 - z_T)^2} \right] \Big|_{\substack{x_T=x_{T,0} \\ y_T=y_{T,0} \\ z_T=z_{T,0}}} = \dots \\
\dots &= - \frac{x_i - x_{T,0}}{\sqrt{(x_i - x_{T,0})^2 + (y_i - y_{T,0})^2 + (z_i - z_{T,0})^2}} + \\
&\quad + \frac{x_1 - x_{T,0}}{\sqrt{(x_1 - x_{T,0})^2 + (y_1 - y_{T,0})^2 + (z_1 - z_{T,0})^2}} \tag{2.18}
\end{aligned}$$

Az f_i függvény a tag koordinátáira nézve szimmetrikus, így a másik két koordináta

szerinti parciális deriválás hasonlóan alakul:

$$\left. \frac{\partial f_i}{\partial y_T} \right|_{\substack{x_T=x_{T,0} \\ y_T=y_{T,0} \\ z_T=z_{T,0}}} = - \frac{y_i - y_{T,0}}{\sqrt{(x_i - x_{T,0})^2 + (y_i - y_{T,0})^2 + (z_i - z_{T,0})^2}} + \frac{y_1 - y_{T,0}}{\sqrt{(x_1 - x_{T,0})^2 + (y_1 - y_{T,0})^2 + (z_1 - z_{T,0})^2}} \quad (2.19)$$

$$\left. \frac{\partial f_i}{\partial z_T} \right|_{\substack{x_T=x_{T,0} \\ y_T=y_{T,0} \\ z_T=z_{T,0}}} = - \frac{z_i - z_{T,0}}{\sqrt{(x_i - x_{T,0})^2 + (y_i - y_{T,0})^2 + (z_i - z_{T,0})^2}} + \frac{z_1 - z_{T,0}}{\sqrt{(x_1 - x_{T,0})^2 + (y_1 - y_{T,0})^2 + (z_1 - z_{T,0})^2}} \quad (2.20)$$

A 2.17 mátrixegyenlet megoldását a legkisebb négyzetek elve szerint adja az alábbi egyenlet:

$$\Delta \hat{\mathbf{p}}_T \cong \underline{\underline{C}}^+ \cdot \underline{D} \quad (2.21)$$

Az ismeretlen tag pozíció a következő formában fogalmazható meg:

$$\hat{\mathbf{p}}_T = \mathbf{p}_{T,0} + \Delta \hat{\mathbf{p}}_T \quad (2.22)$$

Itt fontos megjegyezni, hogy a 2.22 egyenlet az algoritmusban egy frissítési egyenletként fogható fel, azaz az i -edik iterációban kiszámolt \mathbf{p}_T vektor a következő $(i + 1)$ -edik iterációban $\mathbf{p}_{T,0}$ szerepét tölti majd be, és ezzel egyúttal az $\underline{\underline{C}}$ együttműködőmátrixot, és a mérési eredményként kapott \underline{D} vektort is frissíteni kell. Emögött az a szemléletes kép áll, miszerint egy függvényt a pontbeli Taylor-soros közelítése mindössze az a pont környezetében írja le meghatározott hibával terhelt. Minél jobban távolodunk az a ponttól, annál inkább növekszik a közelítés hibája. Az a pont esetünkben egy térbeli pontként fogható fel, így az algoritmus akkor rendelkezik kis hibával, ha az elmozdulásvektor elegendően kicsiny. Ez a követelmény pedig 2 esetben tartható be. Az egyik lehetőség, hogy tag sebesség-idő függvénye nem vesz fel nagy értékeket, a másik lehetőség pedig az, hogy az algoritmus pozíciófrissítési sebessége elegendően nagy.

2.3. Fang módszere

Bertrand T. Fang bizonyította 1989-ben először, hogy ha a TDoA által felhasznált egyenletekben az anchorok pozícióit tekintve néhány kényszert alkalmazunk, akkor a hiperbolikus nemlineáris egyenletrendszerből átalakítások sorozatával koordinátáinként egymástól független másodfokú egyenletre juthatunk [26]. A módszer hasz-

nálhatóságának tehát feltétele, hogy az anchorok által kifeszített térbeli struktúra speciális alakú, melyet a következő 4 feltétel fogalmaz meg:

$$\mathbf{p}_1 = [0, 0, 0]^T \quad (2.23)$$

$$\mathbf{p}_2 = [x_2, 0, 0]^T \quad (2.24)$$

$$\mathbf{p}_3 = [x_3, y_3, 0]^T \quad (2.25)$$

$$\mathbf{p}_4 = [x_4, y_4, z_4]^T \quad (2.26)$$

A fenti 4 feltétel egyúttal azt is jelenti, hogy 4-nél több anchor általános esetben nem ad többletinformációt. A 2.3 egyenlet d_i -re rendezésével, és négyzetre emelésével kapjuk az alábbi egyenletet:

$$\begin{aligned} (d_{1,i} + d_1)^2 &= d_i^2 \\ d_{1,i}^2 + 2 \cdot d_{1,i} \cdot d_1 + d_1^2 &= d_i^2 \quad i = 2, 3, 4 \end{aligned} \quad (2.27)$$

Átrendezve a 2.27 egyenletet, és felhasználva a 2.4 és a 2.5 összefüggéseit, a következő általános egyenletre jutunk:

$$\begin{aligned} d_{1,i}^2 + 2 \cdot d_{1,i} \cdot d_1 &= x_i^2 - 2x_i x_T + x_T^2 + y_i^2 - 2y_i y_T + y_T^2 + z_i^2 - 2z_i z_T + z_T^2 + \\ &- x_1^2 + 2x_1 x_T - x_T^2 - y_1^2 + 2y_1 y_T - y_T^2 - z_1^2 + 2z_1 z_T - z_T^2 \quad i = 2, 3, 4 \end{aligned} \quad (2.28)$$

A 2.23, 2.24, 2.25 és 2.26 feltételek alapján, illetve ezektől függetlenül is egyszerűsíthető a 2.28 egyenlet, melyeket alkalmazva jutunk az alábbi egyenletre, mely a módszerre jellemző általános egyenletként fogható fel:

$$d_{1,i}^2 + 2 \cdot d_{1,i} \cdot d_1 = x_i^2 + y_i^2 + z_i^2 - 2x_i x_T - 2y_i y_T - 2z_i z_T \quad i = 2, 3, 4 \quad (2.29)$$

Ismeretlennek tekinthetők a tag koordinátái, illetve a d_1 távolság. Az anchorok koordinátái adottak, a $d_{1,i}$ távolságkülönbségek pedig a mérési eredményekből számolhatóak. A következő lépésben felírható a megoldandó egyenletrendszer a 2.29 egyenlet alapján, ahol továbbra is célszerű figyelembe venni a a 2.23, 2.24, 2.25 és 2.26 feltételeket.

$$d_{1,2}^2 + 2 \cdot d_{1,2} \cdot d_1 = x_2^2 - 2x_2 x_T \quad (2.30)$$

$$d_{1,3}^2 + 2 \cdot d_{1,3} \cdot d_1 = x_3^2 + y_3^2 - 2x_3 x_T - 2y_3 y_T \quad (2.31)$$

$$d_{1,4}^2 + 2 \cdot d_{1,4} \cdot d_1 = x_4^2 + y_4^2 + z_4^2 - 2x_4 x_T - 2y_4 y_T - 2z_4 z_T \quad (2.32)$$

$$d_1 = \sqrt{x_T^2 + y_T^2 + z_T^2} \quad (2.33)$$

A 4 fenti egyenletből kiindulva a tag 3 koordinátájára egymástól függetlenül egy-egy másodfokú egyenletre juthatunk. Ezen másodfokú egyenletek saját és levezetésük

terjedelme egyaránt jelentős, ezért csak a tag x koordinátájára vonatkozó egyenlet kerül közlésre teljes formában.

$$\begin{aligned}
& x_T^2 \cdot \left[\frac{1}{z_4} \cdot \left(4d_{1,2}^2 \cdot \left[y_4 \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right) - \frac{x_4}{z_4} + \frac{x_2 d_{1,4}}{x_4 d_{1,2}} \right]^2 \right) + \right. \\
& \qquad \qquad \qquad \left. + 4d_{1,2}^2 \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right)^2 - 4x_2^2 + 4d_{1,2}^2 \right] + \\
& + x_T \cdot \left[4x_2^3 - 4x_2 d_{1,2}^2 + 8d_{2,1}^2 \cdot \left[\frac{y_4}{z_4} \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right) - \frac{x_4}{z_4} + \frac{x_2 d_{1,4}}{x_4 d_{1,2}} \right] \cdot \right. \\
& \qquad \qquad \qquad \cdot \left[\frac{x_4^2 + y_4^2 - d_{1,4}^2 - y_3 y_4 + d_{1,2} d_{1,4}}{2z_4} + \frac{-x_3^2 y_4 + y_4 d_{1,3}^2}{2y_3 z_4} + \right. \\
& \qquad \qquad \qquad \left. + \frac{-x_2^2 d_{1,4} - y_4 d_{1,2} d_{1,3}}{2y_3 z_4} + \frac{x_2^2 y_4 d_{1,3}}{2y_3 z_4 d_{1,2}} + \frac{z_4}{2} \right] - \\
& \qquad \qquad \qquad \left. + 8d_{1,2}^2 \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right) \cdot \left(\frac{y_3}{2} + \frac{x_3^2 - d_{1,3}^2 + d_{1,2} d_{1,3}}{2y_3} - \frac{x_2^2 d_{1,3}}{2y_3 d_{1,2}} \right) \right] + \\
& + \left[4d_{1,2}^2 \cdot \left(\frac{y_3}{2} + \frac{x_3^2 - d_{1,3}^2 + d_{1,2} d_{1,3}}{2y_3} - \frac{x_2^2 d_{1,3}}{2y_3 d_{1,2}} \right)^2 - x_2^4 - d_{1,2}^4 + \right. \\
& \qquad \qquad \qquad + 4d_{1,2}^2 \cdot \left(\frac{x_4^2 + y_4^2 - d_{1,4}^2 - y_3 y_4 + d_{1,2} d_{1,4}}{2z_4} + \frac{-x_3^2 y_4 + y_4 d_{1,3}^2}{2y_3 z_4} + \right. \\
& \qquad \qquad \qquad \left. + \frac{-x_2^2 d_{1,4} - y_4 d_{1,2} d_{1,3}}{2y_3 z_4} + \frac{x_2^2 y_4 d_{1,3}}{2y_3 z_4 d_{1,2}} + \frac{z_4}{2} \right) + 2x_2^2 d_{1,2}^2 \left. \right] = 0 \quad (2.34)
\end{aligned}$$

$$y_T^2 \cdot (a_y) + y_T \cdot (b_y) + (c_y) = 0 \quad (2.35)$$

$$z_T^2 \cdot (a_z) + z_T \cdot (b_z) + (c_z) = 0 \quad (2.36)$$

A megoldás pedig a másodfokú egyenlet megoldóképletével számítható:

$$x_{T1,2} = \frac{-b_x \pm \sqrt{b_x^2 - 4a_x c_x}}{2a_x} \quad (2.37)$$

Ahol a_x , b_x , c_x rendre a 2.34 egyenletben kékkel, pirossal, és feketével bekeretezett tényezők.

2.4. Friedlander módszere

Benjamin Friedlander 1987-ben fejtette ki módszerét amely a következőkben bemutatott módon, egy segédmátrix segítségével kiküszöböl egy vektort amely többlet-információt nem hordoz [27]. Kiindulásként a 2.28 egyenlet átrendezésével és az $x_{i,1}x_T = x_i x_T - x_1 x_T$ összefüggés felhasználásával jutunk az alábbi egyenletre:

$$x_{i,1}x_T + y_{i,1}y_T + z_{i,1}z_T = \frac{1}{2} \cdot (x_i^2 + y_i^2 + z_i^2 - x_1^2 - y_1^2 - z_1^2 - d_{1,i}^2) - d_{1,i}^2 d_1$$

$$i = 2 \dots M \quad (2.38)$$

Ez a módszer semmilyen megkötést nem ad az anchorok koordinátáira vonatkozóan, így ez esetben a 2.28 egyenletben egyszerűsíteni csak a tag koordinátáinak négyzeteivel lehet. A 2.38 egyenletet vektor mátrixos alakba írva a következőképp fogalmazhatunk meg:

$$\underbrace{\begin{bmatrix} x_{2,1} & y_{2,1} & z_{2,1} \\ x_{3,1} & y_{3,1} & z_{3,1} \\ \vdots & \vdots & \vdots \\ x_{M,1} & y_{M,1} & z_{M,1} \end{bmatrix}}_{\underline{\underline{S}}} \cdot \underbrace{\begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix}}_{\mathbf{p}_T} =$$

$$= \frac{1}{2} \cdot \underbrace{\begin{bmatrix} x_2^2 + y_2^2 + z_2^2 - x_1^2 - y_1^2 - z_1^2 - d_{2,1}^2 \\ x_3^2 + y_3^2 + z_3^2 - x_1^2 - y_1^2 - z_1^2 - d_{3,1}^2 \\ \vdots \\ x_M^2 + y_M^2 + z_M^2 - x_1^2 - y_1^2 - z_1^2 - d_{M,1}^2 \end{bmatrix}}_{\underline{U}} - d_1 \cdot \underbrace{\begin{bmatrix} d_{2,1} \\ d_{3,1} \\ \vdots \\ d_{M,1} \end{bmatrix}}_{\underline{P}} \quad (2.39)$$

Jól láthatóan a 2.39 egyenletben szerepel a keresett \mathbf{p}_T vektor, a tisztán anchor koordinátáktól függő $\underline{\underline{S}}$ együtthatómátrix, a mérések (és anchor pozíciók) útján meghatározott \underline{U} vektor, illetve egy \underline{P} -vel definiált vektor is, amelyet eliminálva egy lineáris egyenletrendszert kaphatunk.

A \underline{P} vektor eliminálásának folyamata az, hogy a 2.39 egyenlet mindkét oldalát beszorozzuk balról egy $\underline{\underline{N}}$ segédmátrixszal (2.41 egyenlet), amelyet a következő egyenlet definiál:

$$\underline{\underline{N}} = (\underline{\underline{E}} - \underline{\underline{Z}}) \cdot \text{diag}\{\underline{P}\}^{-1} \quad (2.40)$$

Ahol

- $\underline{\underline{E}}$ az egységmátrix

- $\underline{\underline{Z}}$ egy cirkuláris permutáló mátrix:
$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & & 1 & \ddots & \vdots \\ 0 & & \ddots & \ddots & 0 \\ 0 & 0 & & & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

- $\text{diag}\{\underline{\underline{P}}\}^{-1} = \begin{bmatrix} \frac{1}{d_{2,1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{d_{3,1}} & & \ddots \\ \vdots & & \frac{1}{d_{4,1}} & \vdots \\ & \ddots & & \ddots & 0 \\ 0 & & \cdots & 0 & \frac{1}{d_{M,1}} \end{bmatrix}$

$$\underbrace{\underline{\underline{N}} \cdot \underline{\underline{S}}}_{\underline{\underline{T}}} \cdot \hat{\mathbf{p}}_{\text{T}} = \frac{1}{2} \cdot \underline{\underline{N}} \cdot \underline{\underline{U}} - \underbrace{d_1 \cdot \underline{\underline{N}} \cdot \underline{\underline{P}}}_{\underline{\underline{0}}} \quad (2.41)$$

Elvégezve a balról való beszorzást, már egy korábban ismert ismerttetett eljárással számítható az optimális megoldás az alábbi összefüggés segítségével:

$$\hat{\mathbf{p}}_{\text{T}} = \frac{1}{2} \cdot \underline{\underline{T}}^+ \cdot \underline{\underline{N}} \cdot \underline{\underline{U}} \quad (2.42)$$

3. fejezet

Összehasonlító vizsgálat

3.1. Algoritmusok működési tulajdonságai

A következőkben összehasonlításra kerülnek a különböző algoritmusok az alábbi szempontok szerint:

- szükséges anchorok száma
- kezdeti feltételek
- hibaterjedés

Az LSM (Least Square Method-Legkisebb négyzetek módszere) esetében az ismeretlen \underline{X} kifejezéséhez minimum 4 egyenletre van szükség, amelyhez legalább 5 anchor üzemelése szükséges. Kezdeti feltételeket nem kell szabni a módszer működéséhez, tehát az első mérési eredmény sorozatból számítható a becsült pozíció. Nincs szükség a priori ismeretre a tag kezdeti pozícióját illetően.

Foy módszerét alkalmazva a közvetlenül keresett ismeretlen mindig egy $\Delta \mathbf{p}_T$ elmozdulásvektor, amelynek kifejezéséhez szükséges minimális egyenletek száma 3, ehhez pedig 4 anchor tartozik. Az időkülönbség képzés miatt az ismeretlenek számánál egynél több anchor szükséges tehát a számításhoz, csak úgy mint LSM esetén. A kezdeti feltételeket vizsgálva viszont megvan az ára a kevesebb anchornak. Ezt a módszert alkalmazva a priori ismeret nélkül $(\mathbf{p}_{T,0})$ nem kezdődhet el a pozícióbecslés. Ennek oka, hogy az első elmozdulásvektor kiszámítása után nem lenne mihez hozzáadni azt. A módszer sajátossága ezen felül az, hogy a kezdeti pozíció ismeretének hibája integrálódik az egyes számítási ciklusok során, azaz az egyes becslési eredmények nem függetlenek egymástól.

Fang módszerét előtérbe helyezve a tag pozícióját közvetlenül másodfokú egyenletek megoldásai adják, amely nem egy mátrixegyenlet optimális megoldása, hanem egy zajjal terhelt mérési eredményből számított pozíció. A 2.34 egyenlet alapján itt is 4 anchorra van szükség a működéshez, de nagy különbség az eddigiekhez képest,

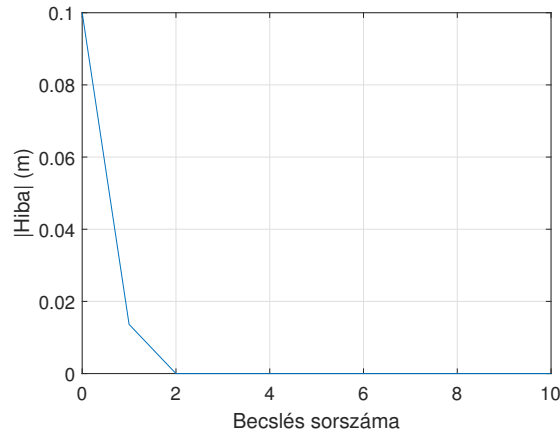
hogy egyértelműen meghatározott az anchorok száma. Minden egyes pozíciószámítási ciklusban 2 eredményt ad vissza az algoritmus a tag x , y , és z koordinátájára, amelyek között általában nehéz eldönteni melyik a helyes. Emiatt a tag kezdeti pozícióját ebben az esetben is ismerni kell, majd egy térbeli komparálást alkalmazva lehet kiszűrni mindig a soron következő eredménypárok közül a hamis megoldást. A térbeli szűrés megvalósításához figyelembe kell venni a tag azon időegység alatti elmozdulásának maximumát, amely időegység alatt egy pozíciószámítási ciklus megtörténik. Így ha a kezdeti pozíció ismert, akkor köré képzelve egy ϵ sugarú gömböt, a következő érvényes pozíciószámításnak a gömbön belülré kell esnie. Előnyként említhető, hogy a kezdeti pozíció hibája nem integrálódik, mivel az egyes számítási eredmények egymástól függetlenek (az előző számítási eredmény csak azt segíti, hogy a következő eredmények közül valószínűleg melyik a helyes, de a megoldóképletekben nincs szerepük).

Friedlander módszere lényegét tekintve az LSM módosítása oly módon, hogy az ismeretlen \underline{X} vektorban szereplő d_1 paramétert, illetve az \underline{A} mátrixban szereplő mérésekből származó változókat átrendezi az egyenlet másik oldalára, majd egy segédmátrixot alkalmazva eliminálja az átrendezett egyenletből a d_1 paraméterrel súlyozott vektort. Ezáltal a meghatározandó ismeretlen vektor már csak a tag koordinátáiból fog állni, így minimum 4 anchor kell az algoritmus működéséhez. A kezdeti feltételeket tekintve a mérési eredmények függetlenek egymástól, és nem kell ismerni a tag kezdeti pozícióját sem.

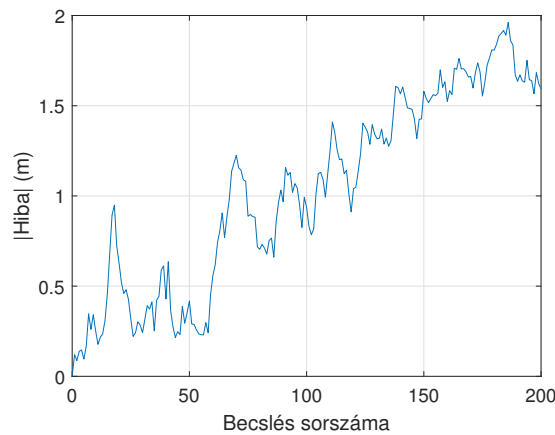
3.2. Szimulációs eredmények

A hibaterjedést MATLAB-ban, általam készített szimulációs kódok segítségével végeztem el, melyek eredményeit az alábbiakban ismertetem.

A 3.1 ábrán az látható, ahogy Foy módszerében egy 0,1 m-es kezdeti pozícióismereti hibával milyen mértékben konvergál a tényleges pozícióba. Ebben az esetben az időmérést nem terheli hiba, azonban a kezdeti pozíciót hibásan kapja meg az algoritmus. Ezzel a szimulációval az igazolható, hogy a módszer mozdulatlan tag esetén (ideális szinkronizációt feltételezve) képes csökkenteni a becslési hibát. A 3.1 ábrán úgy tűnhet, hogy nincs hibája, de valójában az ábrán nehezen kivehető módon $1,83 \cdot 10^{-15}$ m hibával oszcillál a valós pozíció körül. Ez a numerikus megvalósításra jellemző hiba, amely méretét figyelembe véve elhanyagolható. A 3.2 ábrán látható az időmérés hibájának hatása a pozícionálásra a Foy módszer esetén. Standard normális eloszlású hibát feltételez a szimuláció, egy konstans 10^{-10} súlyozással, ezáltal az időméréshez hozzáadott hiba várható értéke 0, szórása 100 ps. A 3.2 ábra alapján ekkora szórású hiba esetén az algoritmus pozícionálási hibája divergál. Iterációs módszerrel azonban megtalálható egy olyan határhelyzet a szórást tekintve, ahol még az algoritmus képes konvergálni a 0 hiba irányába, illetve legalább is



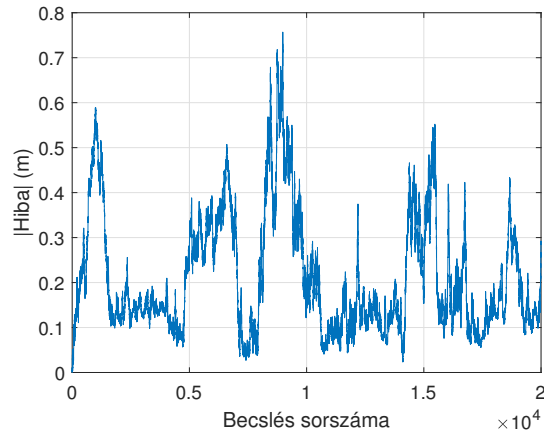
3.1. ábra. Foy algoritmusának konvergálása fix pozícióban



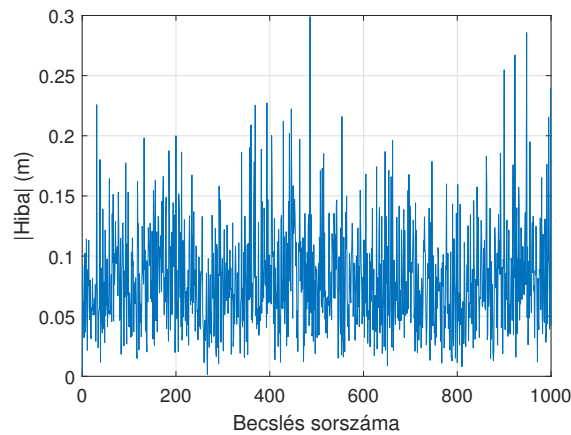
3.2. ábra. Foy algoritmus pozicionálási hibájának növekedése

alacsony szinten tartani a hibát. A 3.3 ábrán látható szimulációs eredmény 10 ps szórás mellett adódott. Az ábrán abszolút értékben van ábrázolva, azonban ez becslési pontonként valójában ellentétes előjelű értékekből áll. Ennek következtében itt több mérés átlagolásával még kellően nagy pontosság érhető el, ha a realizált szinkronizáció pontossága legalább 10 ps, vagy az alatt van.

Fang módszere esetén az első pozíció ismerete a szimulációk alapján egy 2 állapotú döntést befolyásol koordinátáinként, mivel ha az időmérés szinkronizált állapotban történik, akkor ideális esetben a másodfokú egyenletek megoldásai közül az egyik mindig a helyes pozíciót adja vissza. Numerikus hibája elhanyagolható a gyakorlatban. Az algoritmus kódolása során így első becslésnek érdemes mindössze azt a megoldást választani koordinátáinként, amelyik közelebb van az ismertnek tekintett 0. pozícióhoz. A 3.4 ábra mutatja be az algoritmus becslési hibáját abban az esetben, ha a szimuláció minden anchor időmérését 0 várható értékű, 100 ps szórású hibával veszi figyelembe. Jól látható, hogy Foy algoritmusával ellentétben nem egy változó tendenciájú, de növekvő karakterisztikát mutat az idő előrehaladtával, hanem stabilizált várható érték körül marad, a 3.4 ábrán ez 10 cm és 5 cm között

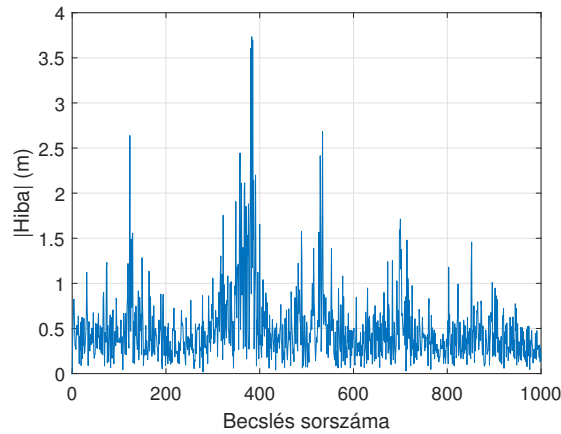


3.3. ábra. Foy algoritmus konvergálási határhelyzet



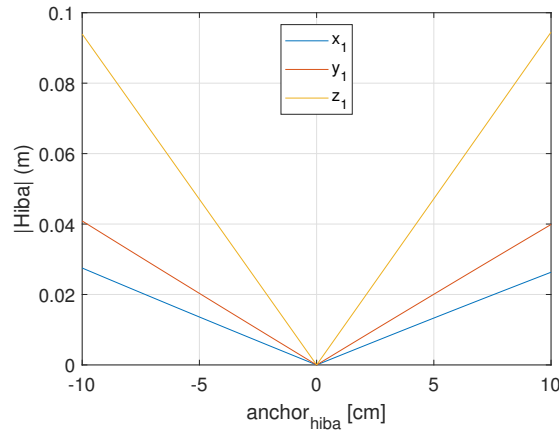
3.4. ábra. Fang algoritmus pozicionálási hibája
100 ps szórású időmérési hiba esetén

található. A 3.5 ábrán megfigyelhető, hogy viselkedik az algoritmus 500 ps szórás esetén, ha minden más szimulációs paraméter rögzítve van. Jól láthatóan a hiba abszolút értékének várható értéke a szórás változásának irányát tekintve hasonlóképp alakul, azaz a szórással együtt nő a pozíció becslési hiba abszolút értéke is. A 3.5 ábrán látható lokális maximumok (főként a 400. becslés előtt található egy ilyen) okai a másodfokú egyenletben keresendők. Az időmérés normális eloszlású hibával való terhelése ugyanis képes olyan szituációt előidézni, hogy amikor a beérkezési időkből időkülönbséget képez az algoritmus, és az időkülönbségek összemérhetővé válnak azzal zajjal, amelyet a szimuláció hozzáad az egyes beérkezési időkhöz, akkor előfordulhat, hogy az időkülönbség így 0 közeli érték lesz. A 2.34 egyenletet, és a hozzá tartozó a 2.37 megoldóképletet figyelembe véve, a tag x koordinátájára a megoldóképlet szerint növekvő értéket fog számítani az algoritmus, ha a $d_{1,2}$ távolságkülönbség abszolút értékben 0-hoz tart, mivel ekkor a megoldóképlet nevezőjében nullához közelítő értékkel oszt. Az algoritmusban a beérkezési időkhöz hozzáadott normális eloszlású időmérési hiba alapvetően kis valószínűséggel produ-



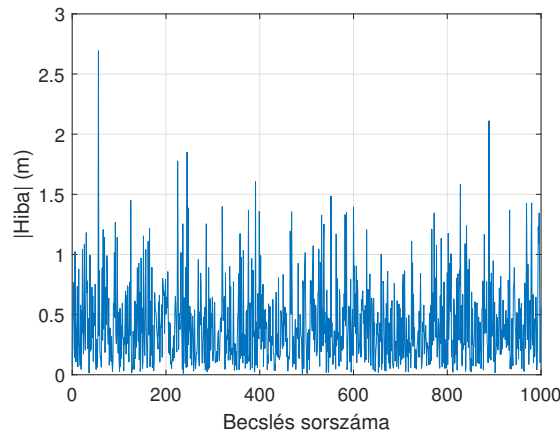
3.5. ábra. Fang algoritmus pozicionálási hibája
500 ps szórású időmérési hiba esetén

kál nagy abszolút értékű hibanövekményt, ezért is jelentkezik kis gyakorisággal az imént említett lokális maximum a becslési hiba abszolút értékében. A gyakorlatban ez a jelenség problémássá válik abban az esetben, ha 2 anchortól közel egyforma távolságra helyezkedik el a tag, ugyanis ekkor a beérkezési idők összemérhetővé válnak, így a 0 várható értékű időmérési hiba a különbségképzésnél nehezen használhatóvá teheti az algoritmust. A szimulációk alapján tehát Fang algoritmus numerikusan rosszul viseli az anchorok és tag által reprezentált térbeli szimmetriákat (amelyek azon alapulnak, hogy a tag távolsága 2 anchorhoz képest közel azonos), amely tehát a Fang algoritmus hátrányaként fogalmazható meg. Egy másik tulajdonsága a Fang módszernek, hogy az anchorok elhelyezése a 2.23, 2.24, 2.25 és 2.26 feltételekhez kötött. Ezek szerint egy adott koordináta rendszert felvéve a térben ezeknek a feltételeknek teljesülnie kell, különben a levezetett másodfokú egyenlet matematikai értelemben érvényét veszti. A gyakorlati eseteket tekintve csak véges pontossággal lehet a fenti feltételeknek megfelelően elhelyezni az anchorokat, így ez a véges pontosság újabb hibát visz a rendszerbe. Ennek a hibának a modellezése egy konstans hozzáadásával történhet az egyes anchorok feltétel szerint 0 koordinátáihoz. Egy konkrét realizációt tekintve ugyanis az anchorok fix rögzítésűek, a pozíciójukat nem terheli számottevő változás. Természetesen a létesítmény rezgései, melyek többek között szeizmikus eredetűek lehetnek kis mértékben elmozdíthatják az anchorokat, de ez a hiba nagyságrendjét tekintve elmarad attól a hibától, amely a kiépítés közben kerül bele a rendszerbe. Az összes feltétel szerint 0 koordináta érzékenysége illetve ezek kombinációjának bemutatásának terjedelme okán néhány variáció kerül bemutatásra. A 3.6 ábra jeleníti meg azt az esetet, amikor az 1 anchor koordinátájaként elmozdul. Gyakorlati realizációt tekintve elegendő modellezni azt az esetet, amikor a koordináták a 0-hoz képest ± 10 cm intervallumban változnak, ennek pedig az az oka, hogy ilyen pontosságra számítani lehet egy megvalósított rendszer esetén. Ha több koordináta egyszerre módosul, akkor a szimulációk alapján legrosszabb



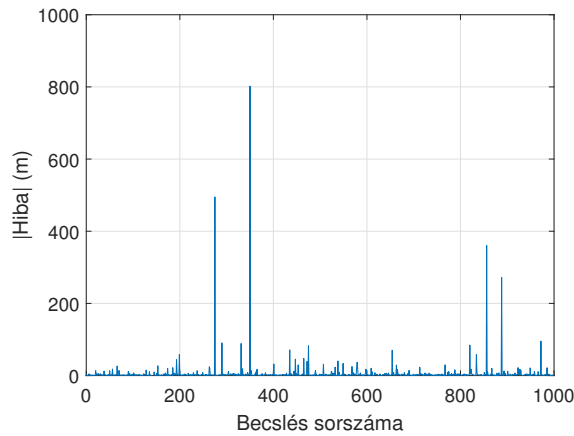
3.6. ábra. Fang algoritmus pozicionálási hibájának érzékenysége az 1. anchor elmozdulására koordinátánként

esetet feltételezve a koordinátánkénti elmozdulások összeadódásával lehet számolni, ezt a szimulációk alátámasztják. A 3.6 ábra alapján elmondható, hogy néhány cm még belefér, és nem veszít sokat a pontosságából az algoritmus. A többi algoritmus esetén csekélyebb az a hibafaktor, amelyet az anchorok elhelyezése hordoz, ennek valószínűleg a numerikus megvalósítás az oka.



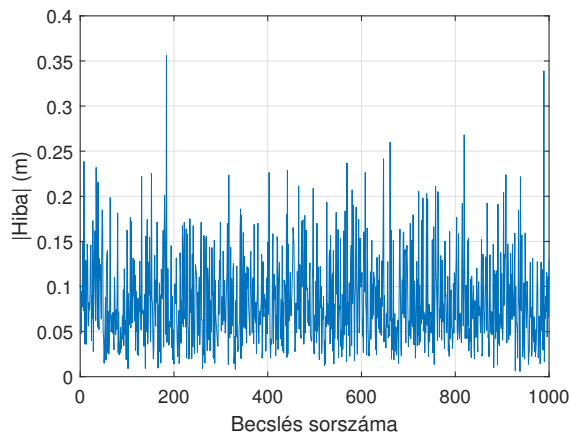
3.7. ábra. LSM pozicionálási hibája 100 ps szórású időmérési hiba esetén

Az LSM esetében kezdeti pozíciót nem kell ismerni, így ez nem szimulálható. Ezzel szemben az időszinkron által bevitt hiba igen. Fang módszeréhez képest az LSM átlagosan jóval rosszabbul viseli a időbeli szinkronizálási hibákat. A 3.7 ábrán látható, hogy 100 ps szórású hiba esetén a pozícióbecslés hibájának abszolút értékének várhatóértéke valamivel 0,5 m alatt található. A 3.8 ábrán érzékelhető az, hogy bizonyos becslési esetekben kiugróan magas hibát produkál az algoritmus. Ez a Fang módszerhez hasonlóan szintén numerikus eredetű. Az LSM numerikus megvalósításában egy lineáris egyenletrendszer megoldása a feladat, amelyben a távolságkülönbségek zajjal terheltek. Ilyen esetben az egyenletrendszerben szereplő



3.8. ábra. LSM pozicionálási hibája
500 ps szórású időmérési hiba esetén

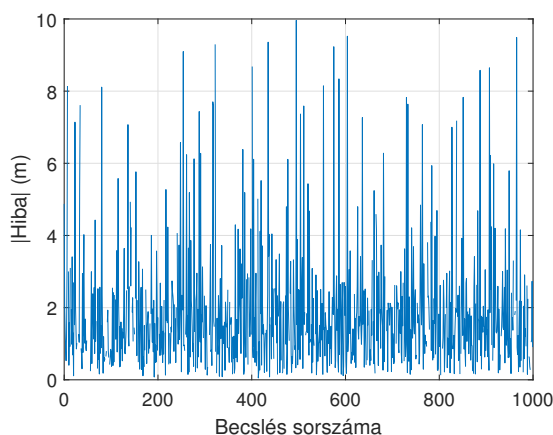
mérési eredményekből számított paraméterek okozhatják azt, hogy rosszul kondicionált/numerikusan instabil egyenletrendszer jöjjön létre. Ezt matematikai absztrakcióval úgy lehet megfogalmazni, hogy az egyenletek között lineáris összefüggés kialakulásának lehetősége áll fent, ha a zajjal való terhelést úgy tekintjük mint egy olyan paramétert amely a lineáris összefüggés mértékét kis mértékben de változtatja. A gyakorlatban ez ellen egy hatékony védekezési lehetőség (ebben a konkrét esetben), ha úgy kerülnek megválasztásra az anchorok koordinátái, hogy azok lineáris összefüggését tekintve minél inkább távol legyenek, ezáltal csökkenthető a numerikus instabilitás kockázata. A szimulációk során kiderült, hogy ez a lehetőség a hiba csökkentésére csak akkor működik, ha egy bizonyos határt nem halad meg az időmérési hibának a szórása. 500 ps szórás esetén már oly mértékben módosul az



3.9. ábra. LSM pozicionálási hibája
100 ps szórású időmérési hiba esetén

egyenletrendszer, hogy a relatív pontosság növelésére csak utólagos adatfeldolgozás útján kerülhet sor. A 3.9 ábrán látható az, hogy módosul a hiba abszolút értéke 100 ps szórás mellett abban az esetben, ha egy másik anchor elhelyezést vesz figye-

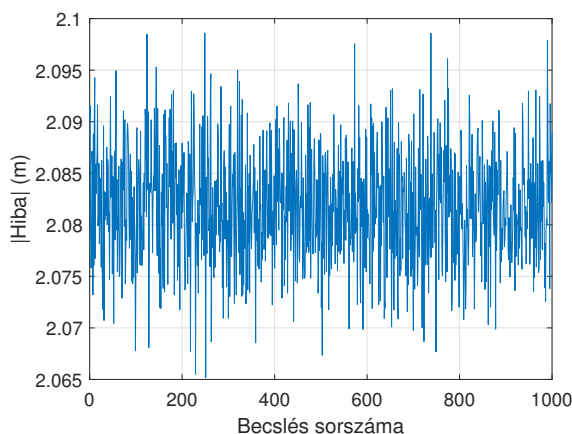
lembe a szimuláció. Jól láthatóan szignifikáns a hiba abszolút értékének csökkenése átlagosan, és az egyes becslési pontokat tekintve is. A 3.10 ábrán látható eredmény



3.10. ábra. LSM pozicionálási hibája
500 ps szórású időmérési hiba esetén

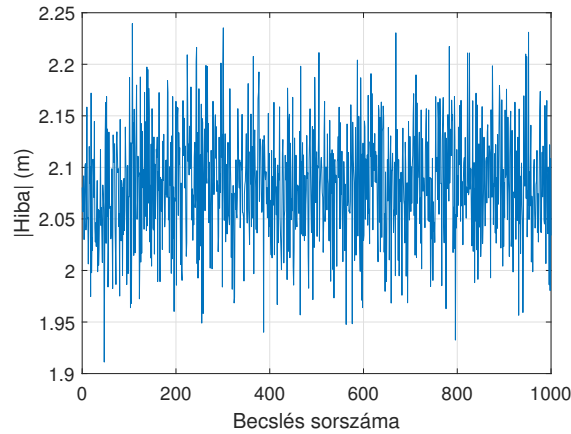
előkészülete az, hogy a szimuláció nem veszi figyelembe azokat abszolút értékben azokat a hibákat, amelyek 10 m-nél nagyobbak, így jobban kivehető, hogy mekkora átlagos pontossággal rendelkezik az algoritmus. A hiba abszolút értékének várható értéke ebben az esetben már 1...2 m körül található, így ekkora időmérési hiba mellett az algoritmus már a gyakorlatban lényegét tekintve nem alkalmazható nagy pontosságú beltéri helymeghatározásra.

Friedlander módszerét alkalmazva a tag kezdeti pozíciójára szintén nincs szükség, így ennek a kezdeti pozíció hiányát szimulálni nem lehetséges. A 3.11 ábrán látható



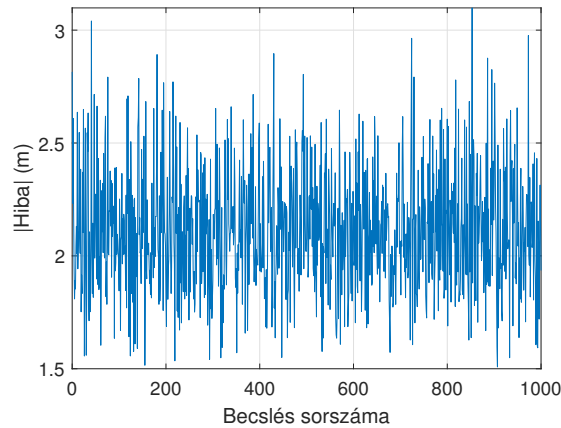
3.11. ábra. Friedlander algoritmus pozicionálási hibája
1 ps szórású időmérési hiba esetén

szimulációs eredmény bemutatását az motiválja, hogy rendkívül kis szórás esetén is a többi algoritmushoz képest nagy hibával rendelkeznek. Ez az algoritmus tehát alapvetően nagyon jó szinkronizáltság esetén is kis pontossággal rendelkezik. A 3.12 ábrán, és a 3.13 ábrán látható eredmény reprezentálja a hiba abszolút értékének



3.12. ábra. Friedlander algoritmus pozicionálási hibája
100 ps szórású időmérési hiba esetén

alakulását abban az esetben, ha az időbeli szórás 100 ps illetve 500 ps értékkel veszi figyelembe a szimuláció. Bár a pontosság továbbra sem az előnye az algoritmusnak, azonban észrevehető egy másik tulajdonság, amely például az LSM esetében problémát jelentett. 100 ps és 500 ps esetében sem lépnek fel kiugróan nagy abszolút értékű hibával rendelkező pozícióbecslési eredmények. Ez valószínűleg annak a 2.41 egyen-



3.13. ábra. Friedlander algoritmus pozicionálási hibája
500 ps szórású időmérési hiba esetén

letben található \underline{N} segédmátrixnak köszönhető, amellyel lényegében a d_1 ismeretlent (és hozzá tartozó \underline{P} vektort) eliminálja a mátrixegyenletből. \underline{N} általi szorzás útján az egész mátrixegyenlet transzformálódik, így az veszít a numerikus instabilitásából, ezáltal az időbeli hozzáadott zaj kevésbé képes jelentékenyen hozzájárulni a becslési hiba növekedéséhez.

4. fejezet

Összefoglalás

Az előzőekben matematikai úton levezetett illetve bemutatott algoritmusok a teljesség igénye nélkül alapvető jelentőségűnek számítanak a TDoA elven működő algoritmusok területén. Az 1970-es illetve 1980-as években született algoritmusok fő gondolatai kombinálhatók, így úgynevezett hibrid algoritmusok hozhatók létre, amennyiben a kombinált módszerek előnyeinek ötvözése a cél. Ezeknek az algoritmusoknak a működése utólagos adatfeldolgozással támogatható. A bemutatott 4 algoritmus szimulációi alapján a 4.1 táblázat foglalja össze tömören a kapott eredményeket.

	LSM	Foy	Fang	Friedlander
konzisztens egyenletrendszerhez szükséges anchorok száma (n)	≥ 5	≥ 4	$\equiv 4$	≥ 4
a priori ismeretek szükségessége	nem	igen	igen	nem
becslési hiba abszolút értékének átlaga (@100 ps)	$\simeq 0,4$ m	\nexists	$\simeq 0,075$ m	$\simeq 2,1$ m
becslési hiba integrálódás	nincs	van	nincs	nincs
numerikus instabilitás	jelentős	nem jellemző	jelentős	alacsony

4.1. táblázat. Tulajdonságok és eredmények összehasonlítása

A táblázatban szereplő tulajdonságok között az első főleg gazdasági/kiépíthetőségi szempontokat előtérbe helyezve válik fontossá, tehát melyik algoritmus esetén hány anchorral kell számolni minimálisan. Az a priori ismeretek szükségessége első sorban akkor válik fontossá, amikor a helymeghatározási szolgáltatás ki-be kapcsolható, ugyanis ekkor minden egyes bekapcsoláskor egy másik módszerre támaszkodva kell tudomást szerezni a tag jelenlegi pozíciójáról. Ha a helymeghatározás például drónok pozíciójának követésére van alkalmazva, és a teljes repülési időtartam alatt üzemel, akkor ugyanazon le- és felszállópálya esetén a tag kiinduló helyzete ismert. A becslési hiba abszolút értékének átlaga közelítőleg került megadásra a 4.1

táblázatban, viszont kiindulási alapnak megfelel mind a 4 algoritmus esetén. Ezek alapján a Fang módszer hibát tekintve kiemelkedő. A becslési hiba integrálódásának hatása egy nagy pontosságú helymeghatározó rendszerrel meglehetősen drámai következményekkel járhat, így törekedni ezen tulajdonság elkerülésére. A numerikus instabilitás egy olyan jellemző, amelyet jó szinkronizáltság és időmérés mellett megfelelő anchor elhelyezésekkel és utólagos adatfeldolgozással kézben lehet tartani. Ilyen utólagos feldolgozás egy lehetséges megvalósítása a Kálmán-szűrő használata.

A 4.1 táblázat a fentiek alapján egy konkrét realizációt tekintve megfelelő alapismereteket nyújt ahhoz, hogy a gyakorlatban adott feltételeknek megfelelő algoritmus kiválasztása egyszerűsödjön, illetve annak optimalizálása megtörténhessen.

Irodalomjegyzék

- [1] Ipar 4.0 weblapja: <http://www.ediplomat.com>, Utolsó elérés dátuma: 2021.10.18.
- [2] P. Puricer and P. Kovar, „Technical limitations of gnss receivers in indoor positioning,” in *2007 17th International Conference Radioelektronika*, pp. 1–5, 2007.
- [3] M. Piras and A. Cina, „Indoor positioning using low cost gps receivers: Tests and statistical analyses,” in *2010 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–7, 2010.
- [4] IT business cikk: <https://www.it-business.hu>, Utolsó elérés dátuma: 2021.10.10.
- [5] UWB rövid történeti összefoglalója: <https://www.eliko.ee/>, Utolsó elérés dátuma: 2021.10.01.
- [6] M. Ture and A. Hatipoglu, „Indoor location finding of the transmitter based on bluetooth received signal strength,” in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–5, 2019.
- [7] K. Phutcharoen, M. Chamchoy, and P. Supanakoon, „Accuracy study of indoor positioning with bluetooth low energy beacons,” in *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON)*, pp. 24–27, 2020.
- [8] V. Grinyak, A. Shurygin, and A. Devyatisilnyi, „Accuracy of indoor navigation with bluetooth beacons,” in *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pp. 1–4, 2019.
- [9] *Revision of Part 15 of the Commission’s Rules Regarding Ultra WideBand Transmission Systems*. First Report and Order, (First R&O), FCC ET Docket 98-153,, Apr. 2002 Apr. 22.
- [10] *IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control*

(MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs): Amendment 1: Add Alternate PHYs, 2007.

- [11] „Ieee standard for low-rate wireless networks,” *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020.
- [12] Proxy NV weblapja: <https://www.pozyx.io/>, Utolsó elérés dátuma: 2021.10.04.
- [13] Sávhasználati feltételek és frekvenciagazdálkodási követelmények az NMHH honlapján: <https://stir.nmhh.hu/>, Utolsó elérés dátuma: 2021.10.10.
- [14] Apple honlapja: <https://www.apple.com/>, Utolsó elérés dátuma: 2021.10.10.
- [15] Iphone 13 adatok az Apple honlapján: <https://www.apple.com/>, Utolsó elérés dátuma: 2021.10.10.
- [16] AirTag adatok az Apple honlapján: <https://www.apple.com/>, Utolsó elérés dátuma: 2021.10.10.
- [17] Tech2 cikk: <https://www.tech2.hu/>, Utolsó elérés dátuma: 2021.10.10.
- [18] Computerworld cikk: <https://www.computerworld.com/>, Utolsó elérés dátuma: 2021.10.10.
- [19] Q. Zhou, C. Shen, X. Chen, and G. Feng, „Uwb wireless positioning technology in the application,” in *2016 IEEE Conference on Wireless Sensors (ICWiSE)*, pp. 106–109, 2016.
- [20] Y. Jiang and V. C. M. Leung, „An asymmetric double sided two-way ranging for crystal offset,” in *2007 International Symposium on Signals, Systems and Electronics*, pp. 525–528, 2007.
- [21] M. Kwak and J. Chong, „A new double two-way ranging algorithm for ranging system,” in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, pp. 470–473, 2010.
- [22] Y. Cheng and T. Zhou, „Uwb indoor positioning algorithm based on tdoa technology,” in *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 777–782, 2019.
- [23] B. Choi, K. La, and S. Lee, „Uwb tdoa/toa measurement system with wireless time synchronization and simultaneous tag and anchor positioning,” in *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1–6, 2018.

- [24] X. Li and T. Zhang, „Research on improved uwb localization algorithm in nlos environment,” in *2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pp. 707–711, 2018.
- [25] W. H. FOY, „Position-location solutions by taylor-series estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-12, no. 2, pp. 187–194, 1976.
- [26] B. Fang, „Simple solutions for hyperbolic and related position fixes,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 748–753, 1990.
- [27] B. Friedlander, „A passive localization algorithm and its accuracy analysis,” *IEEE Journal of Oceanic Engineering*, vol. 12, no. 1, pp. 234–245, 1987.

Ábrák jegyzéke

1.1. Elemi UWB jellel való korreláció kimutatása [10]	7
1.2. Többutas terjedés beltérben	7
1.3. UWB jel vételi időfüggvénye [12]	8
1.4. Pozicionálás folyamata 2 dimenzióban	11
1.5. Egyutas üzenetváltási séma	11
1.6. Kétutas üzenetváltási séma	12
1.7. Szimmetrikus kétutas üzenetváltási séma	12
1.8. Pozicionálás TDoA alkalmazása esetén	13
1.9. Mért távolság hibája a tag és anchor közti frekvenciakülönbség függvényében TWR esetén	14
1.10. Mért távolság hibája a tag és anchor közti frekvenciakülönbség függvényében SDS-TWR esetén	15
2.1. TDoA elrendezés	17
3.1. Foy algoritmusának konvergálása fix pozícióban	29
3.2. Foy algoritmus pozicionálási hibájának növekedése	29
3.3. Foy algoritmus konvergálási határhelyzet	30
3.4. Fang algoritmus pozicionálási hibája 100 ps szórású időmérési hiba esetén	30
3.5. Fang algoritmus pozicionálási hibája 500 ps szórású időmérési hiba esetén	31
3.6. Fang algoritmus pozicionálási hibájának érzékenysége az 1. anchor elmozdulására koordinátánként	32
3.7. LSM pozicionálási hibája 100 ps szórású időmérési hiba esetén	32
3.8. LSM pozicionálási hibája 500 ps szórású időmérési hiba esetén	33
3.9. LSM pozicionálási hibája 100 ps szórású időmérési hiba esetén	33
3.10. LSM pozicionálási hibája 500 ps szórású időmérési hiba esetén	34
3.11. Friedlander algoritmus pozicionálási hibája 1 ps szórású időmérési hiba esetén	34
3.12. Friedlander algoritmus pozicionálási hibája 100 ps szórású időmérési hiba esetén	35

3.13. Friedlander algoritmus pozicionálási hibája 500 ps szórású időmérési hiba esetén	35
---	----