



**Budapesti Műszaki és Gazdaságtudományi Egyetem**

Villamosmérnöki és Informatikai Kar

Méréstechnikai és Információs Tanszék

# **Valós idejű akusztikus iránydetektálás sztereo mikrofonokkal**

*Készítette*

Gera Nándor

*Konzulens*

Dr. Orosz György

2018

# TARTALOMJEGYZÉK

Kivonat.....	5
Abstract.....	6
1. Bevezetés .....	7
1.1. Lokalizáció területei .....	7
1.1.1. Rádiófrekvenciás lokalizáció.....	8
1.1.2. Bluetooth, WiFi alapú lokalizációk .....	8
1.1.3. Ultrahang alapú lokalizáció .....	9
1.1.4. Infrahang alapú lokalizáció.....	9
1.1.5. Akusztikus lokalizáció .....	9
2. Rendszerterv .....	10
2.1. Célok, követelmények .....	10
2.2. Blokkdiagram.....	11
2.3. Rendszer felépítése .....	14
3. Akusztikus lokalizáció.....	17
3.1. Korreláció .....	22
3.2. Beamforming .....	24
4. Megvalósított rendszer.....	26
4.1. Előfeldolgozás .....	27
4.1.1. Szegmentálás .....	27
4.1.2. Szűrés.....	29
4.2. Triggerelés és jelminőség .....	33
4.2.1. Triggerszint meghatározása .....	33
4.2.2. Jelminőség mérése .....	34
4.3. Normalizálás .....	36
4.4. Utófeldolgozás .....	40
4.4.1. Korrekció .....	40
4.4.2. Hampel algoritmus.....	43
4.5. Fúzió .....	44
5. Tesztelés.....	46
5.1. Offline tesztek.....	46

5.1.1. Állandó pozíciójú forrás .....	47
5.1.2. Mozgó forrás.....	52
5.2. Valós idejű tesztek .....	54
5.3. Fúzió hatása .....	57
6. Összefoglalás, konklúzió .....	61
7. Köszönetnyilvánítás.....	62
8. Irodalomjegyzék .....	63

## HALLGATÓI NYILATKOZAT

Alulírott Gera Nándor, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2018. 12. 11.

.....  
Gera Nándor

## Kivonat

Egyes alkalmazásokban fontos lehet a felhasználó, az ember pozíciójának meghatározása valamilyen zárt térben, például egy házban, esetleg egyetlen szobában vagy teremben. Ilyen alkalmazás lehet egy okos otthon, vagy olyan egyéb hangvezérlésű rendszerek, melyeknél fontos az akusztikus forrás meghatározása.

Szakedolgozatom során prezentálni fogom, hogyan terveztem meg az akusztikus jelek feldolgozását, szűrését. Bemutatom az általam alkalmazott különböző lokalizációs algoritmusokat, azok működési elvét, illetve fúziójukat a megbízhatóbb, robosztusabb működés érdekében. A lokalizálásra használt algoritmusok általában idő- vagy frekvenciatartományban késleltetik egymáshoz képest a vett jeleket, és valamilyen összegző módszer segítségével adják meg a forrás irányát, vagy helyét a síkban, térben. Mivel ezek a módszerek a zárt tér tulajdonságaiból következő reflexiók miatt nem működnek megfelelően elő-, és/vagy utófeldolgozás nélkül, ezért a helymeghatározás pontosságának javítása érdekében megismertem az emberi hang, illetve az általam használt akusztikus lokalizációra használt algoritmusok (korreláció, beamforming) azon tulajdonságait, melyekkel az alkalmazás számára megfelelő nagyságú és rendszerességi hibán belüli pontosságú helymeghatározást tudtam megvalósítani valós időbeli működés közben.

## **Abstract**

Localizing the source of the audio sound might be important in some kinds of applications, for example in a house or just in one room. Smart homes and other applications where the position of the acoustic sound is critical are some of them.

In this thesis I will present how the processing and filtering of the sound was designed and implemented. The used acoustic localization algorithms will be introduced, how they work, and the fusion of the methods which helps for a more reliable and robust operation. Algorithms for localization are based on delaying the received sampled signal between each other, and summarize them somehow, so that they can define from which way the sound has been arrived. Detecting the position of the source may be incorrect because of the reflection of the sound or the noisy sampled signal, therefore I examined the properties of the human sound and the chosen methods (correlation, beamforming) so that the application could operate in a more reliable and consistent way in real-time.

# 1. Bevezetés

Szakedolgozatom témája, az akusztikus lokalizáció számos alkalmazási területtel rendelkezik. Ilyen lehet a manapság egyre inkább elterjedő okos eszközök egy egészen új területen való megjelenése, az okos otthon. A beszélő pozíciójának meghatározása után a program az adott helyen szabályozhatja a fűtést vagy a klímaberendezést, kapcsolhatja be vagy ki a különböző elektromos készülékeket, mint például a világítást.



1-1. Okos otthon [3]

Ezen kívül olyan, még intelligensebb funkciókat is betölthet, mint az energiatakarékosság. Olyan helyiségekben például, ahol az ember által végzett tevékenység zajjal jár, egy vezérlőegység irányíthatja a terem egyes részeiben a világítás mértékét. Ehhez hasonló kényelmi funkció autókban is alkalmazható. Attól függően, hogy melyik ülés irányából érkezett a hang, a jármű központi vezérlőegysége vezérelheti a klíma hőfokát, a szellőzés mértékét, illetve az ülés beállításait is. Betölthet akár társadalmi funkciókat is, mint az idősekre, betegekre való felügyelet.

## 1.1. Lokalizáció területei

Különböző iránydetektálási módszerek már régóta léteznek, ezekben közös, hogy valamilyen objektum vagy ember pozícióját, mozgásának útvonalát kell meghatározni. Mivel

az emberi szem és fül, mint lokalizációs „eszközök” szűk korlátok között képesek csak erre, ezért rengeteg egészségügyi, társadalmi, katonai, kényelmi alkalmazás nem létezhetne a különféle lokalizációs technológiák nélkül.

A különböző lokalizációs módszerek három fő csoportra bonthatók működésük szerint. Az első a képfeldolgozáson alapuló módszerek, melyek tárgyak, emberek azonosítására képesek egy képen, mint például a mai okos telefonokban már létező arcfelismerő programok. A második az elektromágneses hullám alapú iránydetektálás. Ebbe a kategóriába sorolható például a rádiófrekvenciás lokalizáció és a GPS is, amelyeket katonai alkalmazásokban használtak először, majd néhány évtizeddel ezelőtt megjelentek lakossági alkalmazásokban is. A harmadik az akusztikus lokalizációk csoportja, amely további három csoportra, infrahang, ember által is hallható hang és ultrahang alapú iránydetektálásra osztható.

Az alábbiakban bemutatok néhány lokalizációs alkalmazást, melyek katonai, egészségügyi vagy biztonságtechnikai alkalmazhatóságuk miatt már elterjedtek a mindennapokban.

### **1.1.1. Rádiófrekvenciás lokalizáció**

A Radio Detection and Ranging, azaz radar a II. világháború során elterjedt lokalizációs, távolságot meghatározó eszköz. A rádiólokátor az antennából kisugárzott rádiófrekvenciás elektromágneses hullám visszatérésének időpontját figyeli. A vétel általában ugyanazzal az antennával történik, mely folyamatosan körbeforogva térképezi fel a teret. A rádiófrekvenciás hullám kisugárzásainak időpontja olyan, hogy két kibocsátott hullámnyaláb között mindenképpen visszatérjen a jel. A radart alkalmazzák objektumok lokalizálására (hajók, repülőek, tengeralattjárók), mozgó objektumok követésére, illetve a meteorológiában is, felhők mozgásának követésére.

### **1.1.2. Bluetooth, WiFi alapú lokalizációk**

A Bluetooth és a WiFi [4] nagyfrekvencián működő, vezetékmentes kommunikációs protokollok. A helymeghatározáshoz szükséges egy eszköz, általában mobil telefon, amely a saját azonosítóját kisugározva próbál elérni egy detektáló eszközt, ezáltal megtudva a relatív pozícióját. Bluetooth vagy WiFi alapú lokalizációk egyik alkalmazási területe a mobil tulajdonosának követése, így ha például valahol kapcsolódik az internetre, akkor célzott, környékbeli hirdetéseket kaphat.



### **1.1.3. Ultrahang alapú lokalizáció**

Ultrahangoknak [1] nevezzük az olyan hangokat, melyek frekvenciája 20 kHz-nél nagyobb, tehát az ember számára hallható tartományon már kívül esnek. Ultrahang alapú lokalizációt használnak az iparban szilárd testek, az egészségügyben pedig emberi szervek, szövetek vizsgálatára. Ha a vizsgált anyagban hiba van, például valamilyen szennyeződés található egy fémtestben, akkor az eltérő felületek határán az ultrahang hullámok egyik része visszaverődik, másik része pedig továbbhalad.

### **1.1.4. Infrahang alapú lokalizáció**

Infrahangnak [2] hívjuk a 16-20 Hz alatti frekvenciájú hangokat. Alacsony frekvenciája miatt akár néhány 100 km-re is terjedhet, melyet bizonyos állatfajok, például a bálnák ki is használnak nagy távolságú kommunikáció céljából. Infrahangok keletkeznek a természetben számos jelenség közben, ilyenek a tornádók, vulkánkitörések, illetve hólavínák megindulása előtt. Ezáltal megfelelő detektorokkal előre jelezhető a természeti katasztrófák bekövetkezése.

### **1.1.5. Akusztikus lokalizáció**

Az ember által hallható hang frekvenciatartománya 20 Hz és 20 kHz között található. Az ebben a tartományban található hang alapú lokalizáció elsődleges alkalmazásai olyan területeken fordulhatnak elő, ahol valamilyen ember által végzett tevékenységet képes kiváltani. Ilyen például az okos otthoni alkalmazás is, hiszen nem kell megkérnünk valakit, hogy kapcsolja le a villanyt, hanem elég utasítani erre egy vezérlőegységet. A lokalizáció hátrányai közt szerepel, hogy viszonylag kis távolságokban működőképes, a forrás irányának detektálása nem lehetséges néhányszor 10, vagy 100 km-es nagyságrendben, mint például az infrahang esetében.

## 2. Rendszerterv

### 2.1. Célok, követelmények

Szakedolgozatom témája emberek pozíciójának meghatározása beltérben. Céлом az volt, hogy egy olyan széles körben használható alkalmazást készítssek, mely kihasználja az emberi beszédet, mint lokalizációra alkalmas hullámot, ezáltal nem szükséges semmilyen elektromágneses vagy hanghullám előállítás. A félév során az alábbi feladatok kerültek megvalósításra:

- sztereo mikrofonok segítségével végzett mérések alapján az akusztikus lokalizáció megtervezése
- több különböző iránydetektálásra alkalmas módszer megismerése, implementálása
- helymeghatározás megbízhatóságának és pontosságának az alkalmazás számára megfelelő mértékűre javítása
- a használt metódusok fűziójával személyi számítógépen valós idejű működés létrehozása

A félév során elkészült alkalmazásnak számos követelménynek kell megfelelnie. Elegendően pontos iránydetektálást kell megvalósítania mindössze egy mikrofonpár segítségével. A későbbi PC-n való alkalmazhatóság érdekében az algoritmusnak megfelelő pontossággal szükséges detektálnia a forrás irányát körülbelül 10 cm-es mikrofontávolság esetén, hogy külső mikrofonok nélkül használható legyen a program személyi számítógépen vagy például sztereo mikrofonokkal rendelkező multimédiás webkamerán.

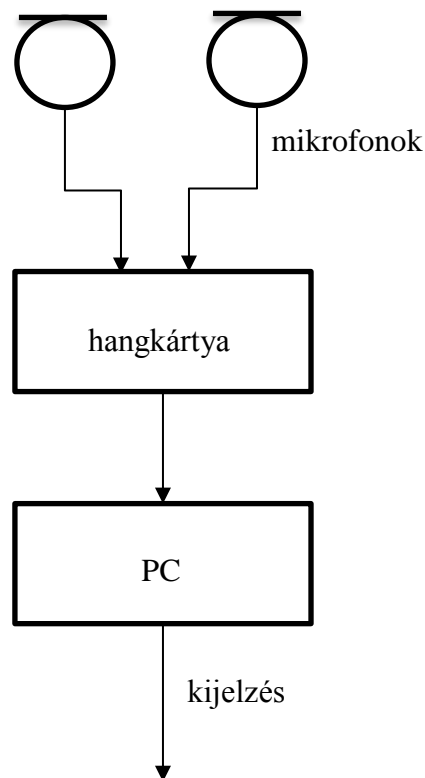


2-1. ábra, sztereo mikrofonokkal rendelkező eszközök [17][18]

A megbízhatóbb működés érdekében több lokalizációs algoritmusnak kell párhuzamosan futnia, hogy fúziójukkal robusztusabb működés legyen elérhető. A beszélő irányának meghatározására mind hangfelvételek által utólag, mind helyben, valós időben is képesnek kell lennie.

## 2.2. Blokkdiagram

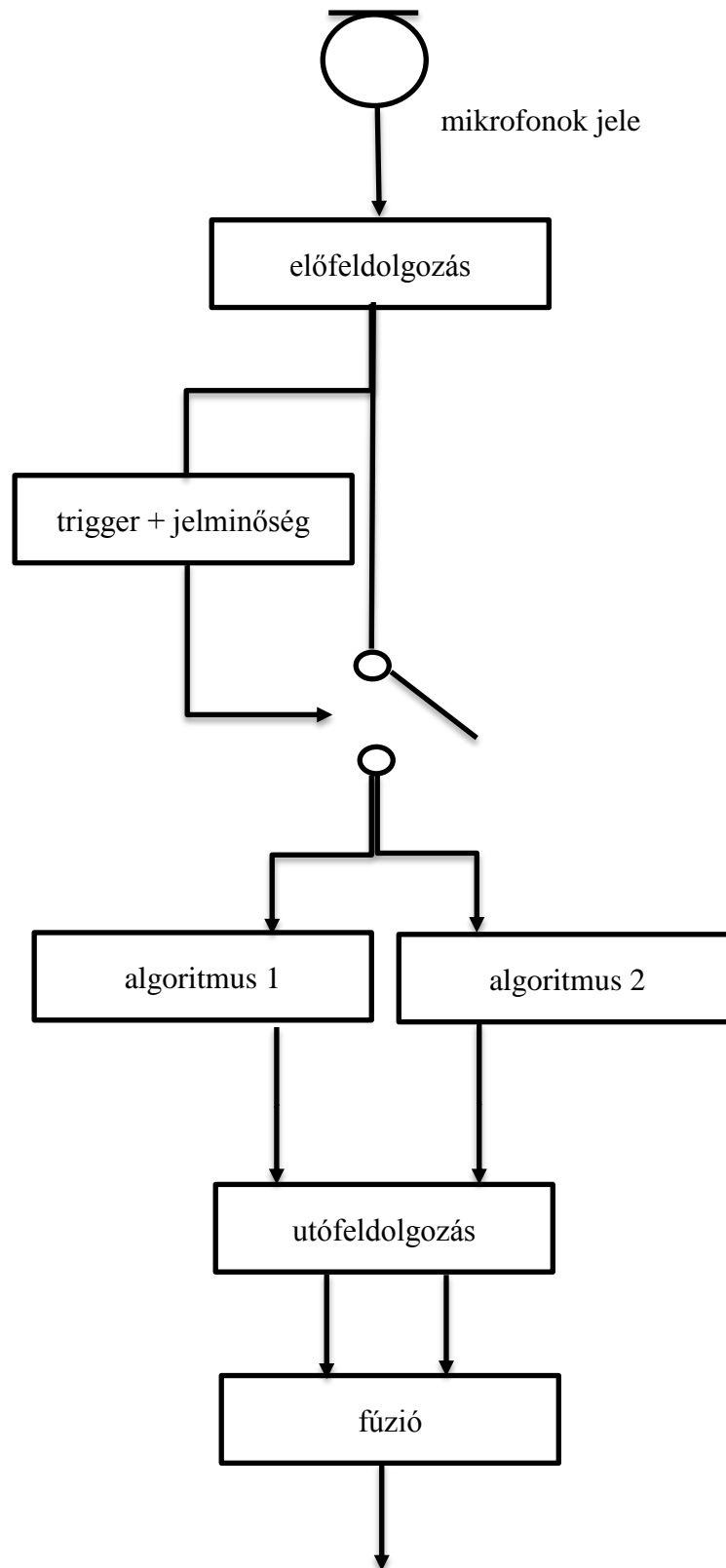
A 2-2. ábrán látható az elkészült alkalmazás blokkdiagramja:



2-2. ábra, alkalmazás blokkdiagramja

Két darab mikrofon szolgáltatja a vett hangot a hangkártya felé. A hangkártya végzi el a jel mintavételezését, majd párhuzamosan továbbítja azt USB kapcsolaton keresztül a személyi számítógép felé. A jelfeldolgozás PC-n zajlik, meghatározott időközönként olvassa ki a program a vett jelet a hangkártyáról, majd azt feldolgozva jeleníti meg az eredményt, tehát a hangforrás pozícióját.

A 2-3. ábrán látható a személyi számítógépen futó algoritmus blokkdiagramja:



2-3. ábra, algoritmus blokkdiagramja

Az **előfeldolgozás** blokk feladata, hogy egyfajta szűrőként funkcionáljon, mellyel a mérés során felvett zajokat szűri ki a mintavételezett jelből. A mikrofonok által vett környezeti zajok befolyásolhatják a helymeghatározás pontosságát, mivel nemcsak a forrás irányából, hanem mindenhonnan érkezik jelteljesítmény a mikrofonok felé.

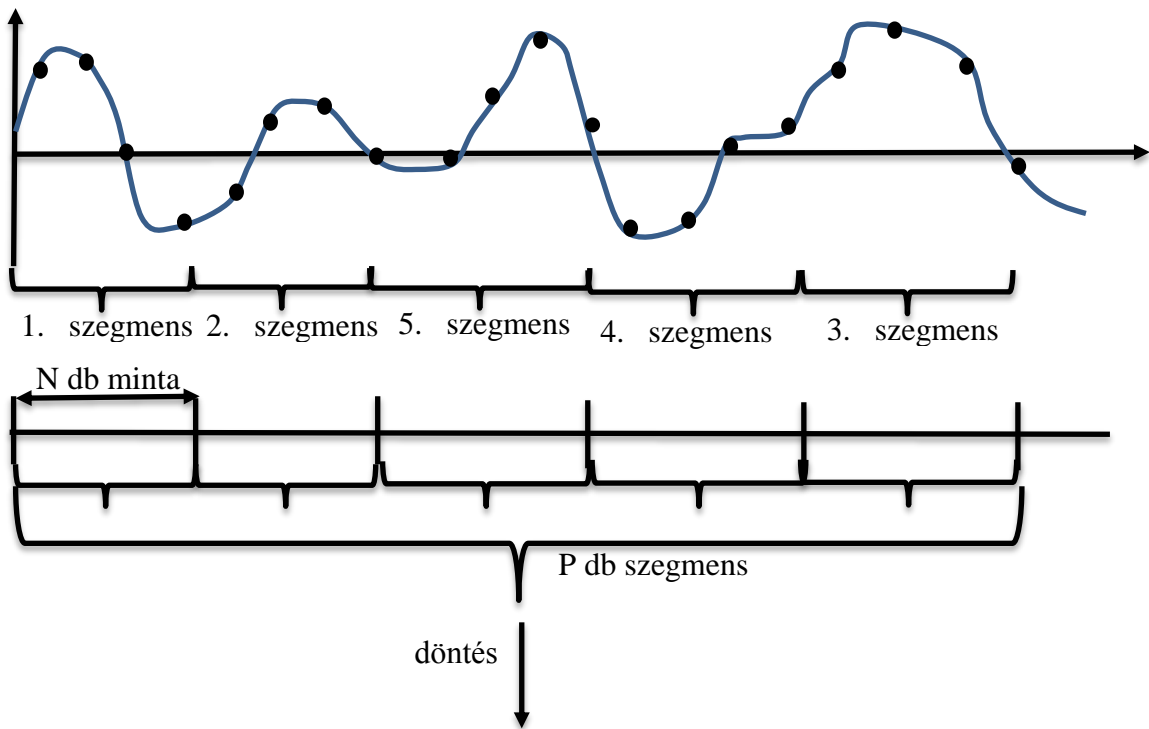
A **triggerelés** és a **jelminőség** között egyfajta „és” kapcsolat áll fenn. Az emberi beszéd meghatározott frekvenciákon jelentősebb teljesítménnyel van jelen, mint az egyéb környezeti zajok. Ezen frekvenciatartományban figyelve egy meghatározott teljesítményszint, triggerszint felett dolgozzuk csak fel a jelet. A jelminőség vizsgálat a felvett jel teljesítményének frekvenciatartománybeli eloszlásából von le következtetéseket, és dönti el, hogy az adott jel megbízható eredményt hordoz-e magában lokalizáció szempontjából. Részletesen a 4.2. fejezetben mutatom be, hogy milyen mérőszám használható a jelminőség jellemzésére.

Az **algoritmus 1** és **algoritmus 2** blokkok fedik le a szakdolgozatom során megvalósított két lokalizációs algoritmust, a korrelációt és a beamforming-ot.

Az **utófeldolgozás** feladata, hogy az algoritmusok által szolgáltatott bizonytalan eredményeket statisztikai módszerek és apriori ismeretek alapján megtisztítsuk a gyanúsan rossz eredményektől. Az utófeldolgozás során outlier kereső algoritmust használtam, illetve figyelembe vettem, hogy feltételezésünk szerint emberi hangot detektálunk, így a forrás nem mozoghat tetszőlegesen gyorsan.

A **fúzió** blokk szemlélteti a két algoritmus eredményeinek utófeldolgozása utáni egyesítését a megbízhatóbb, robusztusabb eredmény érdekében.

Jelfeldolgozás időzítési diagramja a következő ábrán látható:



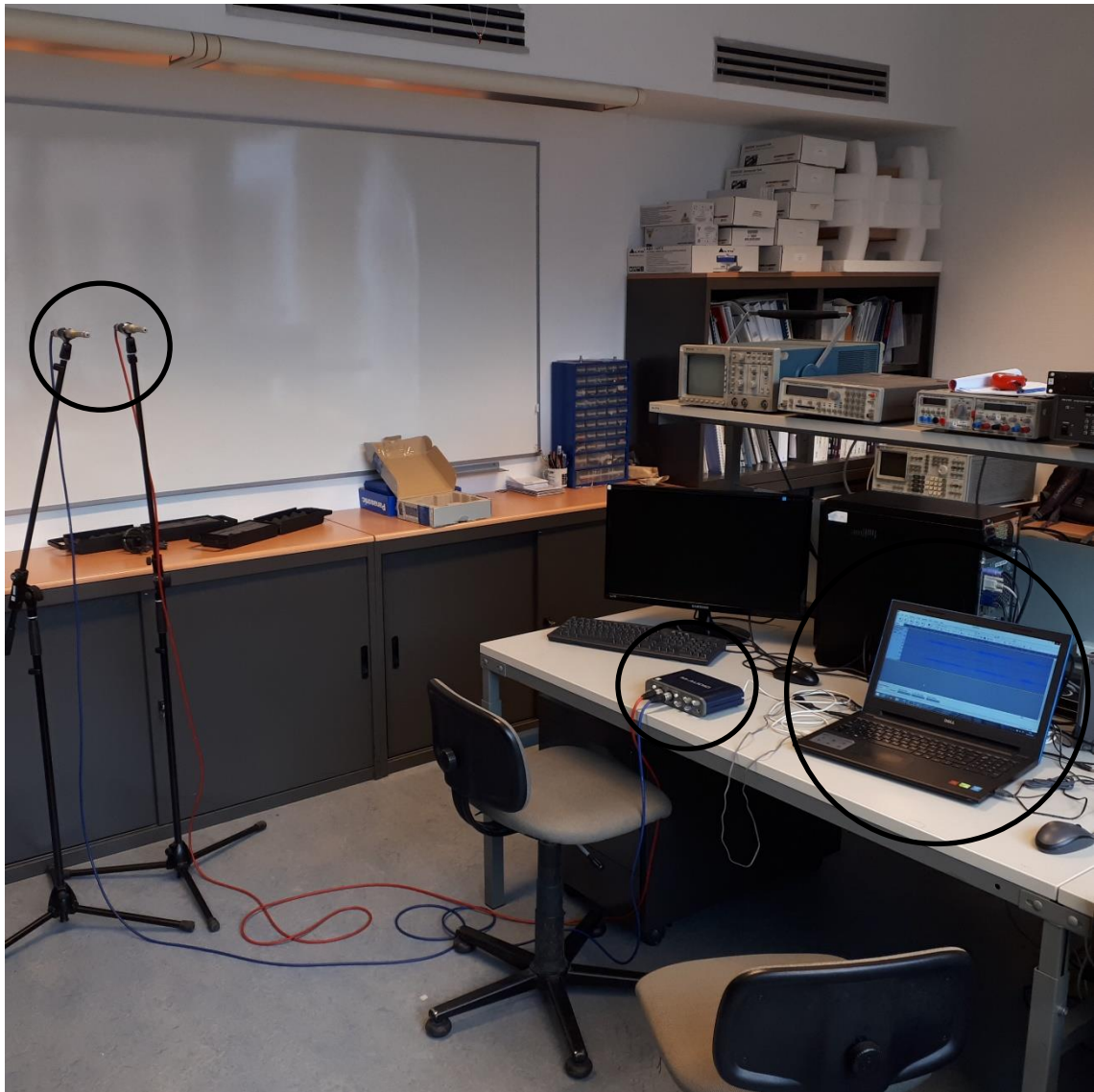
2-4. ábra, jelfeldolgozás időzítési diagramja

Az algoritmus személyi számítógépen való futtatásának következménye, hogy nem lehet erősen valós idejű a program. Az audio lokalizációs módszerek sem egy-egy mintát, hanem több mintából álló sorozatot, szegmenseket igényelnek bemenetként, valamint a személyi számítógépek sem alkalmasak erősen valós idejű algoritmusok futtatására. Tehát első lépésként a beérkező jelet mintavételezzük, majd szegmentáljuk. Az egymás után érkező szegmenseken pedig végrehajtjuk a jelfeldolgozást. A program robusztusságának növelése érdekében lehetőség van a szomszédos szegmensek közötti átlapolódás használatára. Ennek mértéke általában 50-80 %-ig terjed (lásd 4.1.1. fejezet).

### 2.3. Rendszer felépítése

A 2-2. ábrán látható fizikai rendszer megvalósításának eszközei a következők:

- 2 darab mikrofon (Behringer ECM-8000), ismert iránykarakterisztikával [5].
- külső hangkártya (M-Audio Fast Track Pro), mely a mikrofonok által vett jelet pontos időbélyeggel, azaz párhuzamosan továbbítja [6]
- személyi számítógép, amelyen a jelfeldolgozó algoritmus és a mérésekhez szükséges hangrögzítő program (Audacity 1.3.13 verzió) [7] futtat



2-5. ábra, mérési elrendezés illusztrációja

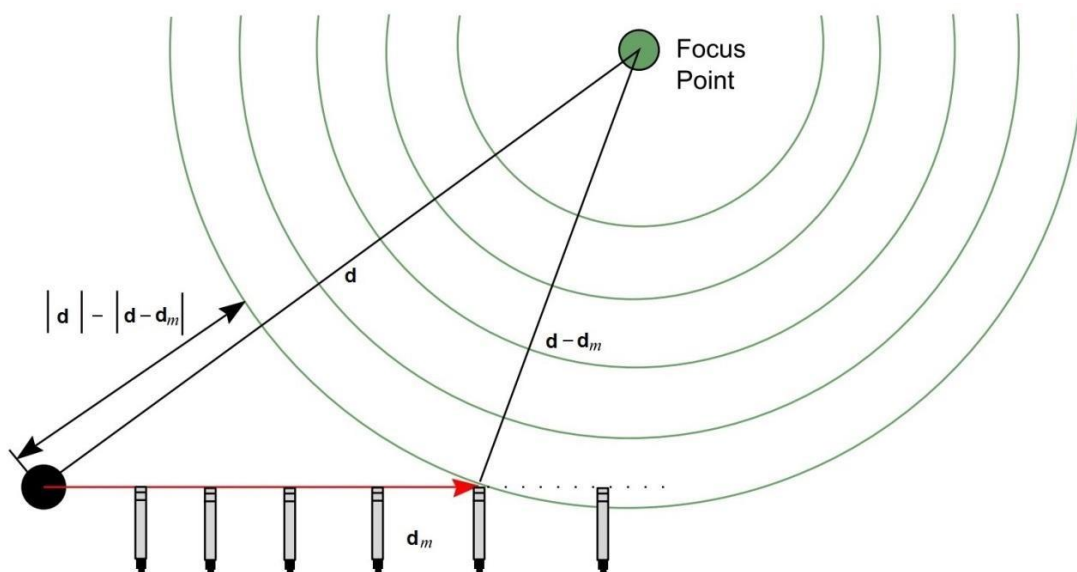
A 2-5. ábra illusztrál egy lehetséges mérési elrendezést, amelyet a mérések során használtam. A két darab mikrofonnal végzett hangrögzítés következménye, hogy csupán irányt lehet meghatározni, azaz egy választott egyenes mentén lehet elhelyezni a forrást. A pozíció meghatározásához síkban minimum 3, térben pedig minimum 4 mikrofonra lenne szükség. Eltekintettem attól, hogy a mikrofonok mögött legyen az akusztikus forrás, mivel a mikrofonok gömbkarakterisztikája olyan, hogy ez a lokalizáció pontosságát valószínűleg nem befolyásolja lényegesen, valamint a későbbi felhasználás sem indokolja az ilyen irányú tesztek, méréseket. A mikrofonpár közti szakasz felezőmerőlegeséhez képest mindkét irányban  $90^\circ$ -os sávot vizsgálok, ezzel lefedve a teljes vizsgálni kívánt tartományt. A mikrofonállvány lehetővé tette, hogy körülbelül az emberi száj, azaz a hangforrás magasságában vegyem fel a hangot.

A jelfeldolgozás implementálására a Matlab R2018a program került kiválasztásra. A valós idejű megvalósításhoz fontos, hogy a mérések során használt hangkártya által szolgáltatott adatot megfelelő idő alatt és megfelelő időpontban lehessen olvasni. Erre kínálja megoldásként a Matlab R2018a szoftverhez a Real-Time Audio Processing Toolbox-ot a Mathworks cég.



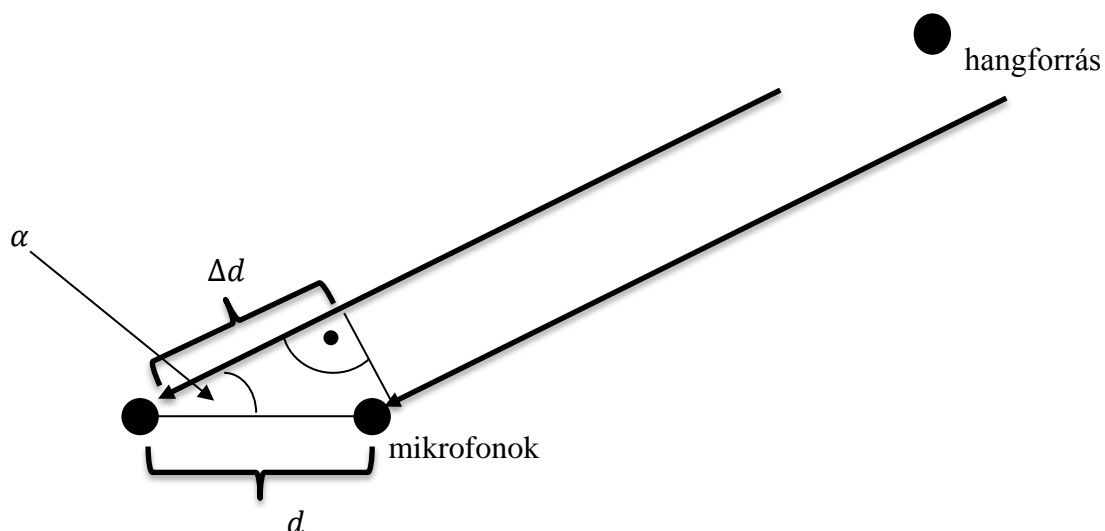
### 3. Akusztikus lokalizáció

Az akusztikus lokalizáció során a hangforrás pozíciója a forrás saját maga által keltett hangja alapján határozható meg, míg más alkalmazásokban (lásd 1.1.3. fejezet) a hullám detektálására szánt eszköz bocsátja ki magát a hullámot is. Tehát audio helymeghatározás során csak a mikrofonok által vett jelek egymáshoz képesti viszonya vizsgálható. A hang a keletkezési helyétől eltérő nagyságú utat jár be, míg elér a mikrofonokhoz. Így ha meghatározzuk, hogy mekkora késleltetéssel érkeztek a mikrofonokhoz a jelek egymáshoz képest, akkor a hang sebességét ismerve kiszámolható a kiindulási pozíció.



3-1. ábra, hang koncentrikus körök menti terjedése [9]

A 3-1. ábrán látható, hogy a hangforrásból koncentrikus körök mentén terjed a hang síkban. A mikrofonokhoz érkező jelutak közti különbség meghatározása általános esetben nehézkes, ezért a távolféri vizsgálatok során a hang terjedését leegyszerűsítjük síkhullámmá [8]. Ehhez az szükséges, hogy elegendően távol legyünk a forrástól.



3-2. ábra, síkhullám terjedése

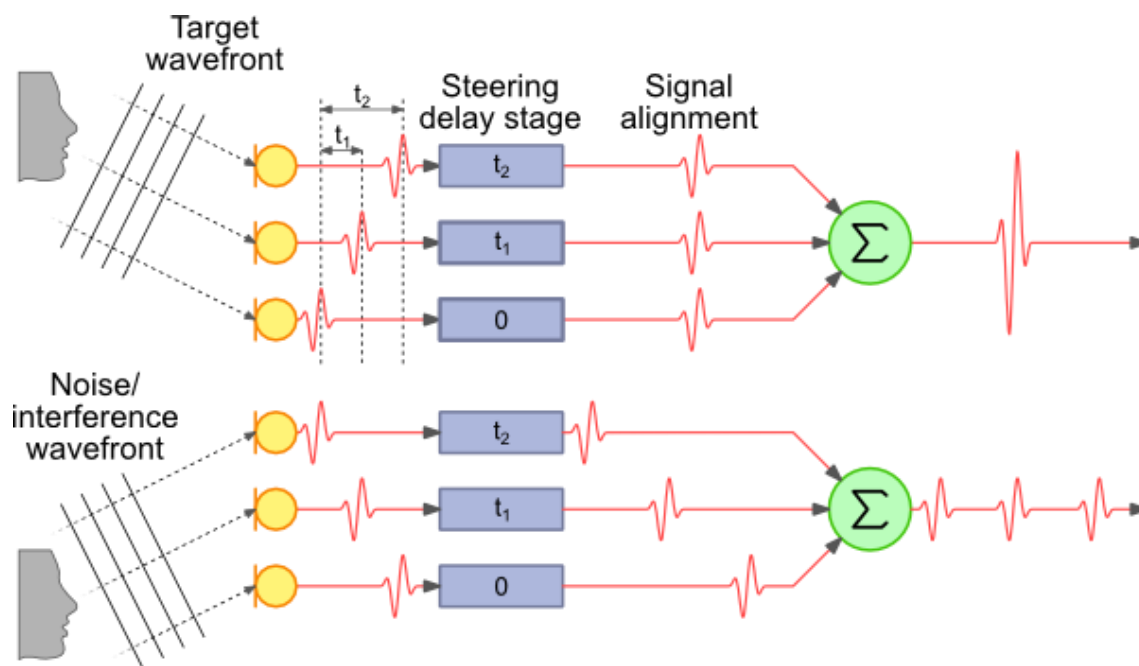
A távotérben síkhulláma egyszerűsített hang terjedése a 3-2. ábrán látható. Ahogy az illusztráción is látható, kellően messze lévő forrás esetén a mikrofonok által vett jelek párhuzamosak. A  $\Delta d$  jelutak közti különbség meghatározható a vett hangok közti időkésltetés ( $\Delta t$ ), valamint a hang terjedési sebességének ( $c$ ) ismeretében.

$$\Delta d = \Delta t \cdot c \quad (3.1)$$

A forrás irányának meghatározásához a kiszámolt  $\Delta d$ , valamint a mikrofonok közti ismert  $d$  távolság szükséges.

$$\alpha = \cos^{-1}\left(\frac{\Delta d}{d}\right) \quad (3.2)$$

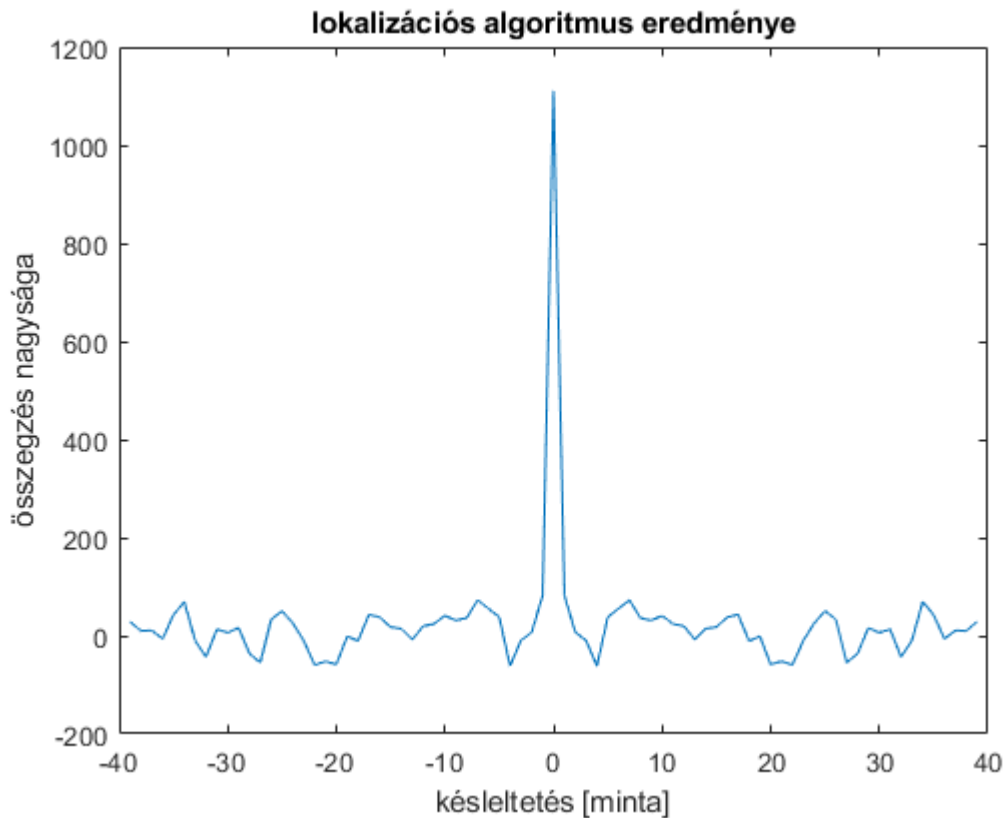
Tehát az akusztikus lokalizáció során a mikrofonokhoz érkező hanghullámok közötti időkésltetés megtalálása az algoritmusok feladata. Ennek egy lehetséges megoldása, hogy a kettő vagy több összehasonlítandó szegmenset időben egymáshoz képest késleltetik, majd valamilyen, az adott módszertől függő módon összegzik a jeleket, ahogyan az a következő ábrán látható:



3-3. ábra, mikrofonokhoz érkező jelek késleltetése, összegzése [10]

Minden lehetséges iránynak megfelelő késleltetés esetén meghatározzuk az összegzett jel teljesítményét, és azt feltételezzük, hogy a legnagyobb eredő teljesítményt akkor kapjuk, ha a beállított késleltetések a forrás valódi pozíciójának felelnek meg. Ha a késleltetés maximális, akkor a hangforrás pozíciója a mikrofonokkal egy vonalban helyezkedik el, azaz a mikrofonokhoz érkező jelek által megtett utak közti különbség a mikrofonok közti távolsággal egyezik meg.

Az alábbi ábra egy tipikus mérési eredményt mutat, ahol a különböző késleltetéssel összegzett jelek eredő teljesítményét ábrázoljuk az eltolás függvényében. Az x tengelyen a jelek közötti eltolás szerepel mintában megadva, az y tengelyen pedig az eltolt jelek összegzésével adott eredő jelteljesítmény. Látható, hogy a jelek közötti késleltetésnek megfelelő helyen egy domináns csúcs jelenik meg, ahogyan elméletileg is várjuk.



3-4. ábra, lokalizációs algoritmus eredménye ideális esetben

A csúcs detektálása során, annak helye határozza meg a késleltetés nagyságát. Ezen késleltetés értéke kvantált, hiszen a jel mintavételezett, eltolása egyszerű esetekben csak mintánként lehetséges. Ha a mintavételi frekvencia  $F_s$ , akkor a mintavételi idő:

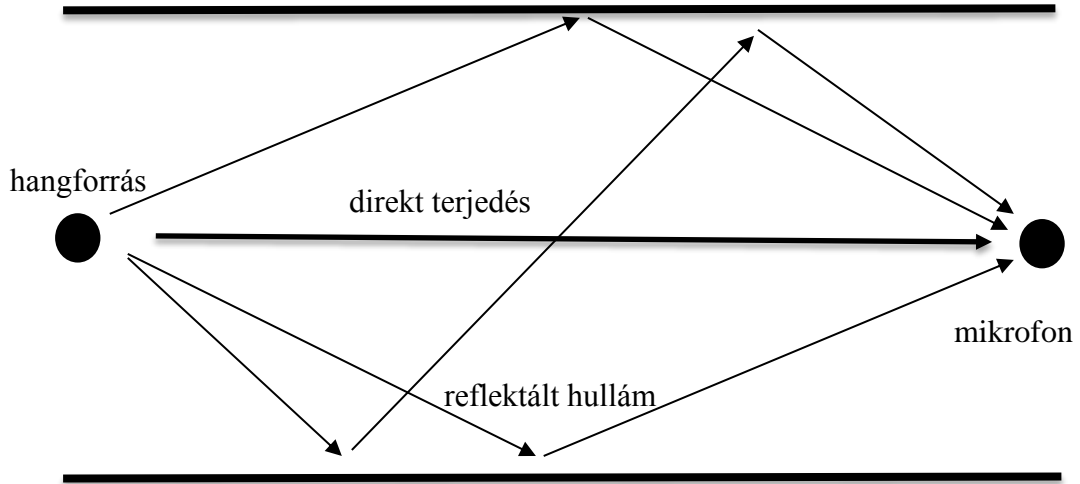
$$T_s = \frac{1}{F_s} \quad (3.3)$$

Azaz, ha a mikrofonok által vett jelek között 5 minta a késleltetés, akkor az a valóságban  $5 \cdot T_s$  időt jelent. Azonban létezik olyan lokalizációs módszer, ahol a késleltetés felbontásának nagysága finomítható, lásd 3.2. fejezet.

Valós esetben előfordulnak olyan nem ideális hatások, mint a mindenhol jelenlévő zaj, vagy a beltéri helyszín miatti többutas terjedés. A hang miután elhagyta a forrást, nem

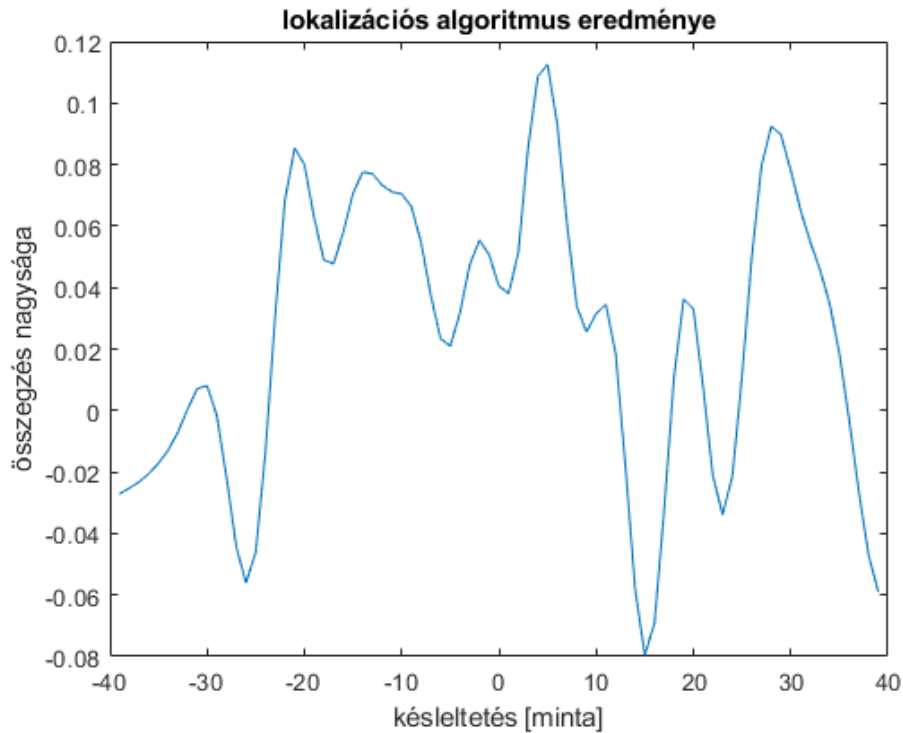
csak direkt, egyenes úton juthat el a mikrofonokhoz, hanem a terem falairól, a benne található tárgyról reflektálódva is.

Ezt a jelenséget mutatja be a következő ábra:



3-5. ábra, többutas terjedés

A 3-5. ábra alapján a mikrofonokhoz nem csak a forrás irányából érkezik meghatározó jelteljesítmény, hanem máshonnan is. Ez a jelenség a lokalizációs algoritmusok kimenetét a következőképpen módosítja:



3-6. ábra, lokalizációs algoritmus eredménye többutas terjedés esetén

A 3-6. ábrán látható grafikonról leolvasható, hogy a maximális csúcs körülbelül 5 mintányi késleltetést határoz meg, holott a valódi késleltetés jelen esetben 30 minta volt. Ugyanakkor 30 mintányi késleltetés környékén is található egy csúcs, vagyis a többutas terjedésnek köszönhetően nem csak a valódi irányból, hanem máshonnan is jelentős jel-teljesítmény érkezett a mikrofonok felé. Az utófeldolgozás (4.4. fejezet) során lehetőség lesz az eredmény utólagos javítására, másik csúcs detektálására.

Néhány bekezdéssel korábban említettem, hogy a jelek közti késleltetés meghatározása kvantált, mely a jel mintavételezéséből adódik. A (3.1), (3.2), (3.3) egyenletek és a 3-2. ábra alapján a forrás irányának azonosítása a következő egyenletek alapján történik (a forrás iránya a jelút vonala és a mikrofonok közti egyenes által bezárt szög kiegészítő szöge):

$$\alpha' = 90^\circ - \cos^{-1}\left(\frac{\Delta t \cdot c}{d}\right) \quad (3.4)$$

$$\Delta t = N_d \cdot \frac{1}{F_s} \quad (3.5)$$

Ahol:

$N_d$  – késleltetés mintában számlálva

Az  $\alpha'$  szög csak meghatározott értékeket vehet fel, mivel  $N_d$  csak egész szám lehet. A mikrofonok közti távolság növelésével a felbontás csökkenthető, mert az  $N_d$  által felvehető értékek száma is növekszik. A felbontás más módszer segítségével is csökkenthető, melyet a 3.2. fejezetben ismertetek.

A következő alfejezetekben néhány lokalizációs módszert ismertetek.

### 3.1. Korreláció

A lokalizációs algoritmusok közül az egyik legismertebb és gyakran használt módszer a korreláció [8]. Korreláció során az eltolást mintánként végezzük időtartományban, összegzés helyett pedig szorzással vizsgáljuk a hasonlóságot. Az algoritmus egyenlete a következő:

$$R_{xy}(n) = \sum_{i=-\infty}^{+\infty} x(i) \cdot y(i+n) \quad (3.6)$$

A (3.6) egyenlet megadja az  $n$  időpontban az  $x(i)$  és  $y(i)$  függvények keresztkorrelációját, ahol  $n$  az eltolás nagysága. Ha a két függvény független egymástól, akkor a szorzatuk egyenlő eséllyel lesz negatív vagy pozitív, ezáltal az összeg valószínűleg 0 közelében marad.

Autokorrelációról beszélünk abban az esetben, ha a jel önmagával vett korrelációját számítjuk:

$$R_{xx}(n) = \sum_{i=-\infty}^{+\infty} x(i) \cdot x(i+n) \quad (3.7)$$

Általában ez egy olyan  $R_{xx}(n)$  függvényt határoz meg, melynek maximuma 0 késleltetés esetén lép fel, hiszen egy jel általában önmagára hasonlít a legjobban, nem más jelekre. Konkrét lokalizációs alkalmazás esetén a két mikrofon által vett  $x_1(i)$  és  $x_2(i)$  jelek közül az egyiket időben késleltetjük, és a másik mikrofon által vett jellel vesszük a korrelációját.

$$R_{xx}(n) = \sum_{i=-\infty}^{+\infty} x_1(i) \cdot x_2(i-n) \quad (3.8)$$

Ahol:

$$x_2(i) = x_1(i - T_d) \quad (3.9)$$

$$T_d = N_d \cdot \frac{1}{F_s} \quad (3.10)$$

$T_d$  - a két jel közti valós késleltetés.

A cél az, hogy meghatározzuk a  $T_d$  késleltetés nagyságát, majd a (3.4) és a (3.5) egyenletek segítségével a forrás iránya kiszámolható.

Korreláció során a csúcsok felismerésének pontosságát ideális esetben is rontja a mintavételi frekvencia. Az  $1/F_s = T_s$  időközönként mintavételezett jel abszolút hibát okoz a forrás detektálásának pontosságában, mely csak abban az esetben fontos, ha ez egy nagyságrendbe esik magának az algoritmusnak a hibájával. Tehát a korreláció felbontása a mintavételi frekvencia növelésével, illetve a (3.4) egyenlet alapján a mikrofonok közti távolság növelésével érhető el.

## 3.2. Beamforming

A Beamforming módszer a jelfeldolgozást frekvenciatartományban végzi el a Fourier-transzformáció segítségével. A Fourier-transzformáció eltolási tétele szerint, ami időtartományban időben eltolás, az frekvenciatartományban szögelfordulás:

$$X(\omega) = F\{x(t)\} \quad (3.11)$$

$$F\{x(t - t_0)\} = X(\omega) \cdot e^{-j\omega t_0} \quad (3.12)$$

A (3.12) egyenlet alapján az eltolást nem mintánként időtartományban, hanem frekvenciatartományban az egyes frekvenciakomponensek vektorainak forgatásával végezzük el. A módszer előnye, hogy bármilyen kicsi felbontás használható, az egyetlen korlátot az ezzel járó számítási kapacitás növekedése jelenti.

Beamforming esetén az összegzés tényleges összeadást jelent. A mikrofonok által vett egyik jelet referencia jelnek tekintjük, a másik jel esetében pedig az egyes frekvenciakomponensekhez tartozó vektorokat elforgatjuk. Minden forgatás után a két jel frekvenciakomponenseihez tartozó teljesítményt összeadjuk. A beamforming algoritmus egyenletei két bemenő jelre:

$$X_1[k] = fft(x_1(t)) \quad (3.13)$$

$$X_{2,n}[k] = fft(x_2(t)) \cdot e^{j \cdot n \cdot 2\pi \cdot \frac{k-1}{N}} \quad (3.14)$$

$$P_n = \sum_{k=1}^N |X_1[k] + X_{2,n}[k]|^2 \quad (3.15)$$

Ahol:

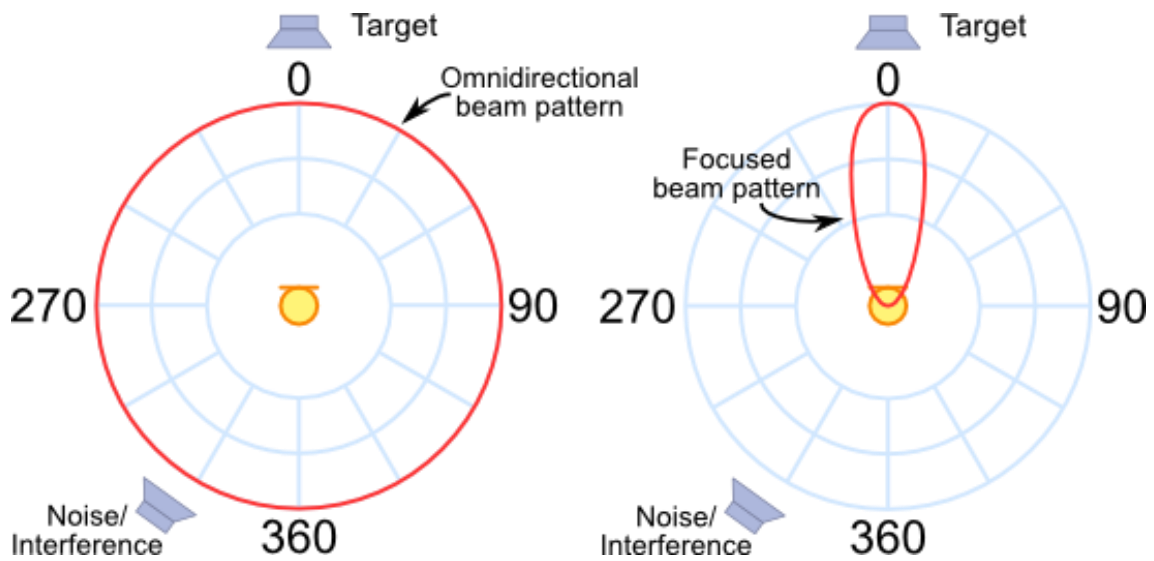
$N$  – szegmens hossza

$n$  – késleltetés időtartományban

A kapott eredményt grafikonon ábrázolva a korrelációhoz hasonlóan általános esetben a legnagyobb csúcs határozza meg a forrás pozícióját.



A módszer hasonló ahhoz, mintha egy pontra fókuszálnánk rá a térben a mikrofonok segítségével, ezt illusztrálja a következő ábra:



3-7. ábra, beamforming algoritmus működése [10]

Az 3-7. ábra bal oldalán található rajz illusztrálja egy darab mikrofon vételi karakteristikáját, a jobb oldali pedig a sztereo mikrofonok által létrehozott karakteristikát. A vett jel frekvenciakomponenseinek vektorainak forgatásával azt a hatást érjük el, mintha a mikrofontömb forogna. A jelenség ahhoz hasonló, mint amikor egy mikrofontömb folyamatos körbe forgatásával térképezzük fel a teljes teret, a különbség, hogy ebben az esetben ez digitálisan kerül megvalósításra, a mikrofontömb a valóságban mozdulatlan marad.

## 4. Megvalósított rendszer

A 3. fejezetben tárgyalt algoritmusok lokalizációs pontossága ideális vagy közel ideális esetekben megfelelő, a valós környezetben történő alkalmazásuk során viszont működésük nem mindig megbízható. Ennek okát a korábbi fejezetek során tárgyaltam, ilyen jelenség például a hang reflektálódása a terem falán és a benne található tárgyakon. Ezen jelenségek hatásainak kiküszöbölése érdekében elő- és utófeldolgozást alkalmazok, valamint a hasznos szegmensek szűrése érdekében csak a triggerszintet meghaladó szegmenseken végzem el a jelfeldolgozást. A triggerelés után a szegmenseken jelminőség vizsgálatot végzek el. Hiába hordoz egy szegmens elegendő nagyságú teljesítményt, bizonyos esetekben a jel tulajdonságai alapján megbízhatatlannak mondható lokalizáció szempontjából. Ezen kívül a program által használt számos paraméter beállítására is sor került. Ezen paraméterek nagysága erősen használatfüggő, ezért statisztikai vizsgálatokat készítettem, hogy a jelen alkalmazás számára milyen értékek, tartományok a legmegfelelőbbek egy-egy paraméter számára.

A teljes alkalmazásban a paraméterek száma túlságosan nagy ahhoz, hogy egyetlen statisztikai vizsgálat során állítsam be a paraméterek ideális értékét. Ezért a 2-3. ábra blokk-diagramja szerint blokkokra osztottam a jelfeldolgozást. A jelfeldolgozás menetével azonos sorrendben hangoltam be az egyes blokkokban található paraméterek ideális értékeit. Azon blokkok esetében, ahol kis számú paraméter található, kézzel hangoltam a paraméterek értékeit, ellenkező esetben statisztikai vizsgálatokat végeztem. Ha egy blokk belsejében 3-4 paraméter van, azok kézzel történő hangolása nehézkes, előfordulhat, hogy nem a legideálisabb paraméterértékeket határozzuk meg, hanem azok csak egy lokális, az egyes paraméterek közeli értéktartományaiban ideálisak. A vizsgálatokat ugyancsak idő hiányában egy, a mérési adatbázisból kiválasztott, megfelelően hosszú hangfájlon végeztem el. Minden egyes lefutás során az egyes szegmensek alapján detektált jelútkülönbségek, azaz a mikrofonok által vett jelek forrástól számított megtett útjai közötti különbség szerint osztályoztam. Az első osztályba került egy szegmens akkor, ha a detektált jelútkülönbség  $\pm 10\%$ -os hibahatáron belül volt a valódi jelútkülönbséghez képest. Az osztályozás skáláját logaritmikus jellegűre állítottam, azaz a következő osztályokba kerültek azok a szegmensek, ahol a hiba  $\pm 30\%$ -os,  $\pm 60\%$ -os,  $\pm 100\%$ -os,

$\pm 150\%$ -os,  $\pm 210\%$ -os,  $\pm 280\%$ -os, és ezen kívül eső volt. A cél az volt, hogy a paraméterek egy választott értéke mellett a szegmensek minél nagyobb része kerüljön az első osztályba, és minél kevesebb az egyre rosszabb osztályokba.

Az adatok olvasását a program számára az `audioDeviceReader()` Matlab paranccsal végzem el. A függvény fontosabb bemeneti paraméterei többek között az olvasandó device neve, a mintavételi frekvencia nagysága (44100 Hz), a szegmensek hossza, valamint az is, hogy milyen adattípusban legyenek eltárolva az adatok a programban.

A következő alfejezetekben bemutatom a megvalósított rendszer blokkjait, azok működését, és hozzájárulásukat az iránydetektálás javításához.

## **4.1. Előfeldolgozás**

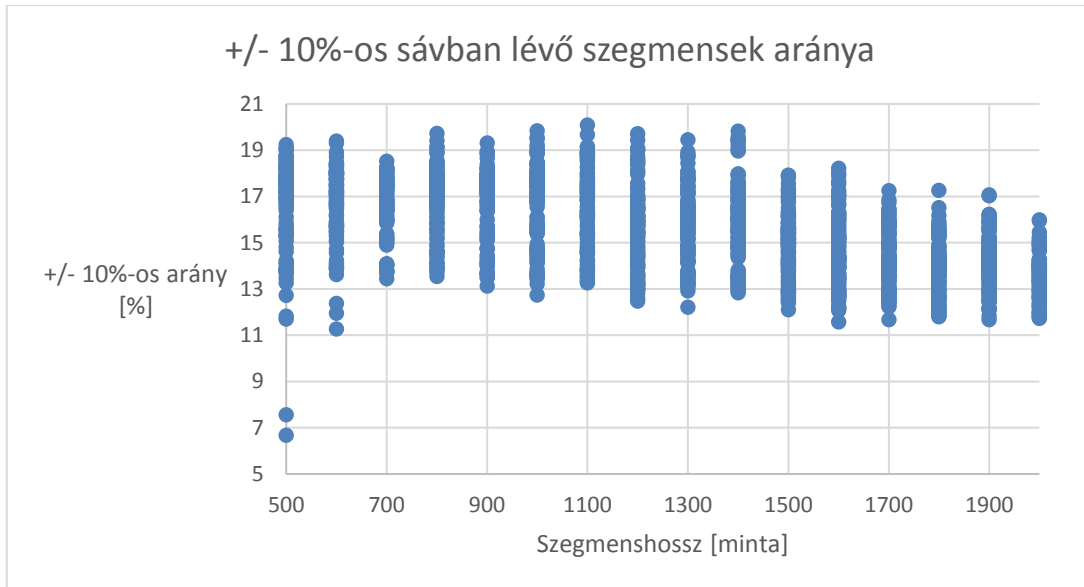
Mielőtt bármiféle vizsgálat, illetve iránydetektáló algoritmus feldolgozná a jelet, szükséges előfeldolgozást végezni. Előfeldolgozás során végzem el a mintavételezett jel szegmentálását. Itt valósul meg a szűrés is, amellyel a nemkívánatos levegőben terjedő zajokat szűröm ki, ezáltal pontosabbá és megbízhatóbbá téve az iránydetektálást.

### **4.1.1. Szegmentálás**

A jelfeldolgozás során az első lépés a jel szegmentálása a 2.2. fejezetben tárgyalt okokból. A használt lokalizációs algoritmusok nem egy-egy mintát, hanem egy mintákból álló sorozatot várnak bemenetükként. A robosztusabb működés érdekében, illetve azért, hogy több szegmensen lehessen dolgozni, ezáltal gyakrabban álljon elő új eredmény, a szegmensek átlapolódnak egymással. Különböző alkalmazásokban más-más mértékű átlapolódás, és különböző hosszúságú szegmensek működnek megfelelően. Időkritikus alkalmazásokban kevesebb mintából álló szegmenseket és nagyobb mértékű átlapolódást használnak, viszont olyan alkalmazásokban, ahol a jel különböző tulajdonságait vizsgálják, hosszabb szegmensekkel, és nem feltétlenül nagy arányú átlapolódással dolgoznak. A jelen alkalmazás során a szegmensek ideális hosszának meghatározásához statisztikai vizsgálatokat végeztem. Az átlapolódás megfelelő arányára is futattam tesztek, de ennek mértékét az elkészült alkalmazásban korlátozza a valós idejű működés kritériuma (5.2. fejezet). 80%-os átlapolódás esetén az algoritmusnak ötöd annyi ideje van a lefutásra, mint átlapolódás nélkül, viszont ugyanakkora adatmennyiséget kell feldolgoznia. A következőkben látható tesztek futtatása során a megvalósított rendszer részei közül csak

a 4.2. fejezetben található triggerelés volt megvalósítva, hogy a tesztek eredményét minél kevésbé befolyásolja a jelfeldolgozási lánc többi eleme.

A szegmens ideális hosszának meghatározásához használt futtatási eredmények a következő ábrán láthatók:

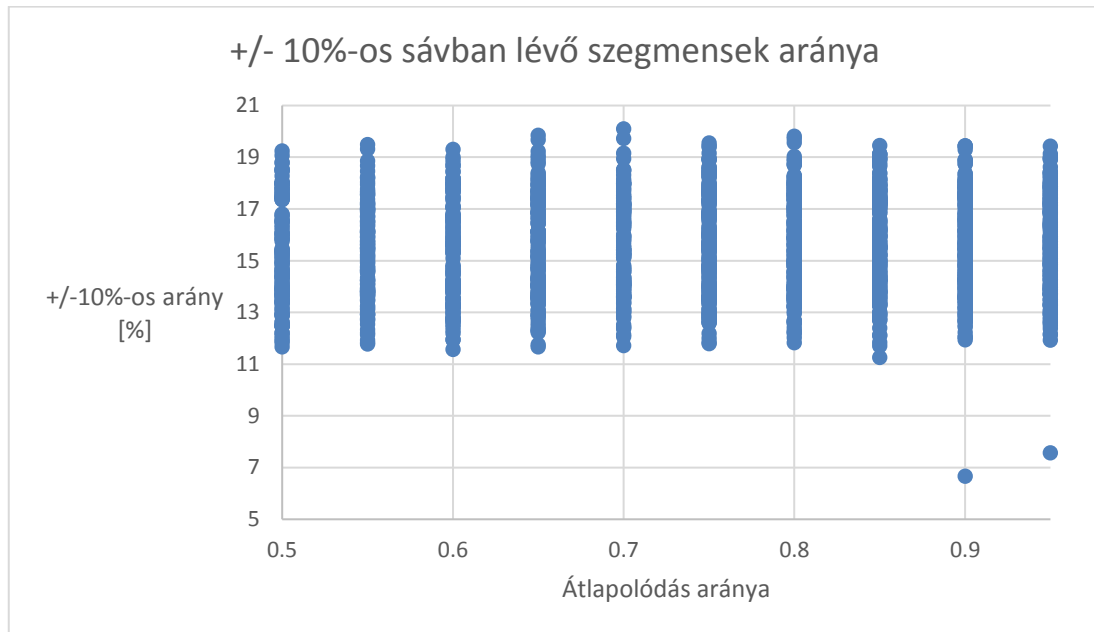


4-1. ábra, szegmens ideális hossza

A grafikon x tengelyén van ábrázolva a szegmens hossza, míg az y tengelyen az első osztályba kerülő szegmensek aránya. A tesztek futtatása során egymásba ágyazott ciklusok segítségével változtattam a szegmens hosszát, az átlapolódás mértékét és egyéb paramétereket, ezért található több eredmény minden szegmenshossz esetén. A fenti grafikonon, és a további ehhez hasonló grafikonokon az éppen vizsgált paraméter, jelen esetben a szegmens hosszának függvényében adják meg teljes hangfájltra azon triggerszint feletti szegmensek arányát, amelyek az első osztályba kerültek.

Láthatóan a szegmenshossz 800 és 1400 minta között a leginkább megfelelő, a megvalósított rendszerben 1100-as értéket használtam. Kisebb hossz esetén bizonyos átlapolódások esetén jóval pontatlanabbá válik az iránydetektálás. Hosszabb szegmensek esetén pedig néhány százalékot romlik az első osztályban lévő szegmensek maximális aránya.

Az átlapolódás ideális arányának meghatározása a következő grafikon segítségével történt:



4-2. ábra, átlapolódás ideális mértéke

Az ábrán a korábban tárgyaltak alapján az átlapolódás mértékének függvényében ábrázoltam az első osztályba kerülő szegmensek arányát, míg a változó paraméter ebben az esetben a szegmens hossza volt. A diagramról leolvasható, hogy az átlapolódás mértéke nem befolyásolja jelentősen az iránydetektálás pontosságát.

#### 4.1.2. Szűrés

Előfeldolgozás során a második lépés a mintavételezett jel szűrése időtartományban. Az emberi beszéd meghatározott frekvenciatartományban tartalmaz a lokalizáció számára hasznos információt. Ha ezen a tartományon kívül nagy teljesítményt tartalmaz a jel, akkor triggerszint feletti jelteljesítményt tartalmazva a lokalizáció eredményét ronthatja. A szűrés által a különböző környezeti zajok csak akkor zavarják az iránydetektálás pontosságát, ha frekvenciatartományuk egybeesik az áteresztő tartománnyal. A szűrő tervezését a Matlab beépített `firls` függvényével végeztem. A parancs segítségével egy lineáris fázisú szűrő tervezhető, amely a legkisebb négyzetek módszere szerint minimalizálja az ideális amplitúdó-karakterisztika és a létrejövő amplitúdó-karakterisztika közti különbséget. A szűrő tervezéséhez használt paraméterek, illetve a szűrő karakterisztikája a következő alfejezetben látható.

#### 4.1.2.1. Szűrő paraméterei

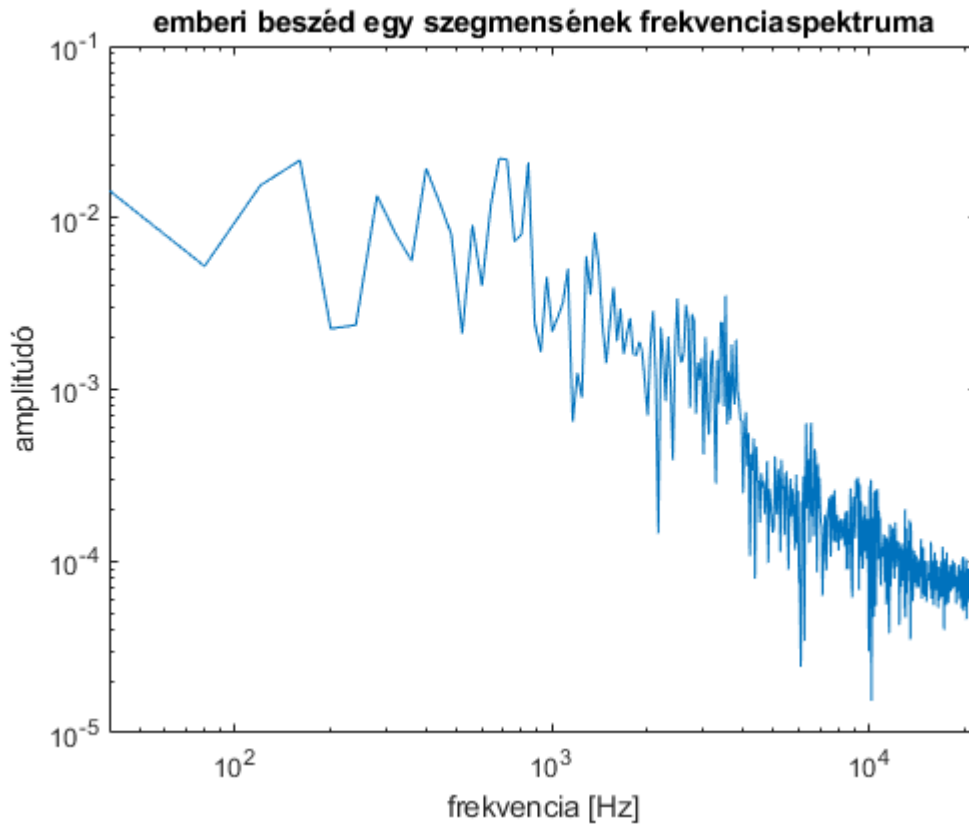
Az ideális paraméterek megtalálása érdekében a 4.1.1. fejezethez hasonlóan statisztikai vizsgálatokat végeztem. A szűrő tervezéséhez szükséges függvény paraméterei:

$b = \text{firls}(n, f, a, w)$ , ahol:

- $n$ : szűrő pontszáma
- $f$ : sáváteresztő és sávzáró tartományok határai vektorban megadva
- $a$ : az  $f$  által meghatározott tartományok áteresztő képessége vektorként megadva
- $w$ : súlyozáshoz használt értékek, amellyel a különböző tartományok ideálishoz képesti távolsága súlyozásra kerül
- $b$ : az elkészített szűrő

A szűrő pontszáma minél nagyobb, a szűrő annál inkább közelíti az ideális karakterisztikát. Túl kicsi pontszám esetén az áteresztő és záró tartományok határain túl nagy kiemelés léphet fel, amely jelenség a pozíció meghatározását pontatlanná teheti. A pontszám nagyságát befolyásolja a jel szűréséhez szükséges idő. Mivel  $n = 1000$  pontszámúnál nagyobb szűrő jelentősen lelassította a program működését, ezért a pontszámot 1000-nek választottam. A szűrő sáváteresztő jellegű, egyrészt azért, mert 4-5 kHz felett már elenyésző jelteljesítményt tartalmaz az emberi beszéd, másrészt pedig kicsi, néhány 100 Hz frekvencián, illetve alatta lévő zajok befolyásolhatják a lokalizáció pontosságát, mivel kisebb frekvenciákon a zaj is nagyobb teljesítménnyel rendelkezik.

A következő ábrán látható a statisztikai vizsgálatok során használt hangminta egy szegmensének frekvenciaspektruma:

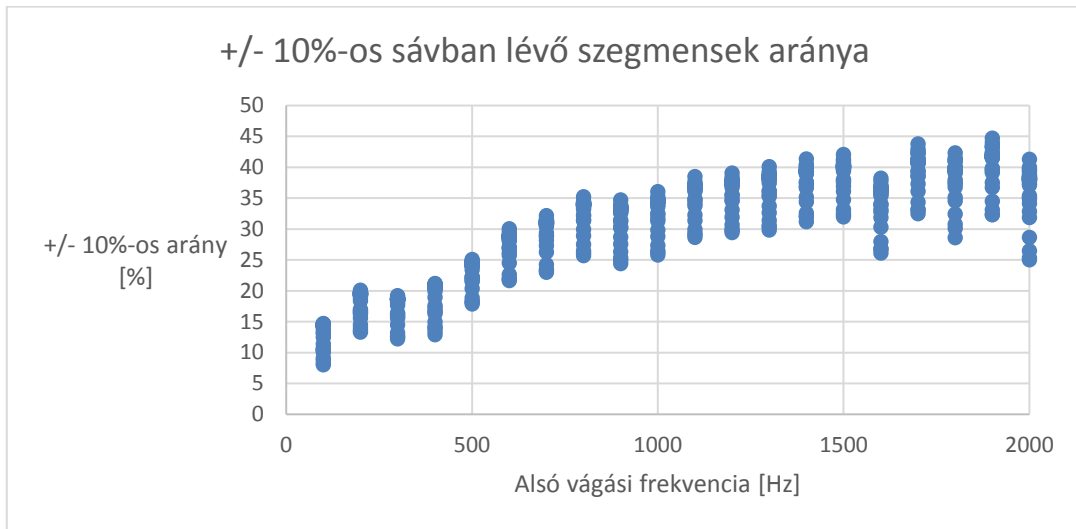


4-3. ábra, emberi beszéd frekvenciaspektruma

A szűrő vágási frekvenciáinak meghatározásakor fontos szempont, hogy csak azokat a frekvenciakomponenseket szűrjük ki a jelből, amelyek az iránydetektálás számára téves információkat hordoznak. A következő grafikonok úgy készültek, hogy a megvalósított rendszer következő fejezeteiben szereplő tételek még nem kerültek megvalósításra, a 4.2-es fejezetben szereplő triggerelésen kívül. A következő ábrákon a 4. fejezet elején ismertett módon elvégzett statisztikai vizsgálatok eredménye látható. A grafikonok azoknak a szegmenseknek az arányát mutatják az összes feldolgozott, tehát triggerszint feletti szegmensek függvényében, ahol a valódi jelutak különbségéhez képesti  $\pm 10\%$ -os tartományon belüli irányt sikerült detektálni az adott szegmens alapján.

A következő grafikonokon a 4.1.1. fejezethez hasonlóan az alsó és felső vágási frekvenciát hangoltam egy beágyazott ciklusban. Mindkét grafikon az egyik vágási frekvencia függvényében mutatja az első osztályba kerülő szegmensek arányát, míg a másik paraméter változó marad.

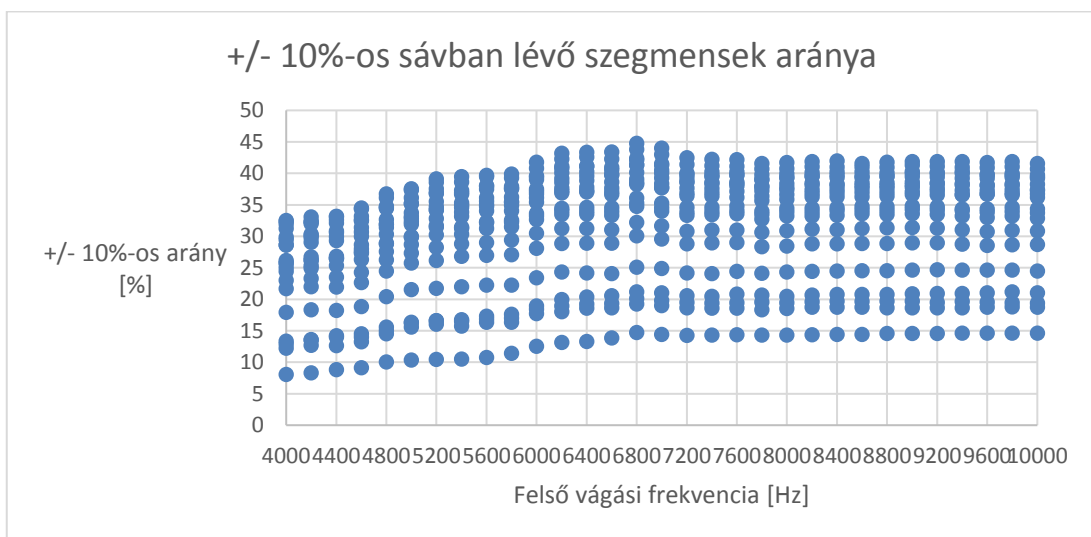
Lineáris fázisú sáváteresztő szűrő alsó vágási frekvenciájának ideális értékének meghatározása:



4-4. ábra, szűrő alsó vágási frekvenciája

Ahogy az a grafikonon látható, az alsó vágási frekvencia 1000 Hz feletti értékei jelentősen jobb pontosságot eredményeznek. Ezáltal a 4-3. ábrán látható spektrumkép alapján a jel teljesítményének nagy része kiszűrődik. Ennek oka tapasztalataim szerint az, hogy a felbontás frekvenciatartományban is lineáris, ezért kis frekvenciák esetében a felbontás viszonylag nagy, kevés frekvenciakomponens található, ezek teljesítménye viszont nagyságrenddel nagyobb. Ezek a jelenségek együttesen okozhatják a lokalizáció robusztusságának romlását.

A felső vágási frekvencia optimális értékének meghatározása:



4-5. ábra, szűrő felső vágási frekvenciája

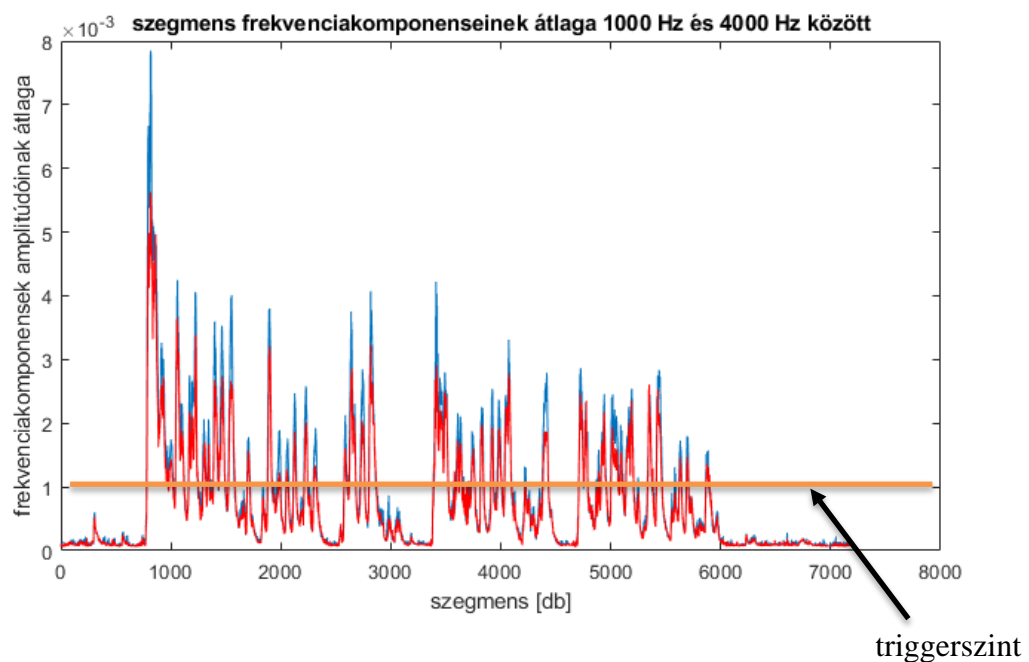


Jól látható, hogy körülbelül 7 kHz-ig a felvett jel tartalmaz hasznos tulajdonságokat az iránydetektálás számára (annak ellenére, hogy 4-5 kHz felett már minimális teljesítménnyel rendelkeznek), a felett pedig a levegőben lévő zajok enyhén rontják a lokalizáció pontosságát.

## 4.2. Triggerelés és jelminőség

### 4.2.1. Triggerszint meghatározása

Jelen alkalmazásban fontos kérdés, hogy az aktuálisan vételezett szegmenseket fel kell-e dolgozni. Ennek eldöntése a feldolgozott jel teljesítményének nagyságán, illetve eloszlásán alapszik. Az, hogy valaki éppen beszél-e, meghatározható triggerszint vizsgálat segítségével. Mivel az emberi hang teljesítményének nagy része egy jól meghatározható frekvenciatartományon belül van jelen, ezért érdemes a trigger vizsgálatot csak ezen a tartományon elvégezni, kizárva ezzel olyan nagy teljesítményű, más frekvenciájú zajokat, amelyek az iránydetektálás pontosságát csökkenthetik. A használni kívánt frekvenciasáv alsó határának az 1000 Hz-et, felsőnek a 4000 Hz-et választottam. A triggerszintnek elegendő nagyságúnak kell lennie, hogy azok a szegmensek, melyek nem tartalmaznak beszédet, ne juthassanak át rajta. A 4-6. ábrán látható a vizsgált hangfelvétel szegmenseiben lévő frekvenciakomponensek átlaga:



4-6. ábra, triggerszint

Az ábrán jól elkülöníthetőek azok a tartományok, amelyeknél csend lett felvéve, illetve azok, ahol beszéd. Mivel a munkám során fix erősítésértékekkel dolgoztam, így nem szükséges adaptív triggerszint használata.

#### 4.2.2. Jelminőség mérése

A jel minőségének analízise a jel teljesítményspektrumának szétterülésén alapszik. Tapasztalataim szerint ugyanis minél inkább szétoszlik az adott szegmens által hordozott teljesítmény a frekvenciakomponensek között, annál megbízhatóbbnak számít az iránydetektálás során. Ha egy szegmens teljesítménye túl kevés frekvenciakomponens között oszlik szét, akkor megbízhatatlannak számít, mivel egy-egy kiugró érték jelentősen eltérítheti az alkalmazás által meghatározott forrás irányát. Az elterülés számítására a következő egyenletet alkalmaztam:

$$leakage[k] = \frac{(X[k])^2}{\frac{\sum_{k=1}^{\frac{N}{2}+1} (X[k])^2}{\frac{N}{2} + 1}} \quad (4.1)$$

Ahol:

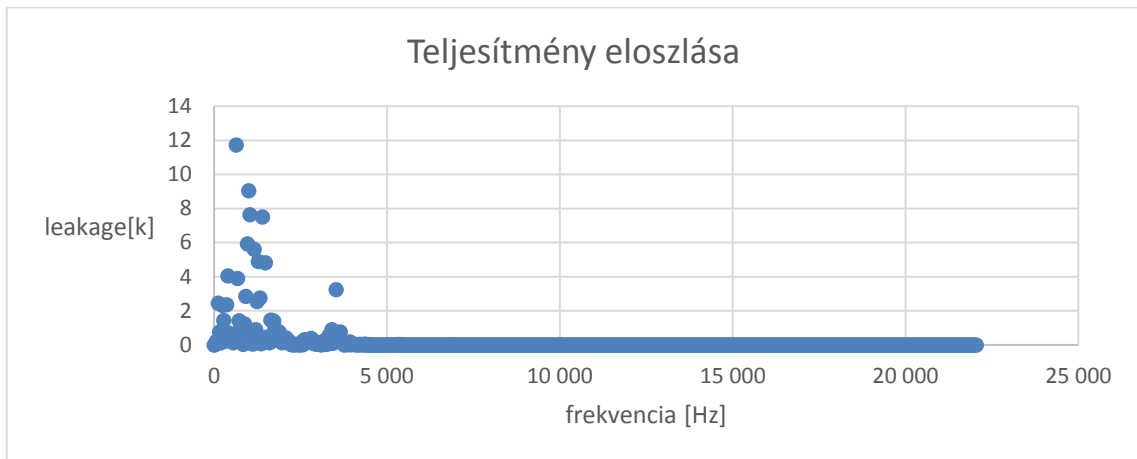
$X[k]$  - a mintavételezett jel frekvenciatartományban

$N$  - szegmens hossza

$leakage[k]$  - teljesítmény elterülésének mértéke

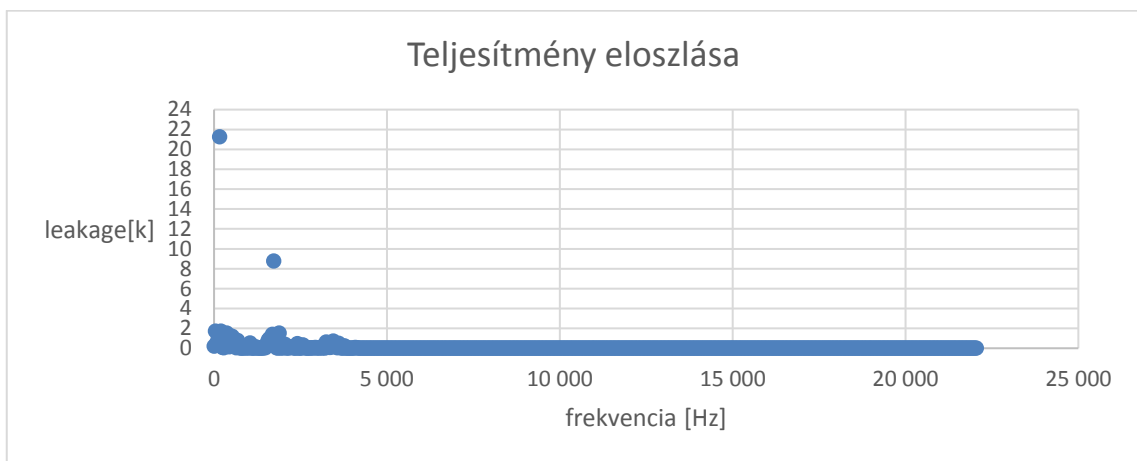
A következő két grafikonon szemléltetem a szegmensek teljesítményének szétterülésének vizsgálatát. Az ábrán a  $leakage[k]$  adattömböt ábrázoltam a frekvencia függvényében.

Az 4-7. ábra egy megfelelő mértékben szétterülő esetet mutat be:



4-7. ábra, teljesítmény nagymértékű szétterülése

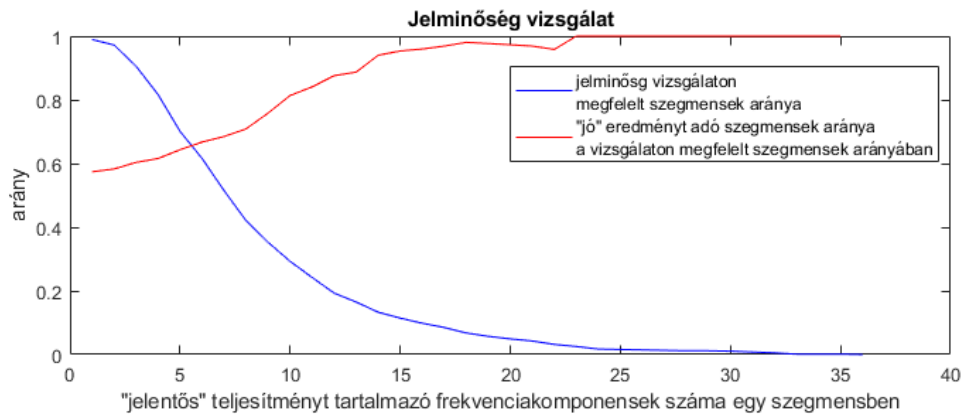
A következő ábra egy nem megfelelően szétterülő teljesítményű szegmenszet mutat be:



4-8. ábra, teljesítmény kismértékű szétterülése

Látható, hogy míg a 4-7. ábrán viszonylag nagyszámú frekvenciakomponens esetében található „jelentős”, tehát kiugró értékű teljesítmény, addig a 4-8. ábrán csak két esetben. A megfigyeléseim alapján a következő következtetésre jutottam: ha a *leakage[k]* tömbben az egyes frekvenciakomponensekhez tartozó érték nagyobb, mint 2, akkor azon a frekvencián „jelentős” teljesítmény van. Ha egy szegmenshez tartozó *leakage[k]* tömbben egy meghatározott számnál több helyen van „jelentős” teljesítmény, akkor az elterülés megfelelően nagy, a szegmens megbízhatónak minősül lokalizációs szempontból. Tehát iránydetektálásra csak azok a szegmenseket használtam fel, amik a triggerszint felett helyezkednek el, illetve a jelminőség vizsgálaton is megfelelnek.

Az 4-9. ábrán látható a jelminőség vizsgálat eredménye:



4-9. ábra, jelminőség vizsgálat

A fenti grafikon alapján el kell dönteni, hogy hány darab „jelentősnek” osztályozott komponens tartalmazzon egy szegmens ahhoz, hogy jó minőségűnek értékeljük. Ennek eldöntésére egzakt mérőszámot nem definiáltam, hanem a következő két mérőszám alapján kompromisszumos döntést hoztam. Amennyiben növeljük az egy szegmensben található domináns komponensek előírt minimális számát (4-9. ábra x tengelye), akkor javul a hibasávon belül található mérések aránya (4-9. ábra piros grafikon), de jelentősen nő az eldobott, rossz minőségűnek sorolt szegmensek aránya (4-9. ábra kék grafikon), ami a rendszer válaszidejét növeli. Kompromisszumos döntésként 10-es küszöbszintet választottam, így az összes szegmens 20%-át felhasználhatjuk további feldolgozásra, a maradó szegmensekre pedig körülbelül 80%-ban 20%-os hibasávon belül teljesít a lokalizációs algoritmus.

### 4.3. Normalizálás

Az előző fejezetben ismertetett jelminőség vizsgálat ellenére a mintavételezett jelnek továbbra is vannak kiugró amplitúdójú frekvenciakomponensei, amelyek pontatlanná tehetik a lokalizációt. Néhány ilyen frekvenciakomponens meghatározhatja, hogy a lokalizációs algoritmusok milyen irányból detektálják a forrást. Emiatt a jel normalizálása szükséges, a használt egyenlet a következő:

$$X[k] = r[k] \cdot e^{j\varphi[k]} \quad (4.2)$$

$$X_{\log}[k] = \log_{10}\left(\frac{ref + r[k]}{ref}\right) \cdot e^{j\varphi[k]} \quad (4.3)$$

Ahol:

$X[k]$  - a mintavételezett jel spektruma

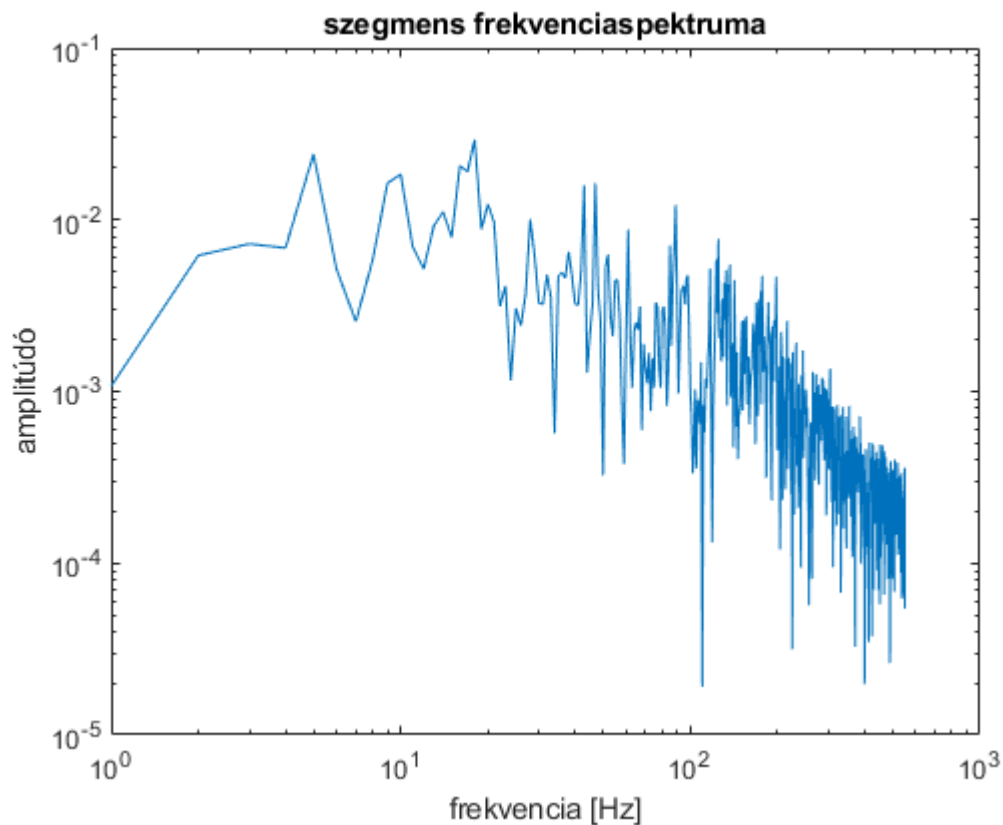
$r[k] \cdot e^{j\varphi[k]}$  -  $X[k]$  Euler alakja

$ref$  – konstans referencia jel

$X_{log}[k]$  – frekvenciatartományban normalizált mintavételezett jel

Normalizálás során szükséges az egyes vektorok szögeinek megtartása, hiszen időtartományban egy jel késleltetése frekvenciatartományban szögelfordulást jelent a Fourier-transzformáció eltolási tételének értelmében ((3.12) egyenlet).

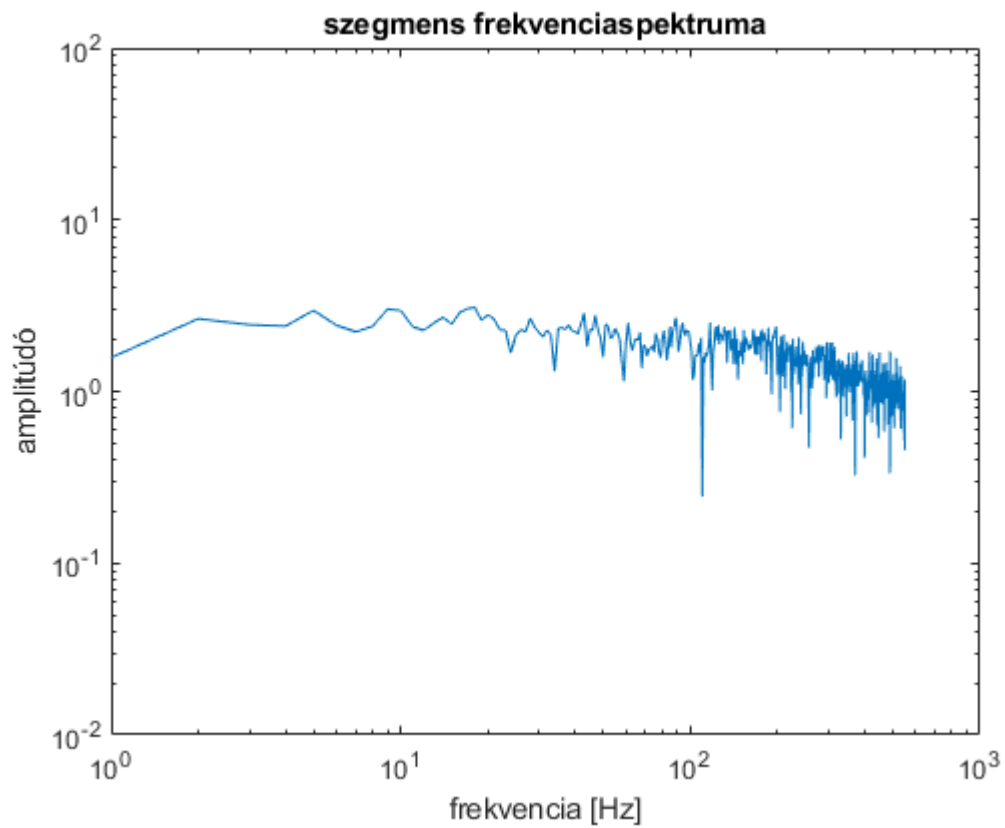
A következő grafikon egy trigger és jelminőség vizsgálaton átesett szegmens frekvenciaspektrumát ábrázolja:



4-10. ábra, szegmens frekvenciaspektruma normalizálás előtt

Az ábráról leolvasható, hogy az egyes frekvenciakomponensek amplitúdói 2 dekádnyi tartományban helyezkednek el.

A következő ábrán a (4.3) egyenlet felhasználásával normalizált szegmens frekvenciaspektruma látható:

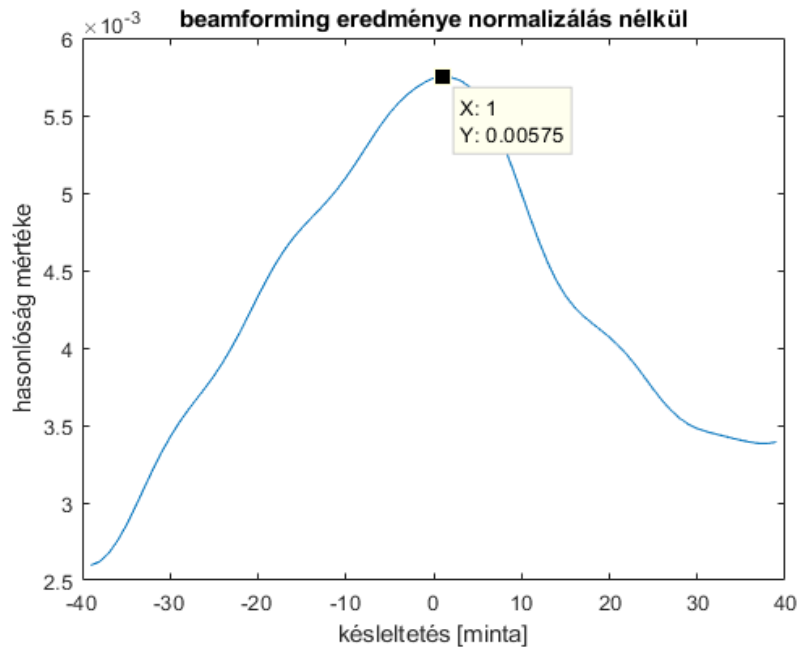


4-11. ábra, szegmens frekvenciaspektruma normalizálás után

A 4-11. ábrán látható, hogy normalizálás után az amplitúdók kevesebb, mint egy dekád szélességű tartományban helyezkednek el.

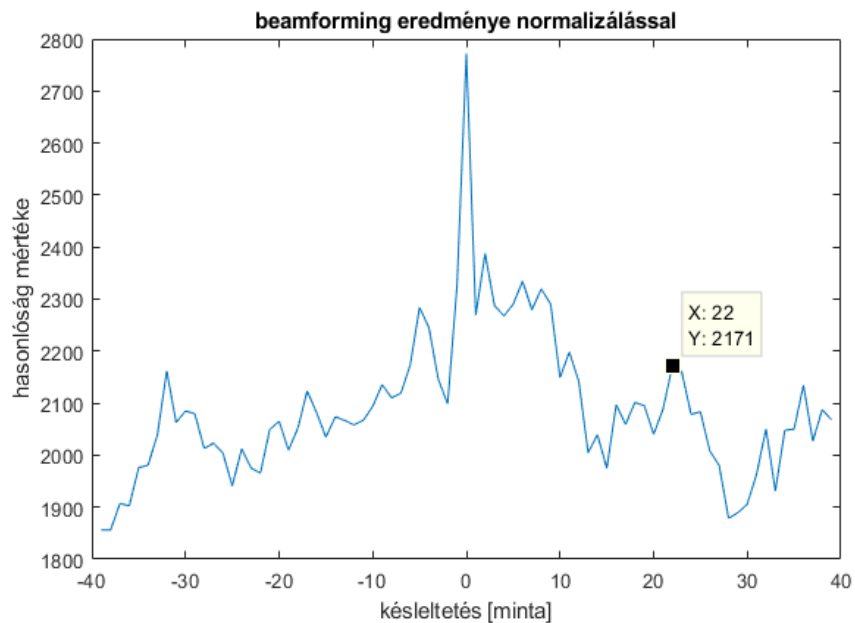
A mintavételezett jel normalizálása egy másik gyakran előforduló problémát is megold. Előfordulhat, hogy egy szegmensre valamelyik lokalizációs algoritmus rossz eredményt ad, vagyis a jelutak közti különbséget nem megfelelő pontossággal számolja ki.

Erre egy példa a következő grafikonon látható:



4-12. ábra, beamforming eredménye normalizálás nélkül

A valódi késleltetés a jelek között 27 minta, az adott szegmensre a beamforming algoritmus mégis 1-et ad. A későbbi utófeldolgozás (4.4. fejezet) során sincs lehetőség másik csúcs detektálására, hiszen az eredménytömb nem tartalmaz másik lokalizációra alkalmas csúcsot. Ha a beamforming algoritmus a fenti szegmenst normalizálva kapja meg a bemenetén, akkor az alábbi eredményt kapjuk:



4-13. ábra, beamforming eredménye normalizálással

Jól látható, hogy a legnagyobb csúcs pozíciója nem változott, de más csúcsok is megjelentek a grafikonon. Az ábrán a harmadik legnagyobb csúcs, a fekete négyzettel jelölt pont viszonylag közeli késleltetést határoz meg a valódi késleltetéshez képest, melyet az utófeldolgozás során akár detektálhat, és fel is használhat a program az eredeti késleltetés érték helyett.

#### **4.4. Utófeldolgozás**

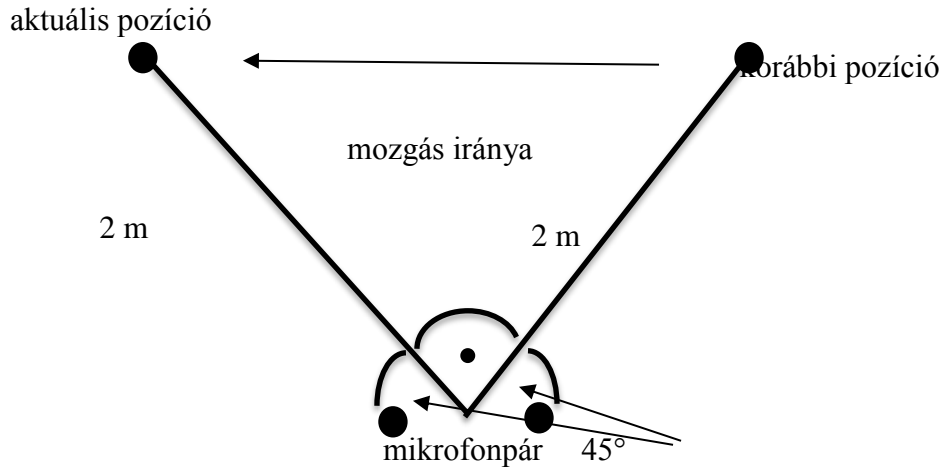
Az iránydetektálásra használt algoritmusok, mint a korreláció és a beamforming, nem minden esetben határozzák meg pontosan a hangforrás irányát. Ilyen esetekben érdemes utófeldolgozást végezni az eredményen, mivel a legnagyobb csúcs lokalizálása a módszerek kimeneti grafikonján számos esetben hoz nem megfelelő eredményt. Az utófeldolgozás blokk alkalmas még az iránydetektálás robusztusságának növelésére is. Az előző szegmensekből számolt pozíciókhoz tartozó időbélyeg alapján becsülhető a hangforrás sebessége, amellyel hihetőségi becsléseket végezhetünk az iránydetektáló algoritmusok által meghatározott pozíción. A következő alfejezetekben bemutatom, hogy milyen utófeldolgozásokat alkalmaztam.

##### **4.4.1. Korrekció**

A program futása közben, a korábbi szegmensek által detektált pozíció alapján becsülhető a hangforrás sebessége. Az alkalmazásban sztereo mikrofonpárt használok, tehát a forrásnak csupán az iránya határozható meg, tehát az, hogy milyen szögből érkezett a hang a mikrofonokhoz képest. A forrás pozíciójának pontos, térbeli meghatározásához, és ezáltal a sebességének becsléséhez szükséges további két paraméter. Az egyik a forrás magassága valamilyen referencia szinthez képest, a másik pedig a jelutak hossza, amit a hang megtesz, amíg eléri a mikrofonokhoz. Ezen két paraméter csupán becsülhető az alkalmazás módjából. Egy lehetséges alkalmazás, hogy egy nagy, sok táblával rendelkező előadóteremben a kamera mozgása kövesse az előadó mozgását a hangja alapján, és így mindig csak az aktuális táblát mutassa. Ebben az esetben becsülhető a magasság és a jelutak nagysága is, mivel az előadó magassága végig ugyanakkora marad, valamint a jelutak hossza is becsülhető egy állandó számmal, vagy akár egy irányfüggő értékkel is. Tehát ha ismerjük a hangforrás térbeli pozícióját, valamint az egyes szegmensekhez tartozó időbélyeg alapján ismerjük az időt is, tudunk sebességet számolni. Ha a korrekció blokkban becsüljük a forrás sebességét, hiszen tudjuk, hogy hol volt a forrás a korábbi szegmens esetén, és hol van most, akkor dönthetünk arról, hogy a kapott eredmény hihető-e vagy



sem. A következő ábra segítségével bemutatok egy példát, hogy mely esetekben nem hihető a lokalizációs algoritmusok eredménye:



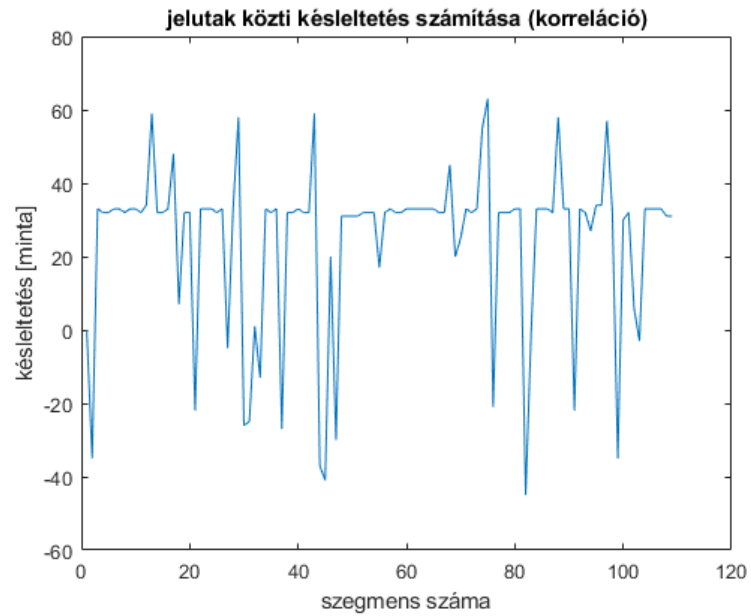
4-14. ábra, hangforrás sebességének becslése

A 4-14. ábrán látható, hogy a korábbi szegmens során meghatározott irány  $45^\circ$  volt, míg az aktuális esetben ugyancsak  $45^\circ$ , de az ellenkező irányban. Ez a jelenség főleg olyan esetekben fordulhat elő, amikor a belső tér falain reflektálódik a hang. Az alkalmazásban használt mintavételi frekvencia  $44100 \text{ Hz}$ , tehát a mintavételi idő körülbelül  $22 \mu\text{s}$ . Egy szegmens nagyságrendileg  $1000$  darab mintát tartalmaz, tehát nagyjából  $20 - 30 \text{ ms}$  hosszú. Két szomszédos szegmens között, amelyek a triggerelésnek és a jelminőség vizsgálatnak is megfeleltek, több másik szegmens is lehet, tehát a köztük eltelt idő néhányszor  $100 \text{ ms}$ . A forrás által megtett távolság állandó magasságot, és egyenes haladást feltételezve majdnem  $3 \text{ m}$ . Tehát a forrás  $v$  sebessége nagyságrendileg  $10 \frac{\text{m}}{\text{s}}$ , ami körülbelül  $30 - 40 \frac{\text{km}}{\text{h}}$ . Ilyen sebességű mozgás irreális a fent említett alkalmazási példa esetében, tehát nem fogadjuk el a lokalizációs algoritmusok kimeneti grafikonján a legnagyobb csúcs által meghatározott pozíciót (3. fejezet), hanem másik csúcsot detektálunk.

Ha tehát egy bizonyos értéknél nagyobb különbség van két szomszédos szegmens által meghatározott irány között, vagyis a hangforrás sebessége irreálisan nagy, akkor új csúcsot kell detektálni az iránydetektáló algoritmusok kimeneti grafikonján. Ezek a grafikonok a mikrofonok által vett jelek közti késleltetést adják meg mintában számolva (3-4. ábra). A maximális késleltetés az az idő, ami alatt a hang megteszi a mikrofonok közti távolságot. Ezen maximálisan lehetséges késleltetés határok között az összes csúcsot detektáljuk a korrekció során. Fontos döntés, hogy melyik új csúcs kerüljön felhasználásra.

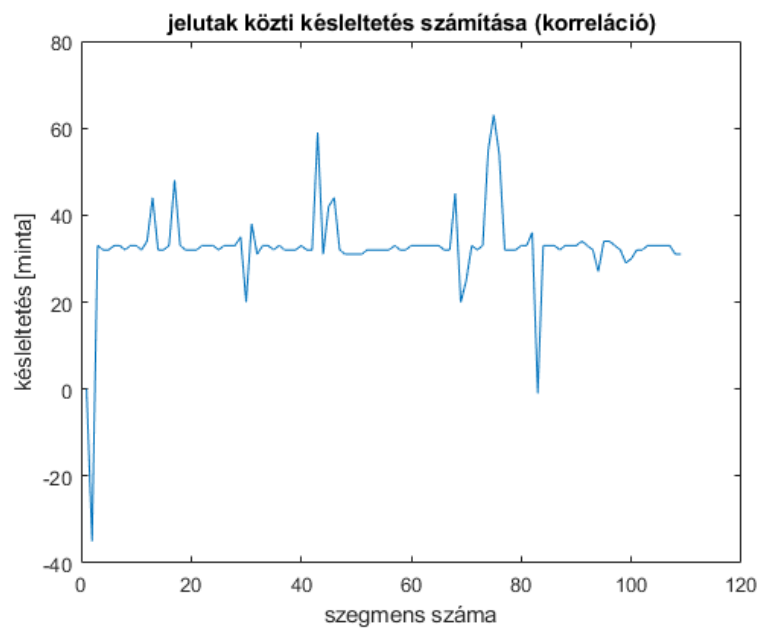
Robusztussági és egyszerűségi okok miatt az előző szegmens által meghatározott késleltetéshez képesti legközelebbi értéket választom ki. A következő két ábrán egy példa látható, hogy a korrekció miként javítja a forrás detektálásának pontosságát.

Korreláció algoritmus segítségével a jelutak közti késleltetés meghatározása korrekció nélkül egy kiválasztott, előre felvett hangfájlon:



4-15. ábra, iránydetektálás korrekció nélkül

Ugyanazon hangfájlon késleltetés meghatározása korrekció segítségével:



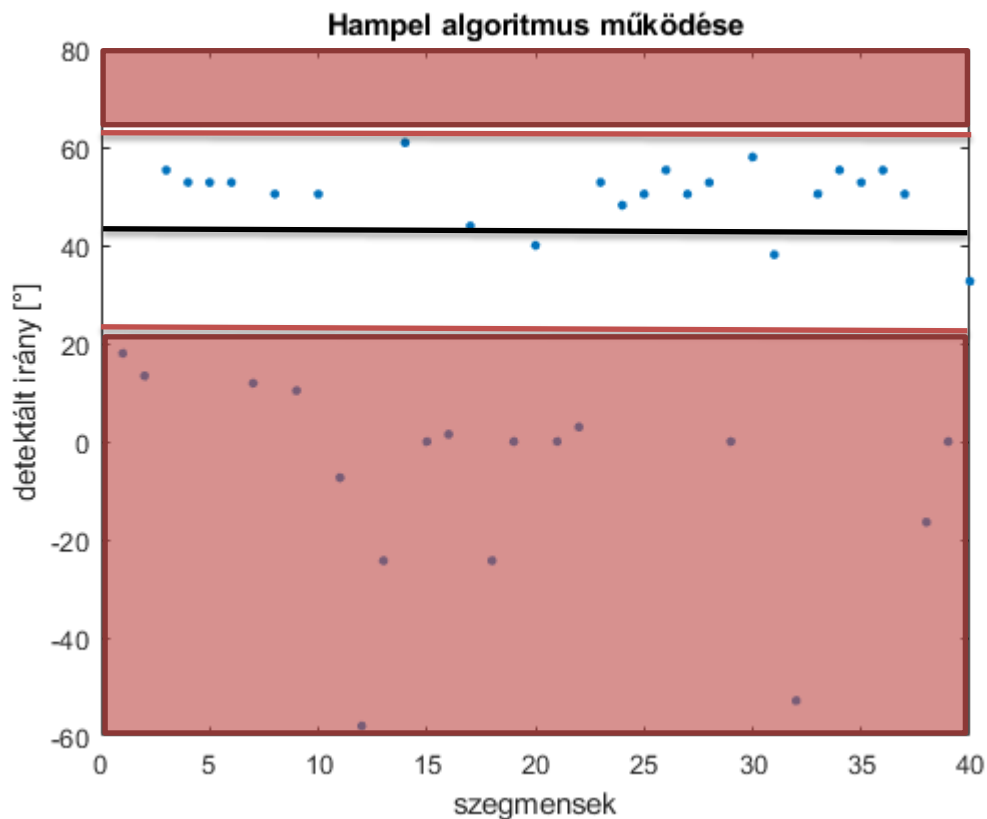
4-16. ábra, iránydetektálás korrekció segítségével

Ebben az esetben a valódi késleltetés a jelutak között 29 minta volt, mely viszonylag jól leolvasható mindkét grafikonról. A 4-16. ábrán jól látható, hogy a fent bemutatott korrekció miként javítja forrás irányának detektálását.

#### 4.4.2. Hampel algoritmus

Az utófeldolgozás során az időbeliség figyelembe vétele mellett egy másik fontos cél a robusztusság növelése. A különböző környezeti zajok és hang többutas terjedése megbízhatatlanná tehetik az alkalmazás működését. Ezért fontos az algoritmus működését minél masszívabbá tenni, hogy különböző körülmények között is megfelelően működjön. Robusztusság növelése érdekében alkalmazom a Hampel algoritmus egy változatát [16], melynek lényege, hogy egy adattömbből kiszűri a kiugró értékeket. Az algoritmus bemenete a korreláció és a beamforming algoritmusok által detektált irányok néhány korábbi, és az aktuális szegmens során. A tömbben lévő egyes adatoknak veszem a mediánjukhoz képesti négyzetes eltérésük. Ezután kiviszem azokat az elemeket, ahol ez a négyzetes eltérés kiugró értéket mutat a négyzetes eltérések mediánjához képest.

A következő ábra illusztrálja a Hampel algoritmus működését:

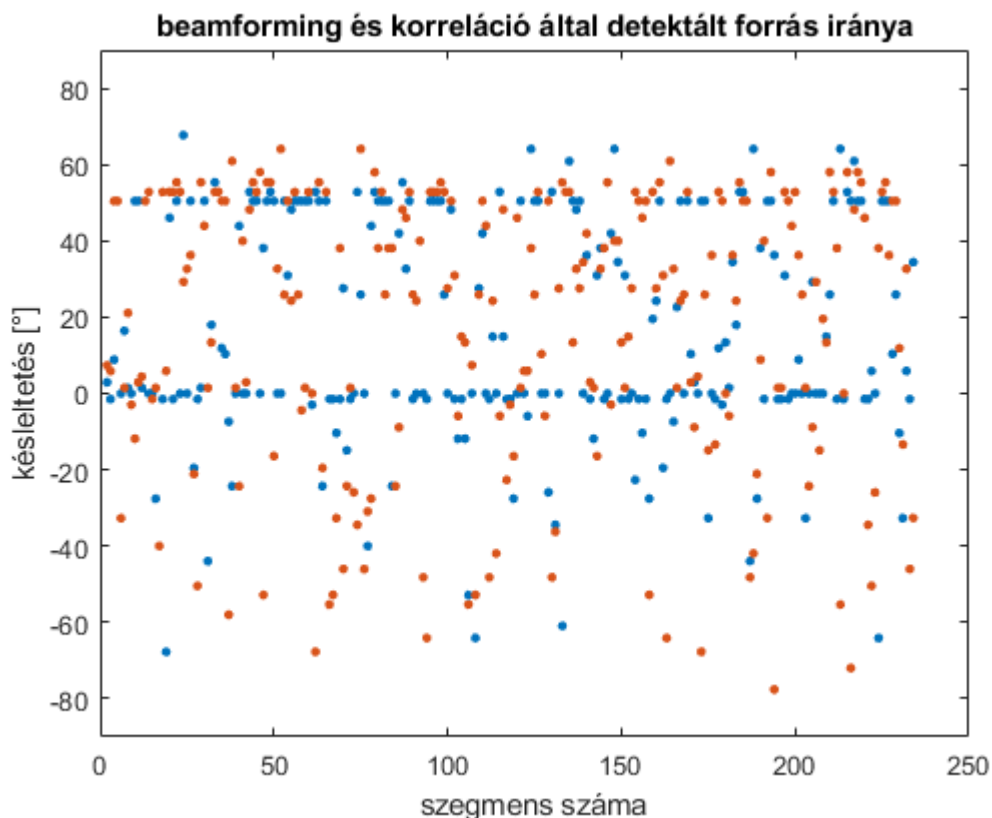


4-17. ábra, Hampel algoritmus működése

A 4-17. ábrán látható esetben a forrás valódi iránya  $52^\circ$  volt. Fekete vonallal jelöltem az adattömb mediánját ( $42^\circ$ ), illetve piros vonallal a mediánhoz képesti négyzetes eltérés mediánját. Azokat a pontokat, amelyek ezen kívül esnek, tehát az áttetsző piros téglalapok valamelyikében vannak, kiveszem az adattömbből. A tömb megmaradt elemeinek újra veszem a mediánját, amely értéket a 4.5. fejezetben leírtak szerint a fúzió során használok majd fel.

## 4.5. Fúzió

Fúzió során a korreláció, a beamforming és a Hampel algoritmusok által meghatározott jelutak közti késleltetés értékek összegzése valósul meg. Előfordulhat olyan eset, hogy valamelyik lokalizációs módszer a korrekció ellenére pontatlan késleltetést határoz meg, de az összegzés által a hiba kisebb mértékű lesz. Az alkalmazás tesztelése közben megfigyeltem, hogy az iránydetektáló algoritmusok hajlamosak tévedni a  $0^\circ$ -os szög felé. Erre mutat példát a 4-18. ábra:



4-18. ábra, lokalizációs algoritmusok tévedése  $0^\circ$  irányába

Az ábrán a korreláció és a beamforming algoritmusok kimeneti grafikonjának legnagyobb csúcsai által meghatározott irányok láthatóak. A diagramról leolvasható, hogy a mérés

során a beszélő a mikrofonokhoz képest körülbelül 50°-os szögben helyezkedett el. Jól látható, hogy mindkét módszer hajlamos a kisebb szögek felé tévedni, ezért a fúzió során nemcsak összegzem az eredményeiket a Hampel algoritmus eredményével együtt, hanem a nagyobb szögek felé súlyozom is azt. Az összegzés és súlyozás egyenlete a következő:

$$result = \sqrt[10]{\frac{K^{10} + B^{10} + H^{10}}{3}} \quad (4.4)$$

Ahol:

$K$  – korreláció által detektált irány

$B$  – beamforming által detektált irány

$H$  – Hampel algoritmus által számolt irány

A (4.4) egyenlet szerinti összegzést empirikus úton hoztam létre, a használata elősegíti az alkalmazás robusztusabb, megbízhatóbb működését, amely kulcsfontosságú annak érdekében, hogy más-más körülmények között is megfelelően működjön az iránydetektálás. A 4. fejezetben ismertetett program tesztelésének folyamatáról és annak eredményeiről a következő fejezetben lesz szó.

## 5. Tesztelés

A 4. fejezetben ismertetett megvalósított rendszernek széleskörű tesztsorozat segítségével megismertem iránydetektálási pontosságát, robusztusságát a körülményektől függően. Minden belső tér más és más hullámkarakterisztikával rendelkezik, ezáltal másképp verődnek vissza a hanghullámok a terem falairól, máshol vannak erősítési és elnyomási pontok. Léteznek olyan hangtechnikai megoldások, ahol ezt a karakterisztikát felméri, például színházak, mozi termek kihangosításánál, hogy minél élvezhetőbb, valóságosabb hangzást érjenek el. Jelen alkalmazásomban erre nem volt lehetőségem, viszont az algoritmus paramétereinek általánosan jó, minél robusztusabb beállításával a célom az volt, hogy a körülményektől függetlenül, megfelelő pontosságú iránydetektálást valósítsak meg. Az egyik ilyen zavaró körülmény a hang többutas terjedése (3-5. ábra), mely jelenség hatásának csökkentése érdekében a mikrofonokat próbáltam fal elé helyezni.

Az alkalmazás tesztelése során a leendő felhasználás szerint teszteltem. A 2.1. és 2.3. fejezetek alapján az alkalmazást elsősorban 10 cm-es mikrofontávolsággal teszteltem. A további 30 és 50 cm-es távolságok azért kerültek kiválasztásra, hogy látható legyen az esetleges javulás vagy romlás az iránydetektálás során a nagyobb mikrofontávolságok hatására. Ilyen esetekben nagyobb a mikrofonokhoz érkező jelek közti útkülönbség, valamint javul a lokalizáció felbontása is a (3.4) és (3.5) egyenletek alapján. A kisebb felbontásnak csupán akkor van jelentősége, ha az iránydetektálás egyébként megfelelően pontos, hibája egy nagyságrendben van a felbontás nagyságával. A nagyobb mikrofontávolságok hátránya viszont, hogy kisebb az a maximális hullámhossz, amelynek esetében még egyértelmű a jel késleltetése. A következő alfejezetekben bemutatom, hogyan teszteltem a lokalizáció működését valós körülmények között.

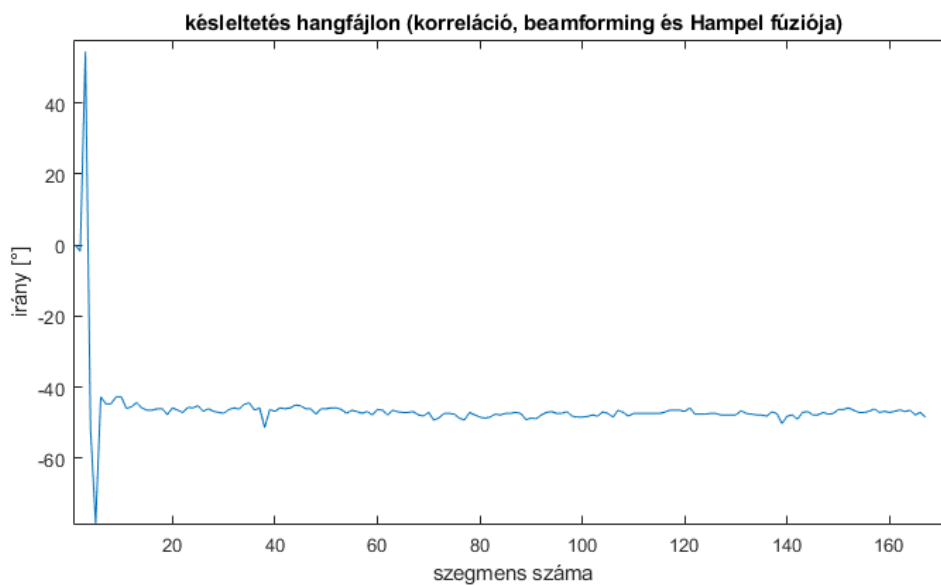
### 5.1. Offline tesztek

Ahhoz, hogy az alkalmazás helyesen működjön valós idejű megvalósítás esetén, megfelelően kell működnie offline módon is, azaz fájlból beolvasva az előre felvett hangot. Ezen tesztek során könnyebb az algoritmus paramétereinek finomhangolása, van idő akár 8 – 10-féle grafikont is kirajzolni, ezáltal látható, hogy milyen esetekben, pontosan melyik szegmensnél miért volt pontatlan a forrás pozíciójának meghatározása. Offline tesztet kétféleképpen végeztem, egyik esetben a forrás pozíciója állandó volt a felvétel során, a másik esetben pedig folyamatosan mozgott.

### 5.1.1. Állandó pozíciójú forrás

A felvétel közben álló forrással 3 különböző teremben végeztem tesztek a BME I épületében, egy kisebb (IE317), egy közepes méretű (IE224) és egy nagy teremben (IB026). Mindhárom teremben készítettem felvételeket 10, 30 és 50 cm-es mikrofontávolságokkal. A mikrofonok felé több irányból is beszéltem, nagyjából egyenletesen lefedve a  $-90^\circ$ -tól  $90^\circ$ -ig tartó tartományt. A felvételek hossza nagyjából 10 másodperc volt, amely elegendően hosszúnak bizonyult ahhoz, hogy az iránydetektálás pontossága látható legyen az adott elrendezés esetén.

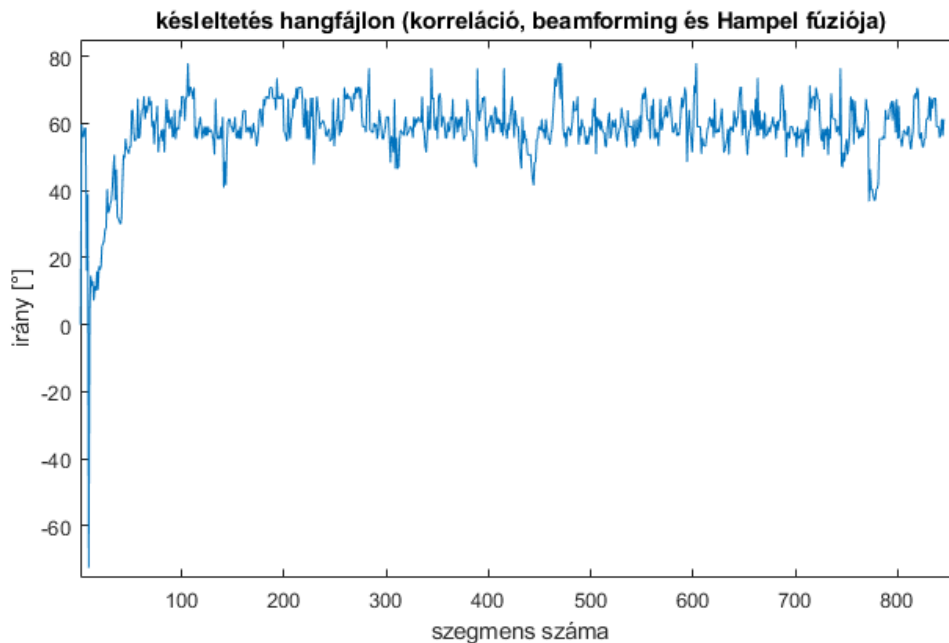
Elmondható, hogy a terem méretétől és a mikrofonok távolságától függetlenül a  $30^\circ$  és  $60^\circ$  közötti tartományban mér a legpontosabban az algoritmus. A hiba általában kisebb, mint 10%, valamint kevés kiugró eredmény jelenik meg a kimeneten, a program működése robusztus. A következő grafikon egy kis teremben készült felvétel eredményét mutatja, a mikrofonok távolsága 30 cm, a forrás valódi iránya  $-45^\circ$ .



5-1. ábra, IE317, 30 cm-es mikrofontávolság,  $-45^\circ$ -os valódi irány

A grafikonról leolvasható, hogy a lokalizáció folyamatos, stabil eredményt hoz. Az x tengelyen szerepelnek sorban azok a szegmensek, amelyek a triggerszint és jelminőség vizsgálatnak megfeleltek.

Ugyanezen beállított paraméterek mellett a következő eredményt kaptam egy nagy teremben készült felvételtől, ahol a mikrofonok távolsága 50 cm volt, a forrás valódi iránya pedig  $60^\circ$ :



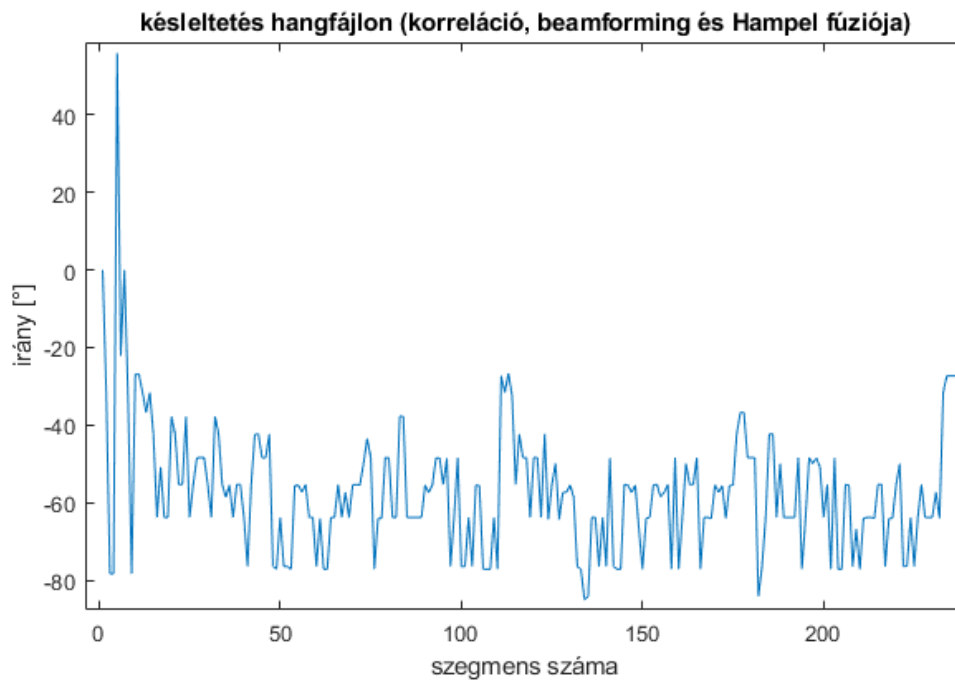
5-2. ábra, IB026, 50 cm-es mikrofontávolság,  $60^\circ$ -os valódi irány

Látható, hogy a detektált forrás iránya ebben az esetben is stabil, rendszeres hibája nem számottevően nagy. A szegmensek nagyobb száma a mintavételezett jel nagyobb teljesítményének tudható be, nem annak, hogy nagyobb teremben jobban eloszlaná a teljesítményspektrum (4.2. fejezet). Mindkét felvétel esetében megfigyelhető egy tranziens állapot, mely kevesebb, mint 1 másodpercig tart. A későbbi valós idejű alkalmazás során a felvétel eleji tranziens hatások elhanyagolhatóak lesznek, az állandósult állapotbeli működés megfelelő.

Az iránydetektálás pontossága romlik, ahogyan a sztereo mikrofonok vonalába ér a forrás, azaz  $90^\circ$ -os vagy  $-90^\circ$ -os szögből érkezik a hang. Az eredmények rendszeres hibával lesznek terhelvek. A hiba nagysága 10 cm-es mikrofontávolság esetén a teremtől függetlenül nagyobb, körülbelül  $30^\circ$ , míg 30 és 50 cm-es mikrofontávolságok esetén kisebb, nagyjából  $15 - 20^\circ$ .

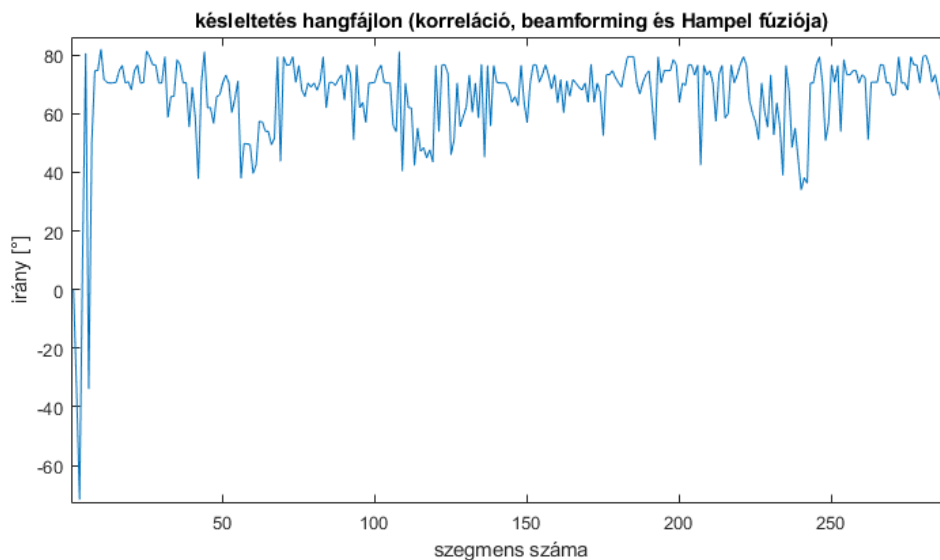


A következő grafikon a 10 cm-es mikrofontávolság esetén létrejövő nagy rendszeres hibát illusztrálja, a felvétel a kis teremben készült:



5-3. ábra IE317, 10 cm-es mikrofontávolság,  $-90^\circ$ -os valódi irány

Látható, hogy nem csak a rendszeres, de a véletlen hibák nagysága is számottevő. A mikrofonok közti távolság növelésével a rendszeres hiba nagysága csökkenthető:



5-4. ábra, IE224, 50 cm-es mikrofontávolság,  $90^\circ$ -os valódi irány

A mérési eredmények legnagyobb része  $70^\circ$  környékén van, a véletlen hibák nagysága nem csökkent. A tranziens hatások itt is megfigyelhetők, ahogyan az előző esetekben.

Annak oka, hogy ha a hang oldalról érkezik, akkor miért ekkora a rendszeres hiba, a koszinusz függvény tulajdonságából következik. A 3-2. ábra szerint  $\Delta d$  maximális értéke  $d$ , ekkor  $\alpha' = 0^\circ$ . Korreláció során mindenképpen kvantált a felbontás. Ennek oka, hogy időtartományban mintavételi időnként lehetséges a mikrofonokhoz érkező jelek egymáshoz képesti eltolása. Beamforming esetén pedig számítási kapacitási okokból kvantált a felbontás. Korrelációnál a felbontás nagysága:

$$\frac{1}{F_s} \cdot c = d_s \quad (5.1)$$

Ahol:

$F_s$  – mintavételi frekvencia, 44100 Hz

$c$  – hang sebessége, 340 m/s

$d_s$  – hang által megtett távolság két mintavétel között

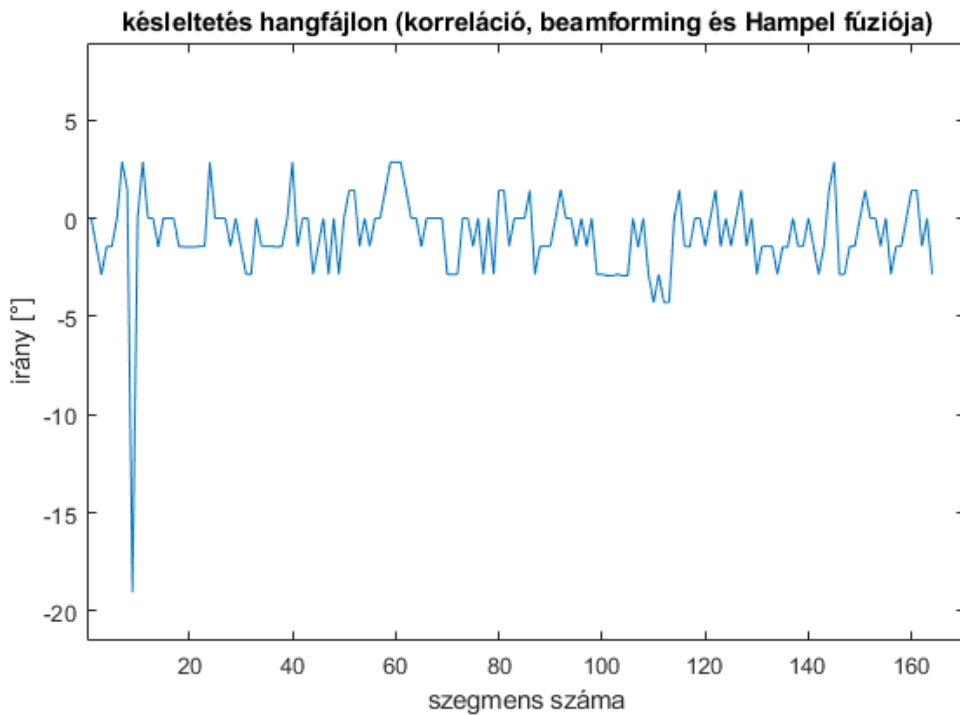
Két mintavétel között a hang 0,77 cm-t tesz meg. Tehát, ha a korrelációs algoritmus a kimeneti grafikonján egyetlen mintányit is téved a késleltetés meghatározásánál 10 cm-es mikrofontávolság esetén, és nem a maximális 10 cm-t határozza meg a jelutak különbségére, hanem 9,23 cm-t, akkor a tévedés nagysága fokban kifejezve:

$$\cos^{-1} \frac{9,23 \text{ cm}}{10 \text{ cm}} = 22,6^\circ \quad (5.2)$$

Ebben az esetben a forrás iránya  $67,4^\circ$  ( $0^\circ$  esetén van pontosan a mikrofonok előtt a forrás). A felbontás a mikrofontávolságok növekedésével csökken, ezáltal a lokalizáció pontossága is nő. Az iránydetektálás továbbá javítható még a beamforming algoritmus felbontásának csökkentésével, amely a fúzió során, az eredmények összegzésénél finomítja, javíthatja a forrás irányának detektálását.

Ha a forrás körülbelül a mikrofonok előtt helyezkedik el, az iránydetektálás stabil, viszonylag kis rendszeres hibával terhelt a terem méretétől és a mikrofonok távolságától függetlenül.

Erre mutat példát a következő grafikon:



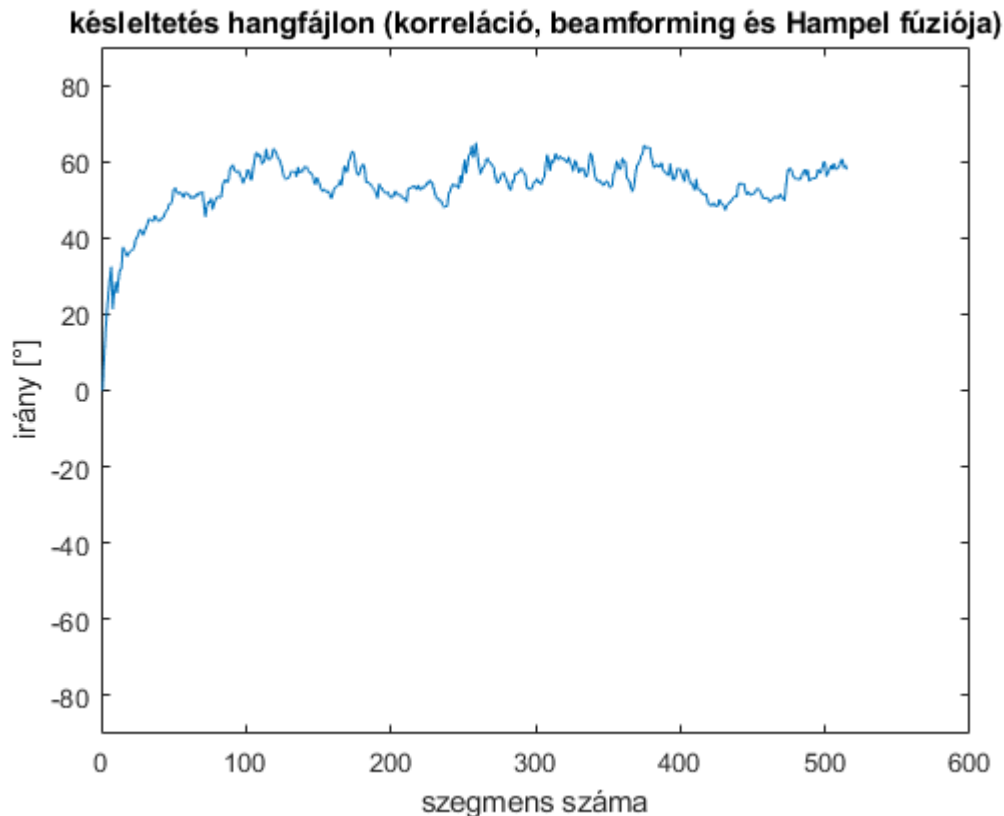
5-5. ábra, IE317, 50 cm-es mikrofontávolság, 0°-os valódi irány

Összességében elmondható, hogy az iránydetektálás működése megfelelő, a rendszeres hiba kicsi, kivéve azokat az eseteket, amikor a forrás nagyjából a mikrofonokkal egy vonalban van. Ebben az esetben a mikrofonok távolságának növelése, illetve a beamforming algoritmus felbontásának csökkentése segít a rendszeres hiba csökkentésében. A véletlen hibák nagysága a 0° és 60° közötti tartományban a kisebb, melynek oka a koszinusz görbe alakjából következik. A véletlen hiba nagyságának csökkentése a lokalizáció eredményének kijelzésében exponenciális átlagolással lehetséges:

$$y[i] = a \cdot y[i - 1] + (1 - a) \cdot x[i] \quad (5.3)$$

Az  $a = 0,8$  paraméter segítségével a kijelzés során látható jel tuskés jellege csökkenthető.

Az exponenciális átlagolás hatása az 5-2. ábra grafikonján:



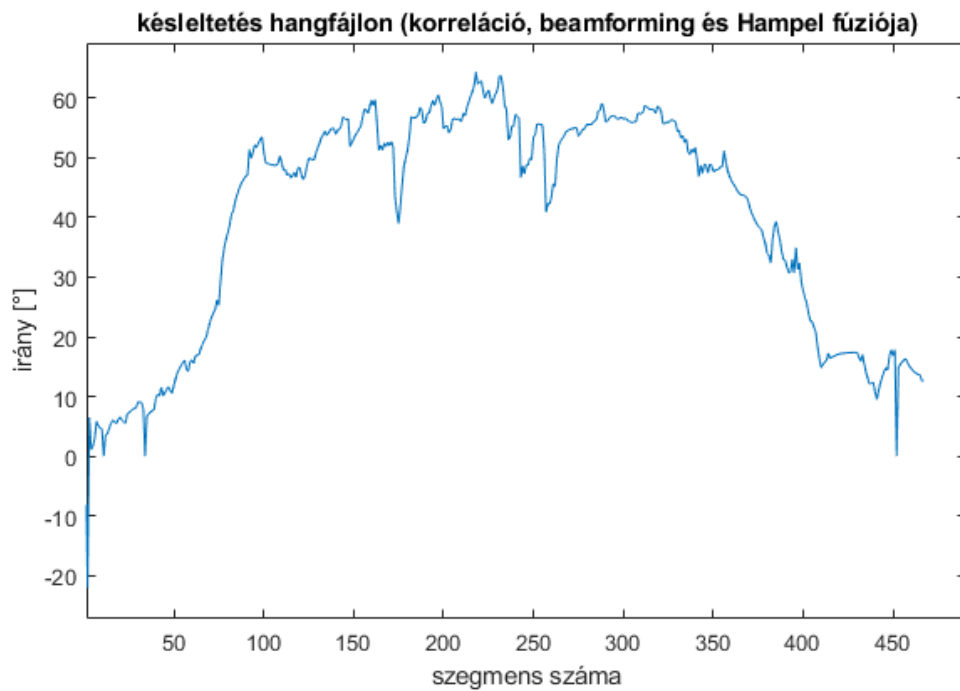
5-6. ábra, exponenciális átlagolás hatása

A következő fejezetekben látható grafikonok már az exponenciális átlagolás utáni eredményeket mutatják a jobb áttekinthetőség miatt.

### 5.1.2. Mozgó forrás

Az állandó pozíciójú forrással végzett tesztek során megismertem az algoritmus működését különböző körülmények között. Mozgó forrás esetén az iránydetektálás jellege, azaz a rendszeres és véletlen hibák nagysága és gyakorisága nem változik. Ami miatt mégis fontosak ezek a tesztek az az, hogy 4.4.1. fejezetben bemutatott korrekció helyesen működik-e. Korrekció során, ha két egymást követő szegmens által meghatározott irány között túl nagy a különbség, tehát a forrás sebessége valószínűleg túl nagy, akkor új pozíció kerül meghatározásra. Ha ennek a bloknak a paraméterei rosszul vannak beállítva, akkor előfordulhat, hogy a forrás valódi mozgását nem követik az eredmények, hanem a valódi pozíció helyett mindig egy, az előzőhöz legközelebbi pozíciót választ. Mozgó forrás esetén 10, 30 és 50 cm-es mikrofontávolságokkal végeztem tesztek az IE317, azaz

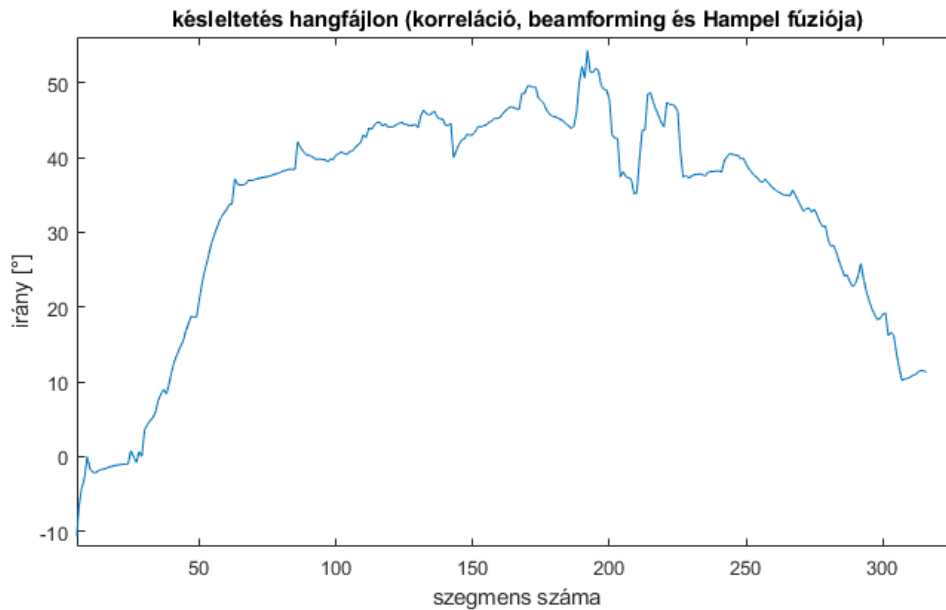
a kis teremben. A következő ábra a 10 cm-es mikrofontávolságú eset eredményét mutatja meg:



5-7. ábra, mozgó forrás, 10 cm-es mikrofontávolság

A felvétel elején a forrás a mikrofonok előtt helyezkedett el, majd elindult oldalra, egészen  $70^\circ$ -os kitérésig, majd visszatért a középső helyzetbe. Az 5-7. grafikonon látható, hogy amikor a forrás kis szögben látszik, akkor a rendszeres hiba kicsi, nagyobb szögek esetén pedig megnő. Ez látható a grafikon ellaposodásán, annak középső részén.

50 cm-es mikrofontávolság esetén is hasonló jelenség figyelhető meg, a maximális kitérés ez esetben  $60^\circ$  volt:



5-8. ábra, mozgó forrás, 50 cm-es mikrofontávolság

A korrekció blokk paraméterei láthatóan jól működtek, mozgó forrás esetén nem tapasztaltam olyan mértékű rendszeres vagy véletlen hibát, amelyet az álló forrás során végzett tesztek során ne tapasztaltam volna.

## 5.2. Valós idejű tesztek

Szakedolgozatomban a cél az volt, hogy az alkalmazás ne csak előre felvett hangfájlok-ból, hanem valós időben, a bemenetet külső hangkártyáról olvasva is megvalósuljon. A valós idejű működés feltétele, hogy az egy szegmens feldolgozásához szükséges idő kisebb legyen, mint két szegmens érkezése közötti idő. A mintavételi idő megegyezik a mintavételi frekvencia reciprokával, azaz:

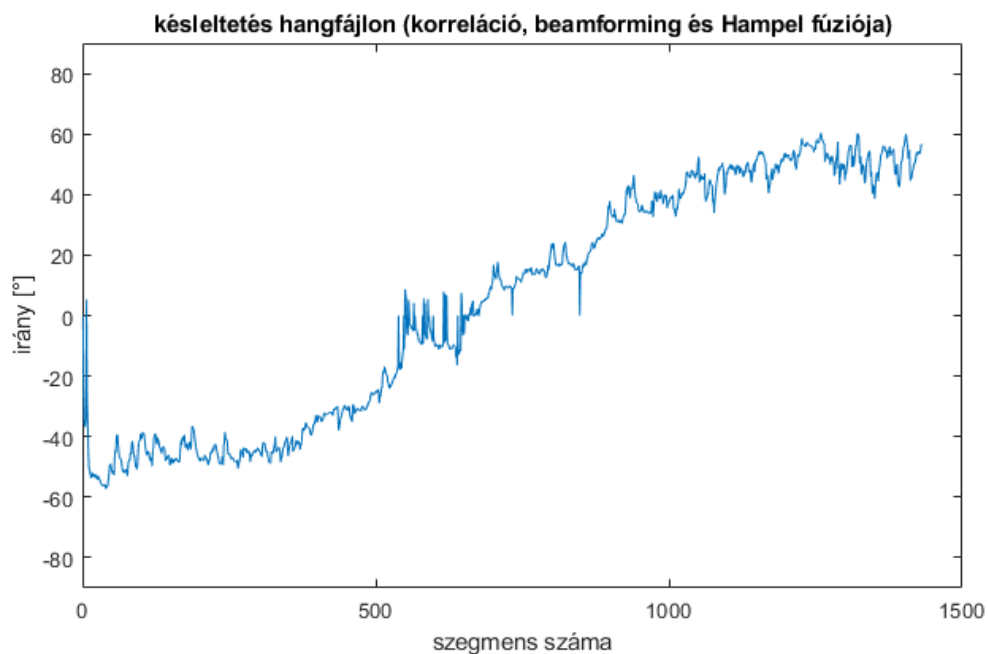
$$T_s = \frac{1}{F_s} = \frac{1}{44100 \text{ Hz}} = 22,6 \mu\text{s} \quad (5.4)$$

A 4.1.1. fejezet alapján 1100 mintát tartalmaz egy szegmens a megvalósított rendszerben. Tehát egy szegmens által felölelt idő:  $1100 \cdot 22,6 \mu\text{s} \approx 25 \text{ ms}$ . A Matlab R2018a program clock függvényének segítségével megmértem, hogy körülbelül 14 – 16 ms ideig tart egy szegmens feldolgozása. A maximális mértékű átlapolódás, ami mellett nem veszik el

adat,  $1 - \frac{16 \text{ ms}}{25 \text{ ms}} = 36\%$ . A valós idejű megvalósított rendszerben 30%-os átlapolódást állítottam be.

A tesztek felépítése hasonló az offline módon, mozgó forrással végzett tesztekhez. A mikrofonok ebben az esetben is 10, 30 és 50 cm-re voltak egymástól, a helyszín pedig az IB026-os, azaz a nagyterem volt, ahol alkalmam volt a mikrofonok előtt végigsétálni egyenletes tempóban egyik irányból a másikba beszéd közben. A következőkben bemutatok néhány grafikont, amelyeket az alkalmazás működés közben rajzolt ki, és a forrás irányát illusztrálják.

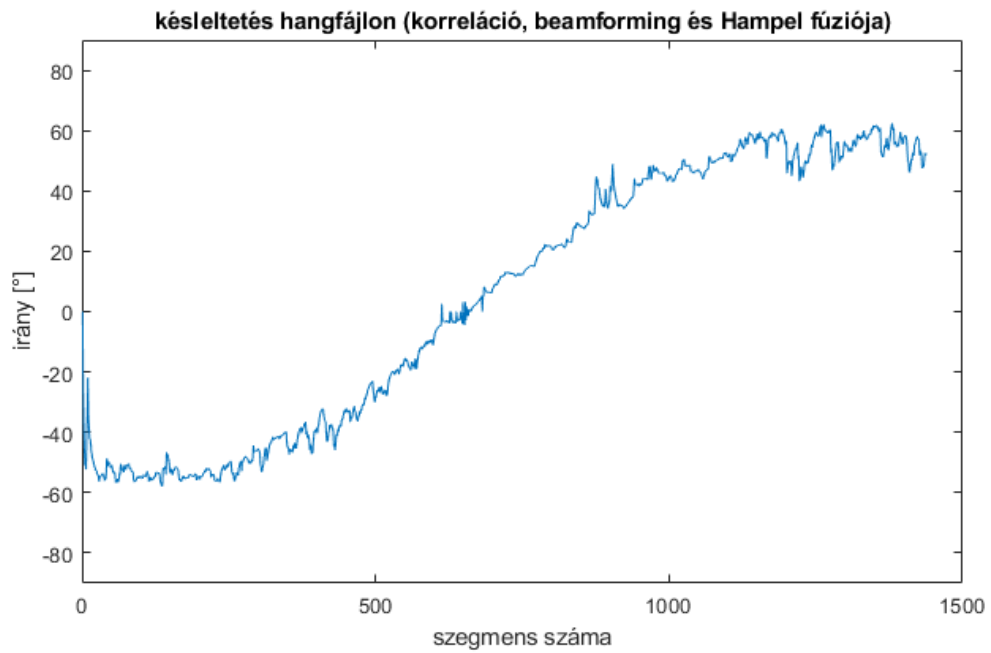
Az 5-9. ábra egy 10 cm-es mikrofontávolságú mérés eredményét mutatja,  $-60^\circ$ -os szögben kezdődött a mérés, majd egyenletes tempóban  $60^\circ$ -os kitérésig haladtam el:



5-9. ábra, valós idejű teszt, 10 cm-es mikrofontávolság

Az 5-9., 5-10. és 5-11. grafikonok ideális esetben az arctangens függvény adott tartománybeli alakját veszik fel. Maximum  $10\text{-}15^\circ$  rendszeres hibával rendelkezik a program a mérés alatt, a véletlen hibák nagysága nem gyakori, nagyságuk elfogadható.

Az 5-10. ábra esetén a mikrofonok 30 cm-re voltak egymástól, ugyancsak  $-60^\circ$ -tól  $60^\circ$ -ig haladtam el egyenletesen a mikrofonok előtt:

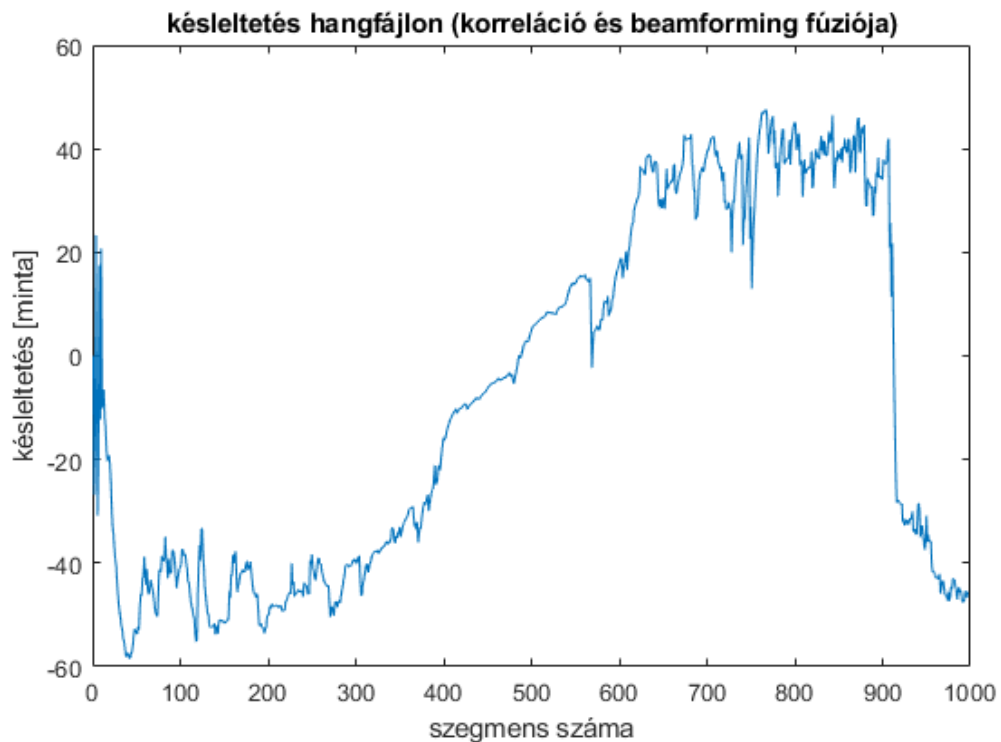


5-10. ábra, valós idejű teszt, 30 cm-es mikrofontávolság

Az lokalizáció pontossága körülbelül megegyezik az előző esetben tapasztalttal.



Ugyanakkor előfordultak hibás esetek is, amikor az algoritmus által detektált forrás iránya nem követte a valódi forrás mozgását, erre mutat példát a következő eset, melyet 50 cm-es mikrofontávolsággal végeztem:



5-11. ábra, valós idejű teszt, 50 cm-es mikrofontávolság

A hangforrás ebben az esetben is  $-60^\circ$ -tól  $60^\circ$ -ig tért ki a mikrofonokhoz képest. A felvétel végén az algoritmus nagy rendszeres hibával mér, szinte ellenkező irányból észleli a forrást. Ennek több oka lehet, egyrészt az olyan korábban tárgyalt jelenségek, mint a hang többutas terjedése, vagy zajjal terheltsége, másrészt a paraméterek nem ideális volta az adott esetben.

Összességében elmondható, hogy a megvalósított alkalmazás többnyire megfelelő pontossággal és megbízhatósággal határozza meg a forrás irányát a mikrofonokhoz képest. Az olyan hatások, mint a terem akusztikája, vagy a hang többutas terjedése okozhatnak olyan jelenségeket, amelyek miatt az algoritmus rosszul detektálja a bejövő hang irányát.

### 5.3. Fúzió hatása

Szakedolgozatom egyik fő témája az volt, hogy a különböző akusztikus lokalizációra használt algoritmusok, mint a korreláció és a beamforming, mennyiben javítják vagy rontják a detektált forrás irányát a valódihoz képest, ha nem külön-külön, hanem egymással pár-

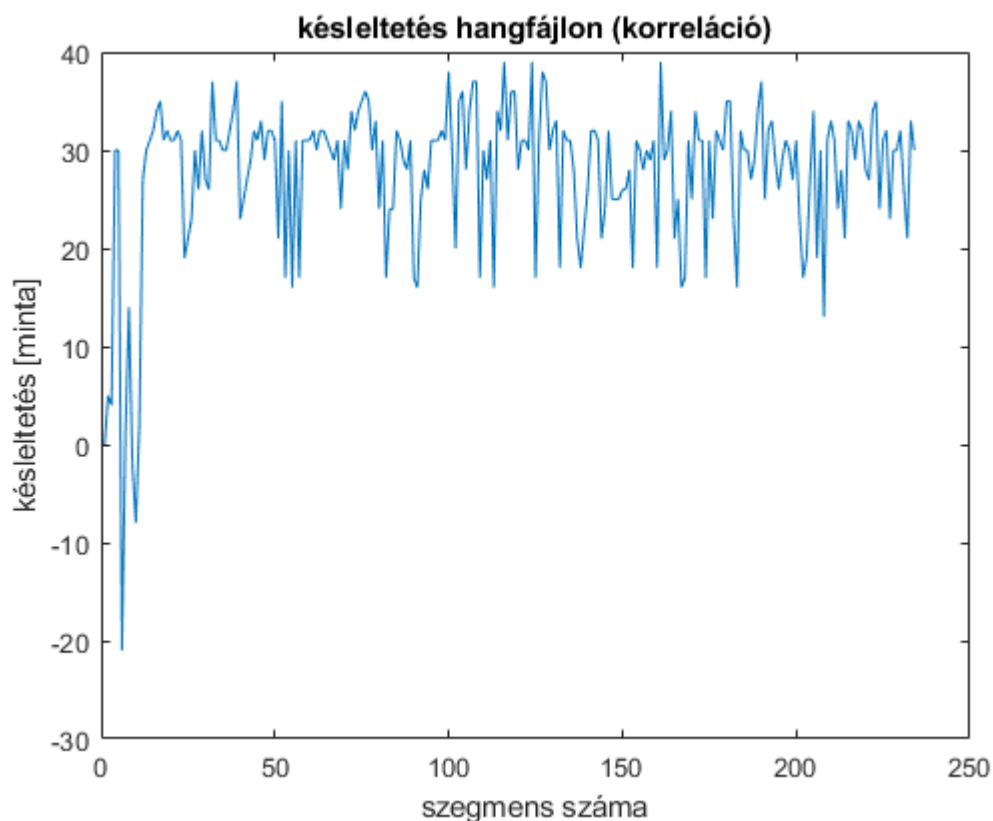
huzamosan használom őket. Ha nem ugyanazon szegmensek esetén lokalizálnak pontatlan pozíciót, akkor az egyes hibák kiegyenlítődnek, nem lesznek olyan nagymértékűek a kiugró eredmények. Ellenben ha mindkét algoritmus helyesen mér, akkor csak robusztusabbá teszik a működést.

Az algoritmusok fúziójának tipikus hatását a következő ábrák szemléltetik:

A mérés 30 cm-es mikrofontávolság során készült, a kis teremben, offline módon, a forrás nem mozgott a felvétel során. A forrás valódi iránya  $52^\circ$ , ami a (3.4) egyenletnek megfelelően körülbelül 23,5 cm-es különbséget jelent a jelutak között. Az (5.1) egyenletnek megfelelően a hang 0,77 cm-t tesz meg két mintavétel között, tehát a késleltetés a két jel között  $\frac{23,5 \text{ cm}}{0,77 \text{ cm}} \approx 30$  minta.

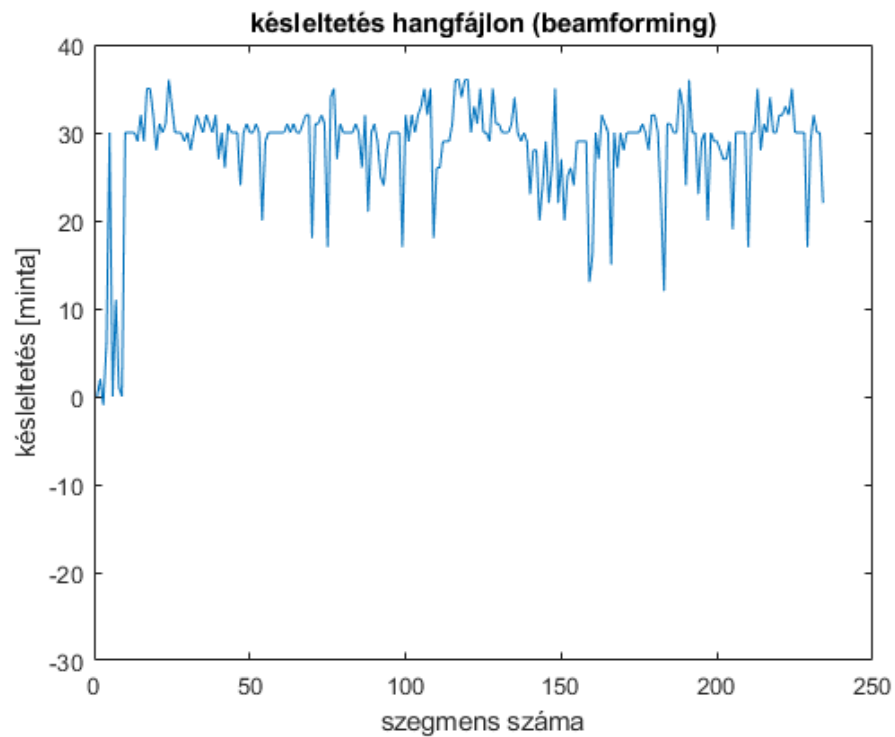
Az x tengelyen a triggerszint és jelminőség vizsgálaton átjutott szegmensek vannak ábrázolva a következő ábrákon.

A korreláció algoritmus által számolt késleltetés mintában:



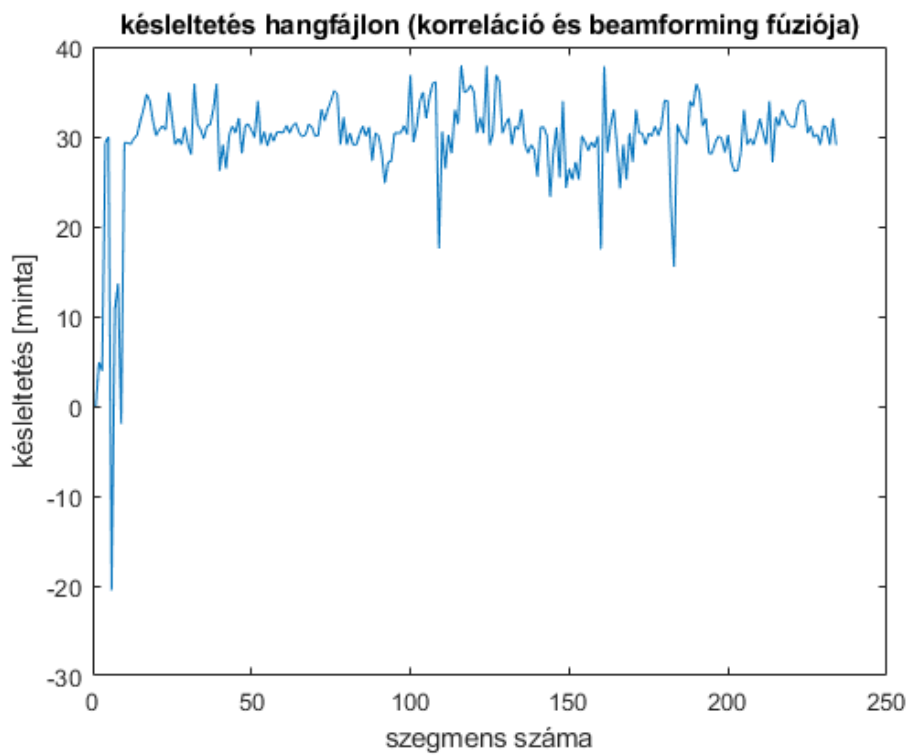
5-12. ábra, korreláció, 30 minta késleltetés a jelutak között

Beamforming által számolt késleltetés mintában:



5-13. ábra, beamforming, 30 minta késleltetés a jelutak között

Eredmények a fúzió után:



5-14. ábra, fúzió után, 30 minta késleltetés a jelutak között

A fúzió hatásának összefoglalása érdekében az egyes algoritmusok és a fúziójuk során ismertetett grafikonokon (5-12., 5-13., 5-14.) szórást számoltam. A szórást a következő képlet segítségével számoltam ki:

$$s_N = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^N (x_i^2 - \bar{x}^2)} \quad (5.5)$$

Ahol:

$N$  – szegmensek száma

$x_i$  – aktuális szegmens által meghatározott irány

$\bar{x}$  – szegmensek által meghatározott irányok mediánja

A fúzió hatását a következő táblázat foglalja össze:

módszer	szórás
korreláció	8,32
beamforming	7,72
fúzió	3,21

5-1. táblázat, fúzió hatása

A fenti táblázatban jól látható, hogy a szórás nagysága kevesebb, mint a felére csökkent a fúzió segítségével, ezáltal robusztusabbá téve az alkalmazást. A lokalizációs algoritmusok párhuzamos alkalmazása más hangfelvételek során is hasonló javulást eredményez, tapasztalataim alapján minden esetben javítja a forrás irányának detektálását.

## 6. Összefoglalás, konklúzió

A félév során sztereo mikrofonok segítségével valósítottam meg valós idejű iránydetektálást emberi beszédhang alapján. Munkám elején megterveztem a forrásazonosítás folyamatát, majd implementáltam azt. Az alkalmazásban a korreláció és a beamforming lokalizációs algoritmusok egymással párhuzamosan működnek a megbízhatóbb működés érdekében. A helymeghatározáshoz szükséges algoritmusok a mintavételezett jeleket egymáshoz képest idő- vagy frekvenciatartományban késleltetik, majd összegzik őket. A forrás pozícióját általános esetben a legnagyobb összeghez tartozó késleltetés adja meg. A módszerek eredményének utófeldolgozásával, illetve fúziójával nagyságrendben sikerült javítanom a lokalizáció robusztusságán. A munkám végén teszteltem az alkalmazás működését előre felvett hangfájlokon, illetve valós időben is különböző körülmények között.

Elmondható, hogy a félév során megvalósított alkalmazás a körülményektől kis mértéktől függően, de robusztus működésre képes, bizonyos feltételek teljesülése esetén a hibája elhanyagolhatóan kicsi a forrás méretéhez képest.

## **7. Köszönetnyilvánítás**

Szakedolgozatom végén szeretném megragadni az alkalmat, hogy köszönetet mondjak mindazoknak, akik segítettek munkámban a félév során. Kiemelném konzulensem, dr. Orosz György segítségét és ösztönzését a szakedolgozatom precíz elkészítéséhez.

## 8. Irodalomjegyzék

- [1] Ultrahangos anyagvizsgálat, Wikipédia, 2018. október 8.  
[https://hu.wikipedia.org/wiki/Ultrahangos\\_anyagvizsg%C3%A1lat](https://hu.wikipedia.org/wiki/Ultrahangos_anyagvizsg%C3%A1lat)
- [2] Infrahang, Wikipédia, 2018. október 8.  
<https://hu.wikipedia.org/wiki/Infrahang>
- [3] New LED lights have a bright future for communication, University of California – Riverside, EurekAlert, 2009. augusztus 12.  
[https://www.eurekalert.org/pub\\_releases/2009-08/uoc--nl1081209.php](https://www.eurekalert.org/pub_releases/2009-08/uoc--nl1081209.php)
- [4] Mazan, Filip, Kovarova, Alena, 2015. A Study of Devising Neural Network Based Indoor Localization Using Beacons: First Results. Faculty of Informatics and Information Technologies Slovak University of Technology in Bratislava, Slovakia, [Online]. 6, 1-6. Available at:  
<http://www2.fiit.stuba.sk/~kovarova/files/publications/CISJ%20Vol19%20No1%202015%20Mazan.pdf>
- [5] Measurement Microphone ECM8000, Technical Specification, Version 1.2, 2000. július, [https://downloads.music-group.com/software/behringer/ECM8000/ECM8000\\_C\\_Specs.pdf](https://downloads.music-group.com/software/behringer/ECM8000/ECM8000_C_Specs.pdf)
- [6] M-Audio Fast Track Pro User Manual, Manuals Directory,  
<http://www.manualsdir.com/manuals/117708/m-audio-fast-track-pro.html>
- [7] Audacity 1.3.13. verzió, Audacity,  
<https://old.audacityteam.org/download/features-1.3-a>
- [8] Orosz György, Dr. Sujbert László, 2008. március, Elosztott rendszerek és szenzorhálózatok 2, Budapesti Műszaki és Gazdaságtudományi Egyetem, Mérés-technikai és Információs Tanszék,  
[https://www.mit.bme.hu/system/files/oktatas/targyak/8605/meres10\\_ElosztottRendszerWSN2\\_2012.pdf](https://www.mit.bme.hu/system/files/oktatas/targyak/8605/meres10_ElosztottRendszerWSN2_2012.pdf)
- [9] Frenyó Péter, 2014, Beltéri akusztikus lokalizáció és forrásazonosítás, Budapesti Műszaki és Gazdaságtudományi Egyetem, Méréstechnikai és Információs Tanszék, Msc Diplomatervezés, Budapest, <http://dsp.mit.bme.hu/userfiles/diploma/frenyodiploma14.pdf>
- [10] Delay Sum Beamforming, Dr. Andrew Greensted, 2012. október, The Lab Book Pages,  
<http://www.labbookpages.co.uk/audio/beamforming/delaySum.html>
- [11] Jungdong Jin, Seunghun Jin, SangJun Lee, Hyung Soon Kim, Jong Suk Choi, Munsang Kim, and Jae Wook Jeon, 2014. április, Real-time Sound Localization Using Generalized Cross Correlation Based on 0.13  $\mu\text{m}$  CMOS Process, Journal of Semiconductor Technology and Science, Volume 14, NO.2,

[http://www.jsts.org/html/journal/journal\\_files/2014/04/Year2014Volume14\\_02\\_06.pdf](http://www.jsts.org/html/journal/journal_files/2014/04/Year2014Volume14_02_06.pdf)

- [12] David Kurc, Vaclav Mach, Kristian Orlovsky, Hassan Khaddour, 2013. július, Sound source localization with DAS beamforming method using small number of microphones, 2013 36th International Conference on Telecommunications and Signal Processing (TSP), Róma, Olaszország, <https://ieeexplore.ieee.org/document/6613988>
- [13] J. N. Moutinho,, R. E. Araújo, D. Freitas, 2014. Indoor localization with audible sound — Towards practical implementation. Pervasive and Mobile Computing, [Online]. 29, 1-16., <https://www.sciencedirect.com/science/article/pii/S157411921500200X>
- [14] Quo, Chen, Muttka, Matt W. , Silent Whistle: Effective Indoor Positioning with Assistance from Acoustic Sensing on Smartphones. Department of Computer Science and Engineering Michigan State University, East Lansing, Michigan, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7974312>
- [15] Ramon F. Brena, Juan Pablo García-Vázquez, Carlos E. Galván-Tejada, David Muñoz-Rodríguez, Cesar Vargas-Rosales and James Fangmeyer Jr., 2017. Evolution of Indoor Positioning Technologies: A Survey. Department of Computer Science and Engineering Michigan State University, Journal of Sensors, 21 pages, <https://www.hindawi.com/journals/js/2017/2630413/>
- [16] Hancong Liu, Sirish Shah, Wei Jiang, 2004. On-line outlier detection and data cleaning. Computers and Chemical Engineering, 28, 1635-1647., [https://sites.ualberta.ca/~slshah/files/on\\_line\\_outlier\\_det.pdf](https://sites.ualberta.ca/~slshah/files/on_line_outlier_det.pdf)
- [17] LaptopOutlet, Logitech C930e USB Full HD Webcam, Built-in dual stereo microphones, 1920 x 1080, <https://www.laptopoutlet.co.uk/logitech-c930e-usb-full-hd-webcam-built-in-dual-stereo-microphones-1920-x-1080.html>
- [18] PCLiquidations, HP ProBook 6570b 15.6" Laptop Core i5-3210M 2.5GHz 4GB Memory 320GB HDD, <https://www.pliquidations.com/p57235-hp-probook-6570b-15#>