



SZÉCHENYI ISTVÁN EGYETEM
MŰSZAKI TUDOMÁNYI KAR
TÁVKÖZLÉSI TANSZÉK



DIPLOMAMUNKA

Virtuális valóság szimulátor programozása és tesztelése MATLAB alatt

Tóth Zoltán

**Villamosmérnöki MSc szak
Távközlési rendszerek és szolgáltatások szakirány**

Győr, 2013

Feladatkiírás

Diplomamunka címe: Virtuális valóság szimulátor programozása és tesztelése MATLAB alatt

Hallgató neve: Tóth Zoltán

Szak: Villamosmérnöki

Képzési szint: MSc

A diplomamunka-feladat leírása:

1. Ellenőrizze és tesztelje a meglévő MATLAB programot hibák után kutatva, az esetleges problémákat felderítve és kijavítva.
2. Különösen ügyeljen és részletesen vizsgálja meg az ITD-információ beállításának lehetőségeit és a csatornák hangerősség-egyensúlyának beállítását.
3. Véglegesítse a programot és készítsen mintahangfájlokat a segítségével.
4. Vizsgálja meg a már meglévő fejhallgató-kiegyenlítő modul integrálásának lehetőségét a programban.

Államvizsga-tárgyak:

- NGM_TA013_1 Digitális műsorszórás
- NGM_TA012_1 Vezeték nélküli és mobil hírközlő rendszerek
- NGM_TA014_1 Optikai hírközlés

Győr, 2013.11.25

belső konzulens

tanszékvezető

Értékelő lap

Diplomamunka címe: Virtuális valóság szimulátor programozása
és tesztelése MATLAB alatt
Hallgató neve: Tóth Zoltán
Szak: Villamosmérnöki
Képzési szint: MSc

A diplomamunka beadható / nem adható be: (a nem kívánt rész törlendő)

dátum

aláírás (belső konzulens)

A diplomamunka bírálatra bocsátható / nem bocsátható: (a nem kívánt rész törlendő)

dátum

aláírás (tanszékvezető)

A bíráló:

- Neve:
- Munkahelye: ...
- Beosztása:...

A bíráló javaslata:

dátum

érdemjegy

aláírás (bíráló vagy tszv.)

A belső konzulens javaslata:

dátum

érdemjegy

aláírás (belső konzulens)

A ZVB döntése:

dátum

érdemjegy

aláírás (ZVB elnök)

Nyilatkozat

Alulírott, Tóth Zoltán, Villamosmérnöki, MSc szakos hallgató kijelentem, hogy a Virtuális valóság szimulátor programozása és tesztelése MATLAB alatt című diplomamunka feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

—
dátum

hallgató aláírása

Összefoglaló

A diplomamunka célja volt egy olyan szimulációs programot megvalósítani, amely képes arra, hogy egy egyszerű, sztereó fülhallgató segítségével, térhallás érzetet keltsen a hallgató személyben. Ehhez rendelkezésemre állt egy ALICE elnevezésű kezdetleges állapotban lévő program.

Így, elsődleges feladataim között egyaránt szerepelt, hogy, hogy elsajátítsam a diplomamunka megírásához szükséges elméleti háttérrel és, hogy részletesen megismerkedjek a MATLAB program felhasználási lehetőségeivel. A MATLAB program egy mátrix műveleteken alapuló programozási nyelv is, melynek elsajátítása a diplomamunka befejezéséig tartott, ismereteim széles körű bővítését követelte meg. Az Alice MATLAB-program gondolatmenetét és forráskódját sikeresen áttanulmányoztam és megértettem. Az elsajátított szakirodalom, és az általam megállapított gyakorlati következtetések alapján, az Alice szimulációs program hibáit sikerrel felderítettem és kijavítottam. Mindig, a felmerülő új igényekhez alkalmazkodva bővítettem és fejlesztettem a szimulációs programot, mely az eredeti szimulációs programot messzemenően felülmúlja.

A jelenlegi Alice program, most már többszörösen ellenőrzöten, a helyes elméletet alkalmazza, mind a HRTF szűrések megvalósításánál, mind az ITD információk előállításánál és felhasználásánál. Továbbá, a horizontális síkon túl, képes a mediális síkban is szimulációkat végezni. Képes, különböző HRTF készleteket kezelni, és ezen felül lehetőséget ad, a szimulációkhoz szükséges bemeneti paraméterek igen nagy skáláját ellenőrizni, beállítani. Számos hibavédelmi, és felhasználóbarát funkcióval bővítettem a programot, annak érdekében, hogy a felhasználása stabil és hibamentes lehessen. További funkciókat is implementáltam, amelyek a távolabbi jövőben tervezett kutatási vizsgálatokhoz lesznek felhasználhatóak.

Továbbá a program forráskódjának a szerkezeti felépítése és átalakítása közben ügyeltem arra, hogy hagyjak lehetőséget a szimulációs program későbbi bővítésére.

Summary

Programming and testing virtual reality simulator in MATLAB

The purpose of the thesis was to realize a simulator program which is able to create the perception of spatial hearing for listener with the help of a simple stereo headphone. To do this, a program was available to me in a rough state, called ALICE.

Thus, my primary responsibilities were to acquire the necessary theoretical background and to know in detail the potential use of the MATLAB software. MATLAB program is a programming language based on matrix operations. To gathering knowledge in MATLAB lasted acquiring the diploma work has been completed. Alice MATLAB program logic of the source code and successfully studied and understood. According to the learned literature, and the practical implications, the errors of the simulation program successfully reconnoitered and corrected. I always adapted the new needs and I developed the simulation program according the new requirements. The last version of Alice is far superior to the original simulation program.

The current program, Alice, now multiply controlled, apply the correct theory and implementation of the HRTF filtering and preparation and usage of ITD information. Also, the program able to execute simulations in the horizontal plane and in the medial plane too. Able to handle different HRTF sets and additionally gives you the ability to control and set a large range of input parameters needed for the simulations. I improved a several fault protection and user-friendly functions for program, in order to allow the user to use the program stable and error free. Other functions are also implemented, which designed to be used in the further future research studies.

In addition, during the construction and transformation of source code I did take care about to leave an opportunity for future expansions and developments of the simulation program.

Tartalomjegyzék:

Feladatkiírás	2
Értékelő lap.....	3
Nyilatkozat	4
Összefoglaló.....	5
Summary	6
Tartalomjegyzék:	7
1. A diplomamunka feladata.....	9
2. A térhallás	11
3. Függvény-leírási módszerek.....	14
3.1 Az időtartomány	14
3.2 A rendszerek leírása	15
3.3 A frekvencia tartomány	16
3.4 Az ablakozó függvények.....	17
3.5 A bizonytalansági elv.....	19
3.6 A csoportfutási idő.....	20
4. Definíciók	21
4.1 A tér síkjai.....	21
4.2 A HRTF-függvény	23
4.2 Az interaurális információk	25
5. A HRTF-függvények mérései	26
5.1 A HRTF-függvények csoportosítása a mérési körülmények szerint:.....	30
5.2 A műfejes technika előnyei és hátrányai	32

6. Az ITD vizsgálata.....	34
6.1 Kuhn egyszerűsített modellje:	34
6.2 Kuhn, elevációs szöget is tartalmazó modellje:	35
6.3 A Woodworth formula:.....	35
6.4 A formulák összehasonlítása.....	35
6.5 További kettő db formula: Larcher and Jot, Savioja	38
6.6 Az ALICE programban eredetileg használt formula	39
6.7 A különböző ITD számoló formulák implementálása a programba.....	41
7. Alice – A szimulációs program vizsgálata, fejlesztése.....	46
7.1 Az Alice futtatása	46
7.2 A bemeneti hangfájl kiválasztása.....	49
7.3 A fejhallgató kiegyenlítés.....	50
7.4 A hangforrás irányának kiválasztása.....	53
7.5 A szimuláció személyre szabása.....	56
7.6 A reflexiók szimulálása.....	57
7.7 Process - a szimulációs számítás indítása	63
7.8 A HRTF szűrés vizsgálata	65
7.9 Az FFT függvény vizsgálata.....	67
7.10 A „wav gyártás” felgyorsítása	74
7.11 További hibajavítások, „debug-olás”, hangerősség-ingadozás	76
7.12 az Alice grafikus felületének végleges kinézete.....	79
Irodalomjegyzék:	80
Mellékletek:	82

1. A diplomamunka feladata

Céлом, egy olyan programot megvalósítani, amely képes arra, hogy egy egyszerű, sztereó fülhallgató segítségével, térhallás érzetet keltsen a hallgató személyben.

A feladat megoldásához adott egy már meglévő program, ALICE. A programot egy francia cserediák írta, Dr. Wersényi György megbízásából. A program ugyancsak abból a célból íródott, ami a szakdolgozat célja is, de nem tudjuk, hogy helyesen működik-e, illetve, hogy ami változtatásokat végez egy-egy hangfájlon az jó-e, illetve milyen elméletet ültet át a gyakorlatba. Ezek után az elsődleges feladatom az volt, hogy elsajátítsam az elméleti alapokat, majd megvizsgáljam az Alice nevezetű program működését.

Az elméleti alapok megismerését, az ember biológiai felépítéséből adódó sajátosságok tanulmányozásával kezdtem. A hallórendszer a felépítéséből adódóan képes rengeteg információt kinyerni a közvetlen környezetéből. Ilyen információ a hangforrás térbeli elhelyezkedése is. A térhallás vizsgálata ugyancsak fontos eleme a szakdolgozatomnak, hiszen így merülnek fel olyan témakörök, mint például a különböző jelfeldolgozási módszerek, idő- és frekvenciatartományban végezhető műveletek. A térhallás tanulmányozása során rengeteg definícióval ismerkedtem meg, amelyek egy részét bele is foglaltam a szakdolgozatba, hiszen ismeretük szükséges ahhoz, hogy megértsük a térhallás-szimulációs programom műveleteinek mértékét. Ilyenek például a térkülönböző síkjaiban történő vizsgálatok, a HRTF-függvények használata és használhatósága. Az ITD információ és a felhasználhatóságának feltételei.

Az ALICE programhoz nem tartozik számottevő leírás, tehát egyedüli kiindulási alap, csakis a forráskód lehet. A forráskódban találhatóak kommentek, amik viszonylag kevés segítséget képesek nyújtani. Ennek oka nagy részben abból adódik, hogy egy harmadik nyelvű gondolatmenetet először angolra fordítottak, majd ezt magyarul próbáljuk értelmezni.

Ezen felül további nehézséget okoz, hogy a grafikus interface is csak forráskód formájában van meg. Általános esetben, a Matlab-ban is, amikor egy programot fejlesztünk, akkor a forráskódtól jól elkülönülten tudjuk fejleszteni a grafikus felületet. Ilyenkor egy „képet” látunk magunk előtt a grafikus felületről, amit alkotóelemeire lehet bontani, és elemenként átméretezni, áthelyezni, funkciókat rendelni hozzájuk. Ezt a „képet” egy .fig kiterjesztésű fájlban tárolhatjuk. Miután ez a felület elkészült, illetve folyamatosan a készítés során, a fordító program egy szöveges fájlt generál (.m file), ami teljes egészében leírja a grafikus felületet. A program futásakor ezt a szöveges fájlt értelmezi a Matlab és újra létrehozza a futtatható objektumokat. Amennyiben a .fig kiterjesztésű fájl nem áll rendelkezésre, a grafikus felület nem szerkeszthető a könnyedebb, interaktív módon, különböző varázslók (wizard) segítségével. A további szerkesztésre az .m file-on keresztül lesz lehetőség, ami meglehetősen lassú és nehézkes. Esetemben, pontosan ilyen módon volt csak lehetőség a grafikus felület szerkesztésére, mert az a fájl (.fig), ami lehetővé tenné a grafikus interface grafikus szerkesztését, egyszerűen hiányzik.

A programot sok új funkcióval kell kiegészíteni, illetve a már meglevőket kijavítani. Az ALICE program felhasználhatóságát is növelni kell, továbbá szeretném szélesíteni a felhasználhatósági területet úgy, hogy minél több lehetőség maradjon a később felmerülő igények implementálására.

2. A térhallás

Már nagyon egyszerű, hétköznapi megfigyelések alapján is feltételezhetjük, hogy a térhallás jelensége létezik. Például, ha az úttesten kelünk át, akkor pusztán a hallásunk alapján nem csak azt tudjuk megállapítani, hogy autó közeledik, hanem azt is igen nagy biztonsággal meg tudjuk mondani, hogy melyik irányból. Minden ember képes arra, hogy egy adott pontossággal meg tudja becsülni egy-egy hangforrás elhelyezkedését az őt körülvevő térben. Ez a pontosság minden embernél más és más, ráadásul sok tényező befolyásolhatja. Befolyásoló tényezők lehetnek ugyanúgy a hangforrás egyes tulajdonságai, mint a hallgató személy egyéni, a lokalizációban szerzett tapasztalatai [1].

A lokalizációnak nevezzük azt a folyamatot, amelynek során a hallgató személy, a hallottak alapján meghatározza a hangforrás helyét. Ennek a helymeghatározásnak a pontatlanságát nevezzük lokalizációs bizonytalanságnak. Ez egy olyan küszöbérték, amely alatt, ha változnak a hallott hang térbeli jellemzői, akkor ezt a változást nem képes a hallórendszer érzékelni. Ez a definíció azt sugallja, hogy a térhallás tulajdonságait kutató vizsgálatok a hangforrás helyének változását tekintik bemenő paraméternek. Ez csak részben igaz, hiszen léteznek olyan vizsgálatok is, amikor szándékosan olyan hangforrásokat szimulálunk egymás után, amelyek a térben egymástól távol helyezkednek el. Már ezen vizsgálati módszerek alapján is a térbeli hallásvizsgálatokat, alapvetően két külön csoportra lehet és szokás osztani. Az egyik csoport az, amikor a kísérleti alanytól azt várjuk, hogy határozza meg egy hangforrás helyét a térben. Ezt hívjuk abszolút vizsgálatoknak. A vizsgálatok másik csoportja, amikor az alany feladata az, hogy a hangforrás elmozdulásának mértékét határozza meg. Ezek a relatív vizsgálatok [3].

A lokalizáció művelete azon a tényen alapszik, hogy a hangforrás hangja különböző torzulásokon, változásokon megy keresztül, amíg eljut a forrástól a hallgató személyig. A térhallás szempontjából, az egész átviteli útnak fontos szerepe van. Mint azt majd később is láthatjuk, például, a hangforrás távolsága is hatással van az átviteli

függvényre. Részben azért, mert a levegő aluláteresztő szűrőként viselkedik, ami nagyobb távolságok esetén jobban érvényesül. Elsők között azonban, azt kell megvizsgálni, hogy a számos hatás közül, melyek azok, amik a térhallás szempontjából a legfontosabbak, illetve melyek azok, amik esetleg a térhallás folyamatát a leginkább akadályozzák.

A vizsgálatok során a legfontosabb paraméter az átviteli útnak a végén előállt hangnyomás időfüggvénye. Ezt a hangnyomás időfüggvényt a dobhártyán, illetve a dobhártya helyén értelmezzük és füljelnek nevezzük. A füljel az a mechanikus rezgés, amit a dobhártya továbbít a középső fül felé, és amit a mikrofon elektromos jellé alakít át. A füljel időfüggvénye hordozza azt az információt, amit később kiértékelve, következtetéseket lehet levonni a hangforrás helyét illetően. A füljel, bizonyos jellemzők több csoportját is hordozza, hordozhatja.

A monaurális jellemzők például azok a jellemzők, amiket egy darab (jobb, vagy bal) füljelből is ki lehet nyerni. Vannak olyan információk, amiket csak két füljelből lehet kinyerni, ezek a binaurális jellemzők. Ebben az esetben, általában a két füljel különbségei a fontosak, ezek az interaurális jellemzők. Ilyen például a két jel beérkezési idejének különbsége.

Néhány további fontos paraméter, amit a lokalizációhoz használ fel a hallás:

- ITD Interaural Time Delay vagy Difference – a füljelek közötti interaurális időkülönbség
- ILD Interaural Level vagy Intensity Difference – a füljelek közötti interaurális szintkülönbség
- IPD Interaural Phase Difference – a füljelek közötti interaurális fáziskülönbség. Ez ritkán használatos, mivel az ITD-vel rokon fogalom
- A hangforrás hangjelén, a külső fül további spektrális szűrést végez, ami hangszínmódosulást eredményez, ami szintén irányfüggő. Ennek a jelentősége akkor számottevőbb, ha interaurális különbségek nincsenek, vagy ellentmondásosak.

Ezek a jellemzők mind frekvencia és irányfüggők, tehát az is, hogy melyik paraméter mikor játszik domináns szerepet a lokalizációban. Például, ha a vízszintes síkot tekintjük, akkor alacsony frekvenciákon IPD, közepes frekvenciákon ITD, magas frekvenciákon pedig az ILD információk alapján történik az iránybecslés.

3. Függvény-leírási módszerek

A vizsgálódásunk egy jelentős hányada arra irányul, hogy kiderítsük, hogy milyen változásokat kell egy adott hangon alkalmazni, ahhoz, hogy az a hang egy előre megtervezett irányból hallatszódjon. Ennek a célnak az eléréséhez nem elég azt ismerni, hogy melyek ezek a változások, hanem azt is, hogy ezek közül melyik megvalósítható, és milyen módszerrel. A módszerek igencsak szerteágazóak lehetnek, de szinte mindegyikük kivitelezhető valamelyik számítógépes szoftverrel, amelyek között akár ingyeneseket is számba vehetünk.

3.1 Az időtartomány

A hangjelek, amelyeken a változtatásokat végezzük, időfüggvények, így kézenfekvőnek tűnhet, hogy a rajtuk végzett változtatásokat is az időtartományban alkalmazzuk. Azonban ezeknek a változtatásoknak a matematikai megvalósítása bonyolult, ami nagy számítási kapacitást igényel. Több alternatív vizsgálati módszer is létezik, például a Hilbert-transzformáció, vagy Wavelet-análízis, de a legelterjedtebb a Fourier transzformáció. A Wavelet és a Hilbert módszerek úgynevezett gömbi alapfüggvények szerinti sorfejtéssel dolgoznak. A Wavelet analízissel lehetőség van arra, hogy az adatfolyamot változó felbontással kezeljük a frekvenciában, azáltal szélesítve a felhasználhatósága körét.

$$f(t) = \frac{4}{\pi} (\sin(\omega t) + \frac{1}{3}\sin(3\omega t) + \frac{1}{5}\sin(5\omega t) + \frac{1}{7}\sin(7\omega t) + \dots)$$

1. ábra: A négyszögjel Fourier sora

A Fourier sorfejtés azt a tényt használja ki, hogy a periodikus jelek felbonthatóak alapfüggvények súlyozott összegére, így az különböző amplitúdóval, frekvenciával és fázissal rendelkező szinuszos jeleket ír le.

3.2 A rendszerek leírása

A Fourier-transzformáció az időtartomány béli függvényről készít egy frekvencia tartomány béli leírást. Ezt más néven spektrálanalízisnek is hívják. A megfelelő matematikai szabályokat, és feltételeket betartva az idő – frekvencia, és frekvencia – idő tartományok közötti átjárás, nem csak lehetséges, de kölcsönösen egyértelmű leírást eredményez. A Fourier-transzformáció a valós időfüggvényből egy komplex frekvenciatartomány béli leírást készít, tehát komplex számokat ad vissza, algebrai alakban. Az algebrai alakot további átalakításnak szoktak alávetni, általában kinyerik az abszolút értéket, és fázis információt és azt használják a számítások során.

$$F(j\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (F(t) * e^{-j\omega t}) dt$$

2. ábra: A folytonos Fourier transzformáció

A gyakorlati felhasználások esetén, nagyon gyakran nem folytonos időben vizsgálódnak, hanem az idő diszkrét pillanataiban. Ez lehetőséget ad arra, hogy egyszerűbb számításokat, közelítéseket alkalmazzanak. Ilyen eljárás a Diszkrét Fourier Transzformáció (DFT), és a Fast Fourier Transformation (FFT). Az FFT eljárás gyorsabb számításokat tesz lehetővé, mint a DFT. A számítógépes FFT eljárások a kettő egész számú hatványának megfelelő pontban számolják ki a spektrumot.

$$X(k) = \frac{1}{N} \sum_{i=0}^{N-1} x(i) e^{-j\frac{2\pi}{N}ik}$$

3. ábra: A diszkrét Fourier transzformáció

A Fourier-transzformációs eljárásokat nem csak egy-egy függvény felbontására használják, hanem nagyon jól használhatók, például átviteli rendszerek leírására. Az időtartományban egy rendszert egyértelműen meghatároz az impulzusválasza. Az

impulzusválasz az a kimeneti időfüggvény, amit az adott rendszer az impulzusos gerjesztésre ad válaszként. Ilyen gerjesztés például a Dirac-delta, ami nagyon rövid idő alatt, nullához közelít, végtelen nagyra tekinthető energiát ad a rendszernek. Egy ilyen gerjesztés elméletben a spektrum minden pontján azonos amplitúdóval rendelkező komponensekből áll. Ennek az a haszna, hogy a vizsgált átviteli rendszert minden frekvencián azonos mértékben gerjeszti. Ilyen tökéletes vizsgálójelet előállítani a gyakorlatban nem lehet, csak megközelíteni. Éppen ezért az átviteli rendszerek vizsgálatára más típusú vizsgálójelek is léteznek, amik könnyebben kivitelezhetőek, ugyanakkor a vizsgált rendszer válaszából, impulzusválaszt további számítások árán lehet csak megkapni. Az impulzusgerjesztés legnagyobb előnye, hogy a hatására közvetlenül az impulzusválaszt lehet megmérni a rendszer kimenetén.

3.3 A frekvencia tartomány

Az átviteli rendszerek egyértelmű leírására nem csak az időtartományban lehetséges, hanem a frekvencia tartományban is. Ez az úgynevezett átviteli függvény. Az impulzusválasz és az átviteli függvény közötti kapcsolatot ugyancsak a Fourier transzformáció adja meg. A transzformáció egyértelmű megfeleltetés a két leírási mód között, és mindkét irányban alkalmazható.

$$A(f) = \int_{-\infty}^{\infty} (a(t) e^{-j2\pi t}) dt \quad a(t) = \int_{-\infty}^{\infty} (A(f) e^{-j2\pi t}) df$$

4. ábra: Az átviteli függvény számítása az impulzusválaszból és fordítva

Általában a gyakorlatban csak periodikus jeleket van értelme a Fourier transzformálni. A periodikus jelek spektruma diszkrét, egymástól jól elkülöníthető frekvenciákon megjelenő spektrumvonalakból áll. Ezeknek a különböző frekvenciájú tagoknak az amplitúdójukkal súlyozott összege pontosan a vizsgált jel Fourier sorát adja. A nem periodikus jelek, mínusz végtelentől plusz végtelenig tartó integráltja, folytonos spektrumot eredményez. A valóságban túlnyomó részt, ilyen nem periodikus

jeleket kell Fourier transzformálni, amely önmagában használhatatlan eredményre vezetne. A megoldás, hogy nagyon rövid, periodikusnak tekinthető időszelletekre kell feldarabolni a vizsgált jelet, és minden ilyen kis darab kvázi periodikus jelhez egy-egy spektrumkép fog tartozni. A feldarabolás ablakozófüggvénnyel való szorzással történik, melynek a vizsgált időintervallumon az amplitúdója nulla és egy közötti szám, azon kívül pedig nulla. Így az integrálást is nullától T ideig kell csak elvégezni, ami diszkrét jelek esetén további egyszerűsítést jelent, hiszen véges sok amplitúdó értékének az összegét kell összeadással kiszámolni.

3.4 Az ablakozó függvények

Az ablakozó függvények kiválasztásánál nagyon körültekintően kell eljárni, ugyanis az ablakozás az egész jelből kivág egy darabot és csak azt vizsgálja tovább, ezzel biztosan hibát visz a számításokba. A hiba fajtája több féle lehet az ablakozó függvények alakjának megfelelően. A négyszög alakú kapuzófüggvény például ideális abból a szempontból, hogy a vizsgált jel alakját nem torzítja, súlyozza a 0-tól T -ig terjedő szakaszon. Ez a függvény leírásából is jól látszik:

$$A(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{egyébként} \end{cases}$$

5. ábra: A négyszögjel leírása

A Fourier transzformáció egyik alapvető tulajdonsága, hogy a műveletekre nézve is egyértelmű. Ez, ebben az esetben azt jelenti, hogy ami az időtartományban szorzás, az Fourier transzformáció után a frekvencia tartományban konvolúció lesz. Tehát, ha az időtartományban egy négyszög alakú kapuzójel szorozódik a bemenő jel időfüggvényével, akkor az a frekvenciatartományban szűrőként fog megjelenni. A szűrés pedig a négyszögjel Fourier transzformáltjával fog történni, aminek a spektruma nagyon széles. Potosan ez az egyik legnagyobb hátránya, egy ilyen alakú jellel történő kapuzásnak, hogy a kapott jel spektruma nagyon elkenődik, széles lesz. Ez a jelenség

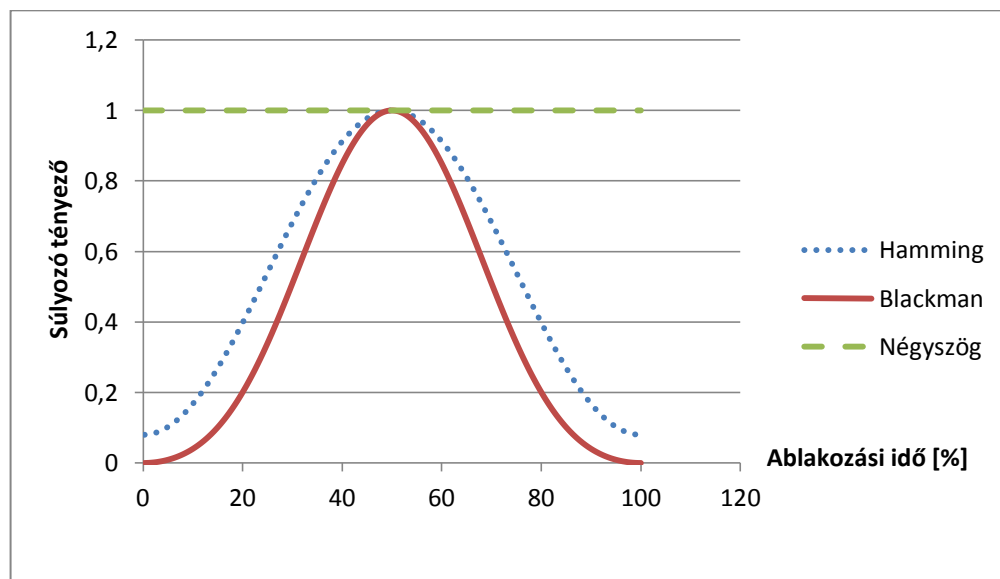
abból adódik, hogy a négyszög alakú függvény szélein, minden periódus után ugrásszerűen változik a súlyozó tényező, azaz a jel amplitúdója, ami igen nagy felharmonikus tartalmat generál.

A kapuzó jelek okozta spektrumbéli elkenődést úgy lehet megváltoztatni esetleg, csökkenteni, hogy nem négyszög alakú függvényt használunk a kapuzáshoz. A nagymértékű spektrumelkenődést a jelben történő ugrásszerű amplitúdó változás okozta. Ennek az elkenődésnek a mértéke hatékonyan csökkenthető, olyan kapuzófüggvény használatával, amely az időben folytonos. Ilyen például a Hamming és a Blackman ablakok:

$$A(t) = \begin{cases} 0,54 - 0,46 \cos\left(\frac{2\pi t}{T}\right), & 0 \leq t \leq T \\ 0, & \text{egyébként} \end{cases}$$

$$A(t) = \begin{cases} 0,42 - 0,5 \cos\left(\frac{2\pi t}{T}\right) + 0,08 \cos\left(\frac{4\pi t}{T}\right), & 0 \leq t \leq T \\ 0, & \text{egyébként} \end{cases}$$

6. ábra: felül: Hamming, alul Blackman ablakozó jelek függvénye



7. ábra: a három említett ablakozó függvény

A Hamming és Blackman ablakoló függvények időbeni folytonossága az folyamatos átmenetet biztosít az átengedett és az elnyomott amplitúdó értékek között. Ezen ablakoló függvények célja, hogy az általuk ablakozott jel spektrumában minimalizálják a legközelebbi, és így a legnagyobb amplitúdóval rendelkező oldalsávot.

3.5 A bizonytalansági elv

Az ablakozó függvényekkel kapcsolatban fontos megemlíteni a bizonytalansági elvet. Ennek értelmében nem lehet tetszőlegesen csökkenteni az ablakozási időt, mert minél rövidebb ideig tart az ablakozás, annál pontatlanabb lesz a kiszámolt spektrumkép. Ez fordítva is igaz. Minél pontosabb spektrumot akarunk kiszámolni annál hosszabb integrálási idővel kell dolgozni. Természetesen a felhasználás függvényében létezik olyan minimum és maximum integrálási idő, amelynél rövidebb illetve hosszabb ideig integrálni pontatlan, illetve fölösleges. Például egy négyszögjel esetében, minél több periódust foglalunk bele az integrálási időbe annál inkább „tisza, vékony” vonalakat lehet látni a spektrumban. Az integrálási idő csökkenésével, például ha ez már kisebb, mint a periódusidő, a spektrumvonalak kiszélesednek és amplitúdójuk egyenletesebb, zajszerűbb lesz. A négyszögjel is inkább egységugrásra illetve impulzusra fog hasonlítani az időtartományban. Összefoglalva, ha olyan ablakozó függvényt használunk, ami az időtartományban ideális, azaz nem súlyozza az időfüggvényt, akkor a spektrumképe lesz olyan, ami a leginkább nem megfelelő. Viszont ha azt akarjuk, hogy a spektrumkép ne kenődjön el akkor olyan ablakozó jelet kell választani, ami az időtartományban súlyozza a vizsgált jelet.

Léteznek továbbá frekvencia tartománybeli ablaktípusok is, melyek ugyancsak lehetnek káros és hasznos időtartomány bel hatásai. Ilyen például:

- Lapos, a frekvenciatartományban megfelelően keskeny, de az időtartományban súlyoz
- Hann, súlyozza a spektrum bel komponenseket, de cserében ez módosítja a legkevésbé az időtartomány bel képet

- Butterworth, módosítja a idő és a spektrum beli képet is de külön-külön, egyiket sem annyira mint az előző kettő. Ez kompromisszum lehet a Lapos és a Hann megoldások között.

3.6 A csoportfutási idő

Egy rendszer átviteli függvényének nagyon fontos része a fáziskarakterisztika, vagy más néven fázisspektrum. A fázisspektrum frekvencia szerinti deriváltja a burkolókésleltetés függvényét határozza meg, amit más néven csoportfutási időnek is neveznek.

Ezt a függvényt szintén a frekvencia függvényében kell ábrázolni, a függőleges tengely pedig idő dimenziójú. Ha a csoportfutási idő függvény konstans nulla vagy t , akkor a vizsgált rendszernek a bemenetére adott jel késleltetés nélkül, illetve t idő múlva, megjelenik a kimenetén, az amplitúdókarakteristikájának megfelelően. Ha ez a görbe szigorúan monoton növekvő, akkor az azt jelenti, hogy az alacsonyabb frekvenciákat lassabban, a magasabb frekvenciákat pedig gyorsabban jeleníti meg a kimenetén. A stúdiótechnikai alkalmazásokban a csoportfutási idő frekvenciafüggése nagyon nemkívánatos jelenség, ezért a leggyakrabban ezt ki is egyenlítik. Ezt, egy nulla erősítésű és ellentétes csoportfutási idővel rendelkező rendszerrel szokták megtenni, ez a futásidő korrekció.

$$T_{futás}(f) = \frac{1}{2\pi} \frac{d\varphi(f)}{df}$$

8. ábra: A csoportfutási idő a fáziskarakterisztika frekvencia szerinti deriváltja

A HRTF-függvényeket, átviteli rendszerként értelmezve, szintén meg kell vizsgálni a csoportfutási idő karakterisztikáját. Ha a HRTF-függvények ezen tulajdonsága valamilyen oknál fogva nemkívánatos, akkor kompenzálni kell. A másik megoldás, hogy HRTF-függvények fázisinformációját figyelmen kívül hagyjuk, és valamilyen más módszerrel helyettesítjük.

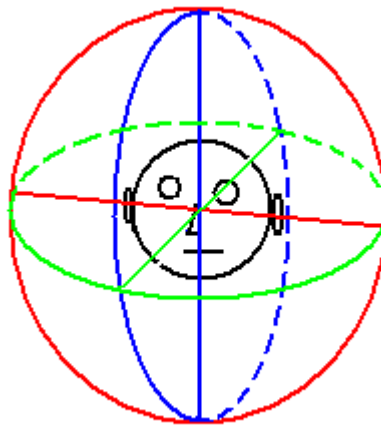
4. Definíciók

4.1 A tér síkjai

Mivel irányfüggő vizsgálatokkal foglalkozunk, ezért mindenképp definiálni kell, hogy milyen vonatkoztatási rendszerben értelmezzük az irányokat. Ez a vonatkoztatási rendszer szükségszerűen háromdimenziós, és praktikusan a középpontja mindig fej középpontja. Ez a fejhez kötött, más néven head-related koordináta rendszer.

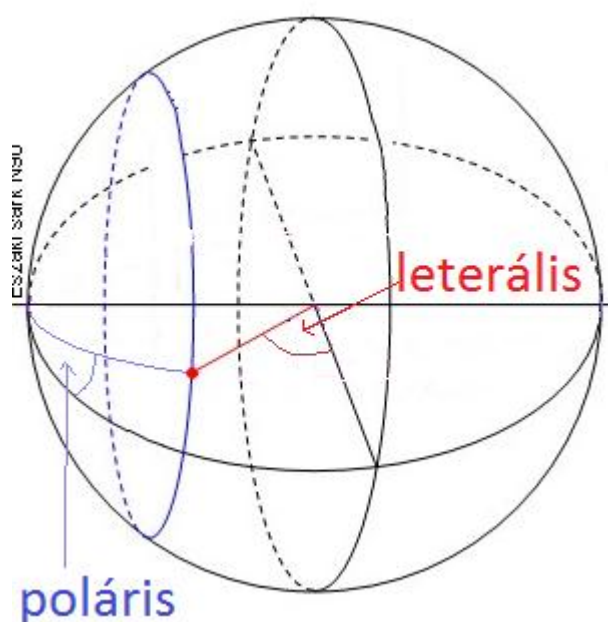
Három darab nevezetes síkja van:

- A mediális, más néven szagittális sík pontosan a fej szimmetriasíkjára illeszkedik.
- A horizontális sík, a füleket összekötő képzeletbeli egyenes, és a szemgödör alsó csontjaira fektetett sík.
- A frontális sík, a hallójáratok elülső peremét köti össze, és merőleges az első két síkra, amelyek szintén merőlegesek egymásra.



9. ábra: kék: mediális, zöld: horizontális, piros: frontális sík

A síkok metszéspontja az origó, ami a fej belsejében található. A mérések során a hangforrást az origótól állandó távolságra helyezzük el, amit r -rel jelölünk. A vizsgálatok túlnyomó hányadában a hangforrást vagy csak a mediális, vagy csak a horizontális síkban mozgatjuk. A horizontális sík béli elmozdulást ϕ -vel jelöljük, és a „pontosan szemben” irányhoz viszonyítjuk. Az óramutató járásával megegyező irány a pozitív irányú kitérés, ezzel ellentétesen a negatív irány. A mediális sík béli kitérést δ -val jelöljük. A $\delta = 0^\circ$ a vízszintes előre, $\delta = 180^\circ$ a vízszintesen hátra irány. δ szöget elevációs, ϕ szöget azimuth, ezt a koordináta rendszert pedig „hoop” koordináta rendszernek is nevezzük [4].



10. ábra: a laterális-poláris koordináta rendszer

Egy másik fontos koordináta rendszer a laterális-poláris koordináta rendszer. Ebben a laterális szög adja meg a horizontális sík béli eltérést a mediális síktól. A hangforrás ebben a koordináta rendszerben is állandó, r távolságra helyezkedik el az origótól. Így, a laterális szög kijelöl egy, a mediális síkkal párhuzamos kört. A polár szög, más néven sarki szög, pedig ezen a kijelölt körön adja meg a vízszintestől való eltérés irányát.

4.2 A HRTF-függvény

A hangnyomás átviteli függvényét a hangforrástól a dobhártyáig értelmezzük, és függ a test, a fej, a fülkagyló, a hallójárat, és a dobhártya impedanciájának együttes hatásától. A szabad tértől a dobhártyáig értelmezett átviteli függvényt az angol irodalmakban Head Related Transfer Function néven említik. Ennek a rövidítését a HRTF szót, a magyar irodalmakban is használják [6]. A HRTF-függvény lényegében egy szűrő, ami leírja, hogy miként módosul a szabad tér felől érkező hangjel a frekvencia függvényében. A HRTF tehát komplex átviteli függvény, ami amplitúdó és fázisinformációt is tartalmaz. Nagyon sok esetben azonban, csak az amplitúdó információt használják fel, és a fázisinformációt más módon adják a rendszerhez. A HRTF ilyen módon azt adja meg, hogy a fejünk hatására miként módosul a hangnyomás a koordináta rendszer origójában, ahhoz képest, ha a tér teljesen üres. A HRTF definíciója tehát:

$$HRTF = \frac{P_1(j\omega)}{P_2(j\omega)}$$

11. ábra: a HRTF-függvény definíciója

P_1 a hangnyomás a dobhártyán, P_2 pedig a hangnyomás a koordináta rendszer origójában ugyan azzal a hangjellel és hangnyomással mérve. P_2 -t irányfüggetlen mikrofonnal kell mérni, ugyanis ez lesz a referenciajel, P_2 hangnyomást ehhez viszonyítjuk. Kicsit másképp fogalmazva: a kezdetben irányfüggetlen mikrofonunk iránykarakterisztikája a hallórendszerünk hatására minden irányban más érzékenységgű lesz. Ezt a karakterisztika változást tároljuk a HRTF-függvényben. minden irányhoz más HRTF tartozik, így végtelen sok HRTF-függvény létezik, amit a gyakorlatban véges sok függvénnyel közelítünk.

A Head Related Impulse Response, HRIR az időtartományban írja le azokat az átviteli függvényeket, amiket a HRTF a frekvenciatartományban. A két leírási mód azonos értékű és egyik a másikba átszámolható. Mint minden átviteli rendszerben,

ebben is a komplex átviteli függvény és az impulzusválasz között Fourier-transzformáció teremti meg a kapcsolatot. Tehát ha a komplex HRTF-függvényt visszatranszformáljuk az időtartományba, akkor a rendszer impulzusválaszát, azaz a HRIR-t kapjuk.

Az átviteli függvények rögzítéséhez több féle referenciajelet lehet használni. Ennek három ismertebb csoportja a következők: (referenciajel: P_2 , hangnyomás a hallójáratban: P_1)

1. Szabadtéri: Ez lényegében megfelel a korábban leírtaknak. Tetszőleges irányokból működik, a hangforrás állandó távolságban helyezkedik el a mérés végéig. A referenciajelet a koordináta rendszer központjában kell rögzíteni mindegyik irányból, úgy hogy a tér üres. A fej elhelyezése után a hangnyomást a dobhártya helyén kell mérni. Kényelmi és kivitelezhetőségi szempontok miatt azonban szokás ezt a hallójárat bementén mérni, ugyanis a hallójáratnak, a beérkező jel irányinformáció tartalmára csak elhanyagolható hatása van.
2. Monaurális: Az előző esetben minden irányban felvett P_1 jelhez tartozott egy P_1 referenciajel. Monaurális esetben viszont a tetszőleges irányból felvett P_1 -hez ugyan az a P_2 referenciajel tartozik, és ez lesz a viszonyítási alap. Ez a kitüntetett referenciajel a $\phi = \delta = 0^\circ$ -os irányhoz tartozik. Itt tehát átviteli függvény alatt azt értjük, hogy az adott irányban P_1 -ben milyen spektrális eltérések vannak P_2 -hez képest. P_1 nem csak a $\phi = \delta = 0^\circ$ -os irányhoz tartozó referencia lehet, hanem diffúztéri referencia is. A módszer előnye, hogy a P_1 és a P_2 jel felvétele is a fül ugyanazon pontján történik. Így az előző esetben elhanyagolt hallójárat átviteli függvényt ki lehet küszöbölni a vizsgálatból.
3. Interaurális: Ebben az esetben a két fül jelét hasonlítjuk össze, és nincs harmadik jel, amit referenciaként használnánk. Az egyik fülét kinevezzük referenciajelnek, és a másik fül jelét pedig ehhez képest vizsgáljuk. A referenciajel így minden irányból más lesz, hiszen mindkét fül átviteli karakterisztikája irányfüggő. Az Interaurális átviteli függvényt Monaurális módon rögzített átviteli függvényekből számolni is lehet:

$$HRTF_{interaurális}(f, \varphi = 30^\circ) = \frac{P_1(f, \varphi = 330^\circ)}{P_1(f, \varphi = 30^\circ)}$$

12. ábra: az interaurális HRTF kiszámolható két monaurális HRTF hányadosaként

4.2 Az interaurális információk

A térhallás interaurális információszerzése alapvetően arra épül, hogy kihasználja az ember azon tulajdonságát, hogy két füllel rendelkezik. A fülek elhelyezkedéséből adódóan pedig a két füljel között különbségek lépnek fel, melyek közül a térhallás szempontjából az interaurális különbségek a legfontosabbak, ezek közül is az ITD és ILD információ.

A monaurális jellemzők azok, amiket az egy füllel történő hallás esetén is fel tudunk ismerni. Ezekből a jellemzőkből lehet a forrás távolságára, magasságára következtetni és az irányára is.

Azonban, előfordulhatnak olyan esetek is, amikor az ITD információ hiányzik, vagy ellentmondásos. Ilyen például a mediális síkban való mozgás, vagy a hallászavar kúpján való elmozdulás. A HRTF-függvények hangszínszűrő hatása azonban ilyenkor is jelen van, amiből irányinformációk kinyerhetők. A HRTF-függvények is félrevezetőek lehetnek a szimulációk során. Ez abból adódik, hogy minden egyes ember és vizsgálati alany a saját individuális HRTF-függvényeit használja a mindennapi lokalizáció során, viszont a szimulációk során általában egy másik ember, vagy eszköz individuális HRTF-függvényeit használjuk.

5. A HRTF-függvények mérései

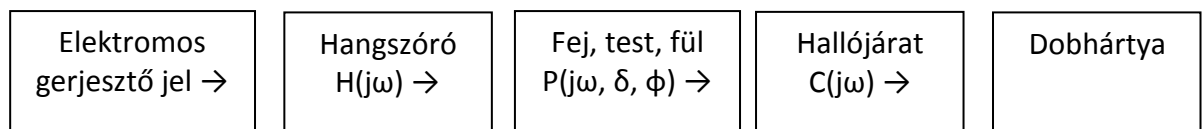
A nagy pontosságú HRTF-függvények rögzítése nagyon fontos feladat. A HRTF-függvények már önmagukban is elég információt hordoznak ahhoz, hogy a hallórendszer meg tudja határozni a hangforrás helyét. Ez azért van így, mert a HRTF-függvény magában foglalja a frekvenciafüggő amplitúdó fázisinformációkat. Az amplitúdó menet tartalmazza az ILD információt, fázismenet információja pedig tartalmazza az ITD információkat is. Ahhoz, hogy a lokalizáció helyes lehessen, minden térirányhoz külön-külön meg kell határozni a HRTF-függvényt. A komplex HRTF-függvény információtartalmát a HRIR függvény is tartalmazza, de az időtartományban. A HRTF fázisinformációja a HRIR függvénynél időbeli eltolásként jelentkezik. A HRTF rendszerek közelíthetőek minimálfázisú rendszereként, ami azt jelenti, hogy azt elég csak az amplitúdómenettel megadni, az fázisinformációt pedig elég utólag hozzáadni a rendszerhez. Ekkor a fázisinformációt ITD információként kezeljük, és a hallgató személy fejméretéből becsüljük. Az ITD információ értékének becsülésére több módszer is létezik, amit később ismertetek. [19]

A minimálfázisú rendszer legfontosabb tulajdonsága, hogy a lehetséges változatok közül mindig a legkisebb fáziseltolást adja. A Fáziskarakterisztika és az amplitúdókarakterisztika viszonya rögzített, így elég csak az utóbbi ismerete. A nem minimálfázisú rendszerek felbonthatók egy minimálfázisú és egy mindentáteresztő tag szorzatára. A mindentáteresztő tag egységnyi erősítéssel rendelkezik. Ezeket futásidő korrektoroknak is szokás nevezni. [19]

A HRTF-függvény alapvetően egy irányfüggő szűrő ami, definíció szerint a hangforrástól a dobhártyáig tartó átviteli utat írja le. Így a HRTF-függvényeket a dobhártya helyén kell mérni, rögzíteni. A dobhártya helyén mikrofont elhelyezni azonban nagyon nehéz feladat, ami pontatlanságot visz a mérésekbe [7]. A hallójárat és az azt lezáró dobhártya akusztikai szempontból nem elhanyagolható, viszont a hangnyomásfüggvény irányinformáció tartalmára elhanyagolható hatása van, így kijelenthetjük, hogy a hallójáratban való hangterjedés irányfüggetlen. Ennek az oka,

hogy a hallójárat méreteinél fogva csak longitudinális terjedési mód lép fel, és körülbelül 17 kHz-ig a hallójárat bemenete pontforrásként viselkedik, és továbbítja a hanghullámokat a hallójáratba. Szintén irányfüggetlen tényező a sugárzási impedancia és a hallójárat bemeneti impedanciája közötti hányados is. Így a HRTF méréseknél megengedhető, hogy a mikrofon a hallójárat bemeneténél helyezkedjen el. A mérési pontosság tovább növelhető, ha a hallójárat bemenetét lezárjuk, mert így a különböző személyek közötti változékonyság csökken (closed-canal).

Az átviteli utat három főbb részre bonthatjuk. Az első az elektromos gerjesztő jel és a szabad tér közötti hangsugárzó, amelynek az átvitele nem garantált, de ennek a pontossága nem is feltétlenül szigorú követelmény.



13. ábra: az átviteli út felosztása

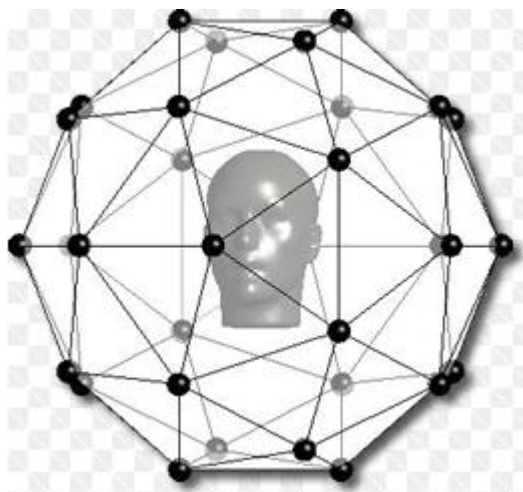
A HRTF-függvények az átviteli utat a szabad tértől a dobhártyáig írják le, amit további két részben lehet szemléltetni. Az egyik a hallójárat bemenetétől a dobhártyáig, amit az irányfüggetlensége miatt elhanyagolunk. A második a szabadterétől a lezárt hallójárat bemenetéig, ami erősen irányfüggő.

A HRTF-függvények szabad térben értelmezettek, tehát a méréseket süketszobában kell végezni, hogy minél jobban megközelítsük a szabadtéri terjedés tulajdonságait. Sajnálatos módon a süketszobák nem ideálisak, így hibát vihetnek a mérési eredményekbe, amiket különböző módokon lehet feljavítani. HRIR méréseknél például számottevő hibát okozhatnak a süketszoba falairól érkező visszaverődések. Ezeket úgy lehet kiküszöbölni, ha az eredmények végéből egyszerűen levágjuk a visszaverődéseket. Ez gyakorlatilag azt jelenti, hogy a hangforrás fizikai távolságából adódó terjedési időn belül érkező jeleket tekintjük hasznosnak, a többi pedig valamilyen visszaverődésből adódik, amiket eltávolítunk a HRIR-ből [8].

A HRTF-függvények mérésénél, alapvetően a HRTF definíciójából indulunk ki, mely szerint a HRTF-függvény, két hangnyomásfüggvény hányadosával egyezik meg. A hányados nevezőjében a referencijel található, amit az üres koordináta rendszer origójában mértünk. A hányados számlálójában pedig a mérendő személy jelenléte által módosított hangnyomásfüggvény található. Ennek a módszernek nagy előnye, hogy a mindkét jel tartalmazza az egész átviteli lánc tulajdonságait, így amikor a HRTF kiszámolása történik, az osztás során a közös részek kioltásra kerülnek. Ezért nem kritikus kérdés például a hangsugárzó átviteli tulajdonsága. A káros visszaverődések viszont már nem ennyire állandó szabályosan, és állandó irányba befolyásolják a mérési eredményt, így az nem garantált, hogy ezek is kioltódnak az osztás művelete során.

A mérésekben használt hangsugárzók vihetnek olyan pontatlanságot is a mérési eredményekbe, amit a HRTF számolásánál az osztás művelete nem olt ki, esetleg még erősít is. A hangsugárzók átvitele nem teljesen lineáris, az alsó és felső határfrekvenciájuk alatti és feletti frekvenciákat egye kisebb teljesítménnyel képesek kiadni. E miatt a jel-zaj viszonyuk megnő, és ez rontja a HRTF frekvenciában értelmezett érvényességét. Hasonló problémák adódhatnak a lejátszás során is. Így, utólagos számítások alapján jelkompenzálás, szűrés szükséges lehet.

Közelítéri HRTF-ek mérésére létezik egy nagyon másik eljárás, amit reciprocitás, vagy transzfer mérési elvnek neveznek [9]. Ekkor a hang terjedési iránya nem a szabadterétől a dobhártya felé, hanem a hallójárat bemenetétől a szabadter felé. Ezt úgy valósítják meg, hogy a hallójárat bemenetéhez helyeznek egy-egy hangszórót, és a mérendő személy vagy műfej köré pedig mikrofonrendszert helyeznek. Ez a fordított elrendezés a Helmholtz-egyenlet reciprocitásából fakadóan engedhető meg. A módszer előnye, hogy közelítéri HRTF-eket is pontosan lehet mérni, 250 Hz - 13 kHz-ig jó jel-zaj viszony mellett. A hátrány azonban, hogy 400 Hz alatt a mérés pontatlan, főleg a hangszórók kis fizikai méretéből fakadóan.



143. ábra: A reciprocitás elve. A fekete pontok a mikrofonokat jelképezik [20].

Ugyancsak fontos lépés megállapítani azt, hogy milyen felbontást akarunk elérni a HRTF-függvények segítségével. Természetesen az ideális az lenne, ha a végtelen sok térirányhoz, végtelen sok HRTF-függvény állna rendelkezésre. Azonban a végtelen sok mérés nem csak megoldhatatlan, de az emberi térhallás felbontóképességénél jobb felbontás nem is indokolt. A HRTF-függvények szükséges számát és felbontását az alapján is lehet vizsgálni, hogy a spektrumképük mennyire hasonlítanak egymásra. Végeredményben ez azért érdekes mert ezek a különbségek alapján dől el, hogy a hallórendszer tud-e különbséget tenni az egyes térirányok között vagy sem. Egy vizsgálat [10] célja az volt, hogy megállapítsák, hogy mi az a legkisebb eltérés a HRTF-függvények spektrumában, ami már hallható különbséget okoz. A hangforrások azonos ITD-vel, de különböző HRTF szűréssel rendelkeztek. A vizsgálat eredménye alapján a függőleges síkban sokkal érzékenyebb a hallás a HRTF változásokra, mint a vízszintes síkban. Ennek az is lehet az oka, hogy függőleges síkban csak ez az információ áll rendelkezésre, tehát erre van specializálódva a hallórendszer. Másik vizsgálatok [11] kimutatták, hogy minél nagyobb emelkedési szögél található a hangforrás, annál kisebb térbeli felbontás is elegendő.

A HRTF méréseknél figyelni kell arra, hogy a mérési eredménybe belekerül az a terjedési idő is, ami ahhoz kell, hogy a hang a hangsugárzóból a mikrofonhoz érjen. A

mérések célja ezzel ellentétben csak az ITD információ kinyerése. Egy megoldás szerint [6] a válaszfüggvényt szét kell bontani egy mindentáteresztő és egy utána kapcsolt minimálfázisú rendszerre. A mindentáteresztő tagból el kell távolítani a lineáris késleltetést, és újra egyesíteni a minimálfázisú taggal. A mérések azt mutatják [12], hogy az ilyen módon kapott fázis megkülönböztethetetlen csak minimálfázisú tag fázisától, tehát a HRTF-függvények rendszerét minimálfázisú rendszernek tekinthetjük.

5.1 A HRTF-függvények csoportosítása a mérési körülmények szerint:

- **individuális HRTF:** Egy HRTF készletet akkor hívunk individuálisnak, ha a kísérletben résztvevő alany, ugyan azoknak a HRTF-függvényeknek a szűrőhatását hallja amiket az ő saját fejével rögzítettek.
- **nem individuális HRTF:** az ilyen HRTF-eket nem a hallgató alany, hanem egy másik ember fejével rögzítették. Ezt tovább lehet bontani:
 - ellenőrzöten jó lokalizációs képességgel rendelkező alany fejével,
 - átlagos geometriai méretekkel rendelkező emberi fejjel,
 - véletlenül kiválasztott emberi fejjel felvett HRTF készlet.
- **Perszonalizált vagyis egyénreszabott, személyreszabott HRTF:** ennek a módszernek az alapvető tulajdonsága, hogy megpróbál individualizálni egy általánosan rögzített HRTF-et. Ebben az esetben rendelkezésre kell álljon, egy nem individuális HRTF készlet, amit utólag ki kell egészíteni olyan információkkal amik a mérést végző alanyra jellemzőek. Ez általában az ITD információ hozzáadását jelenti. Az ITD információt tehát nem bonyolult mérési eljárással kell kinyerni, hanem az alany fizikai méreteit lemérve, és behelyettesítve egy-egy formulába az ITD értéke személyre szabottan számolható. Ilyen fizikai méretek elsősorban a hallójárat bemeneteinek távolsága, a fej mérete, és a fülkagyló mérete. A perszonalizált HRTF-ek előállítására több módszer is létezik, és kísérletek bizonyítják, hogy az alanyok

jobban tudtak lokalizálni a személyre szabott HRTF-ekkel, mint a nem személyre szabott HRTF-ekkel [13].

- Optimálisan kiválasztott HRTF: Ilyenkor egy előre meglévő HRTF adatbázisból lehet válogatni, hogy a hallgató melyik HRTF-ek segítségével tud a legjobban lokalizálni. Ez a módszer is tekinthető személyre szabottnak, azonban lényeges különbség, hogy nem az alany fizikai méretei a meghatározóak, hanem csak a lokalizációs képesség. A módszer használatában a legnagyobb nehézséget az jelenti, hogy mi alapján történjen a hozzárendelés a hallgató alany és a számára legoptimálisabb HRTF között. Az összes HRTF végigpróbálása igen időigényes feladat lenne. Ezért dolgoztak ki különböző eljárásokat arra, hogy a kiválasztás minél hatékonyabb legyen. Ilyen eljárás a DOMISO, és az úgynevezett svájci rendszerű kiesés. A módszer érdekessége, hogy voltak olyan kísérletek, amelyekben az egyes alanyok jobb lokalizációt értek el egy jól kiválasztott HRTF készlettel, mint a saját individuális készletükkel [14]. A módszer eredményei tovább javíthatóak az ITD személyre szabásával, a fejátmérő, vagy a csoportfutási idő alapján.
- Műfejes HRTF: ebben az esetben nem emberi alannal történik a HRTF-ek rögzítése, hanem egy olyan eszközzel, ami az emberi testet hivatott helyettesíteni. A műfejeknek igen sok fajtája létezik. Van olyan, ami ténylegesen csak egy gömbalak, van ami fülkagylóval van ellátva, tartalmazhat szájszimulátort, orrot, szemgödröt, nyakat, vagy akár egy egész torzót. A modernebb eszközöket, fej és torzószimulátornak hívják. Angol elnevezéssel: Head And Torso simulator – HAT. Mindegyik eszköz közös jellemzője, hogy a lehető legpontosabban próbálják meg modellezni az emberi testet alakban, méretben, és leginkább az akusztikai tulajdonságaiban. Sok esetben, esetleg kiegészítik ruházattal, parókával, szemüveggel.

5.2 A műfejes technika előnyei és hátrányai

Az embereken végzett mérések egyik nagy hátránya, hogy nehézkes és pontatlan. A pontatlanság leginkább az alanyok reflexszerű mozgásából adódik. Valamint ha a mérendő személyek egyszerre töltik be a mérőeszköz és a mérés tárgya szerepét is, akkor kénytelenek vagyunk elfogadni a megfigyeléseiket, amik nagyon szubjektívak lehetnek. Ezek számszerűsítése nehézkes és nem is mindig lehetséges. Ezzel szemben a műfej a mérés időtartama alatt nem mozdul el, és mivel minden tulajdonsága, és mérete állandó az időben, így a rajta végzett mérés könnyen reprodukálható. További előnye a műfej használatának, hogy hosszú mérések is megvalósíthatók, és a nagy átlagolásszámú mérések megnövelik a pontosságot.



15. ábra: Műfejes mérésekhez használt eszközök [15], [16]

Mégis még a modern műfejek anyaga hőmérséklete, geometriája, és akusztikai tulajdonsága is eltér az emberi fej tulajdonságaitól, ami jelentős hiányosságot okoz a műfejes méréssel készült HRTF-függvények funkcionalitásában. Egy kísérlet [17] eredménye szerint a saját hallás 15%-os a véletlenszerűen kiválasztott emberen készült HRTF 36%-os, az átlagos emberi fejen 22%-os, a műfejes méréssel felvett HRTF-ek használata esetén pedig 40-50% a hibás lokalizációk aránya.

Összességében tehát bármelyik emberi fejhez tartozó HRTF készlet használata pontosabb lokalizációt eredményez, mint a műfejhez tartozó HRTF használata [17].

Ezért ha a felhasználás célja a virtuális hangtér szimulálása, akkor nem műfejjel felvett HRTF-eket célszerű használni. Ha a mérés célja a pontosság, és a HRTF-ek összehasonlítása, akkor viszont a műfejjel készült HRTF-függvények használata az indokolt.

6. Az ITD vizsgálata

Az ITD, Interaural Time Delay-t azaz Interaurális időkülönbséget jelent. Ez azt a jelenséget takarja, amikor a különböző irányokból érkező hangok, zajok, különböző időpontokban érkeznek a hallgató jobb és bal fülébe.

Az ITD érték meghatározásához különböző formulák állnak rendelkezésre, melyeket általában tapasztalati, vagy elméleti úton fejlesztettek ki. Ebben a fejezetben különböző formulákat ismertetek és hasonlítok össze több szempontból is.

Az első összehasonlításban az ITD megállapításához négy, ismertebb formula is rendelkezésemre állt. Ezek közül hármat neveznek Kuhn-formulának és egyet pedig Woodworth-formulának.

6.1 Kuhn egyszerűsített modellje:

A Kuhn-féle formulák közül az elsőt 500Hz alatti, a másodikat pedig 2000Hz feletti frekvenciákra ajánlják. Ezeket, Kuhn egyszerűsített modellje alapján dolgozta ki, ahol gömb alakú fejet feltételezett.

$$ITD_{kiszf} \approx \frac{3a}{c} \sin \varphi \qquad ITD_{nagyf} \approx \frac{2a}{c} \sin \varphi$$

16. ábra: Kuhn egyszerűsített modellje
 $f < 500\text{Hz}$

17. ábra: Kuhn egyszerűsített modellje
 $f > 2000\text{Hz}$

Ebben a modellben az 500Hz és a 2000Hz közötti frekvenciákon az ITD frekvenciafüggő, enyhén csökkenő meredekséggel. 500Hz alatt és 2000Hz felett a frekvenciafüggés elhanyagolható.

6.2 Kuhn, elevációs szöget is tartalmazó modellje:

A harmadik Kuhn-formula, a teljes hangfrekvencia tartományban alkalmazható, és ezen felül nem csak az azimuth, hanem az elevációs szöget is meg lehet adni bemenő paraméterként.

$$ITD \approx \frac{a\varphi}{c} (\arcsin(\cos \delta \sin \varphi) + \cos \delta \sin \varphi)$$

18. ábra: Kuhn elevációs szöget is tartalmazó modellje

6.3 A Woodworth formula:

Egy másik gyakran használt formula a Woodworth- formula, mely szintén a teljes hangfrekvenciás sávban használható, és elevációs szöget is tartalmaz.

$$ITD = \frac{d(\varphi + \sin \varphi) \cos \delta}{2c}$$

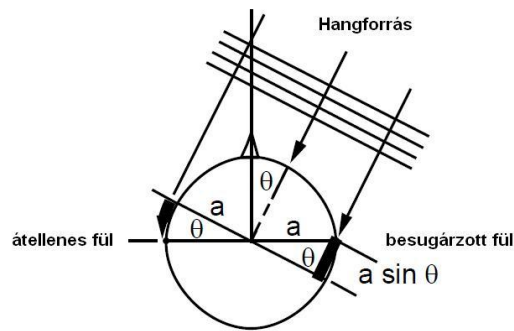
19. ábra: A Woodworth formula

A fenti képletekben használt jelölések: d a fejtátmérő, a a fej sugara, c a hangsebesség, φ az azimuth, δ pedig az elevációs szög.

6.4 A formulák összehasonlítása

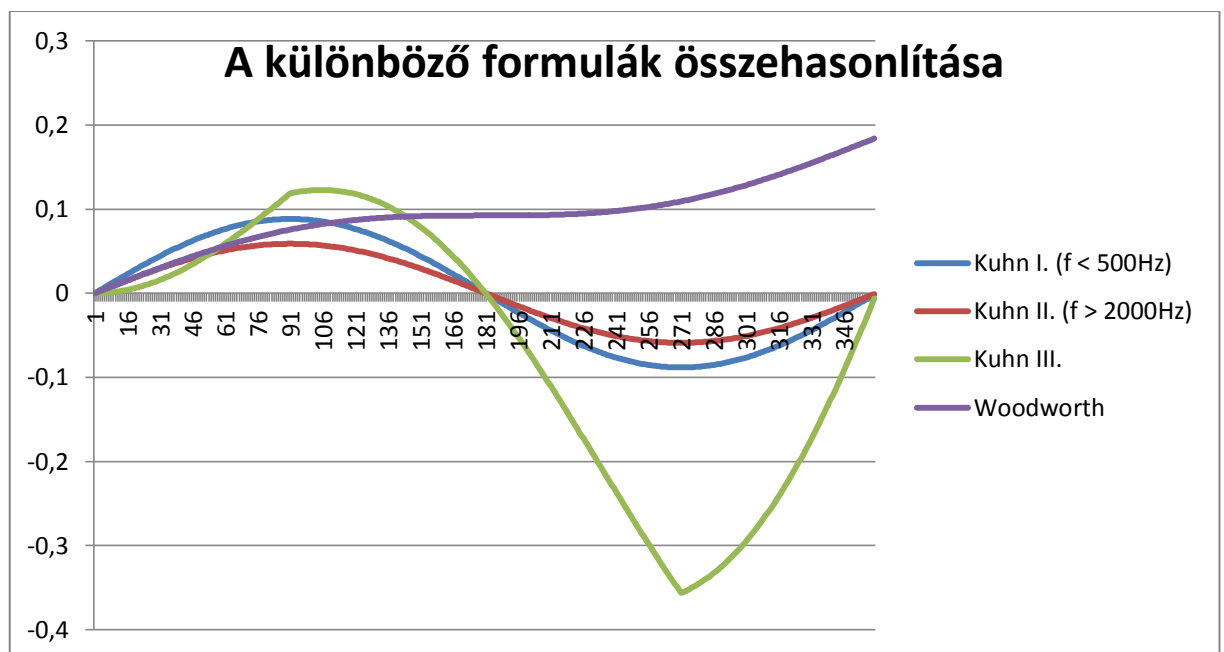
A különböző formulák, nyilván nem egyforma pontossággal közelítik a valóságot. Ezért ugyan azokra a bemenő adatokra más-más ITD értéket fognak visszaadni. Ebben a fejezetben, a különböző formulákkal számolt, különböző ITD értékeket fogom összehasonlítani.

Először is, az összehasonlításban állandó d , c és δ értékeket fogok használni. A változó érték a φ lesz. A φ szöget az 5. ábrának megfelelően fogom mérni.



20. ábra: A hangforrás irányának jelölése

Első alkalommal, a szögeértékeket $0^\circ - 360^\circ$ -ig helyettesítettem be a különböző formulákba. Az eredmény:

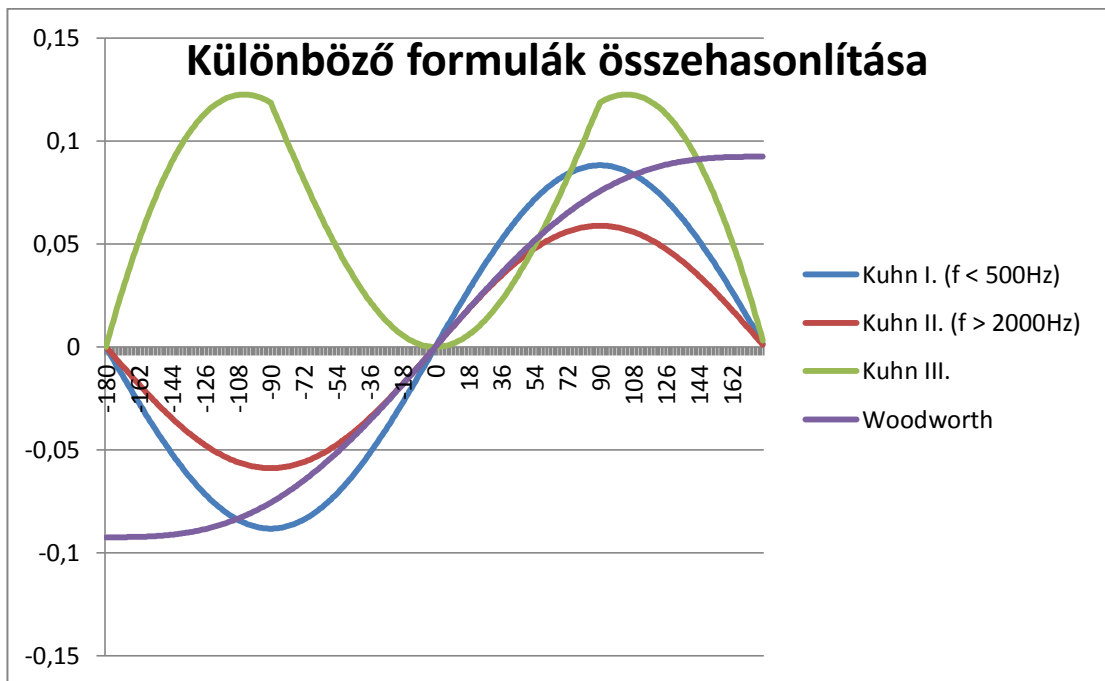


21. ábra: a szögeértékek behelyettesítése a formulákba $0^\circ - 360^\circ$ -ig

Erről a diagramról a következő megállapításokat tettem. Mind a négy esetben, $0^\circ - 90^\circ$ -ig a φ szög növelése az ITD növekedését okozza. A $90^\circ - 180^\circ$ -ig terjedő tartományban az ITD nek csökkennie kell, ez azonban csak a Kuhn formuláknál teljesül. Woodworth formula esetén valószínűleg a $90^\circ - 180^\circ$ -ig tartományt egy $90^\circ - 0^\circ$ -ig

terjedő tartománnyal kell helyettesíteni. A Kuhn I. és II. formulák jellege, a várakozásnak megfelelően, megegyeznek, 1,5 szerez konstans szorzó segítségével, a Kuhn II.-ből Kuhn I. megkapható. Kuhn III. formula esetén a $180^\circ - 360^\circ$ -ig terjedő tartományhoz tartozó ITD-eket is a $0^\circ - 180^\circ$ -ig terjedő tartományhoz tartozó ITD-kből kell meghatározni, azért, hogy a ITD „szimmetrikus” legyen.

Ha a szögtartományt $-180^\circ - +180^\circ$ -ig választom meg, akkor az eredmény:



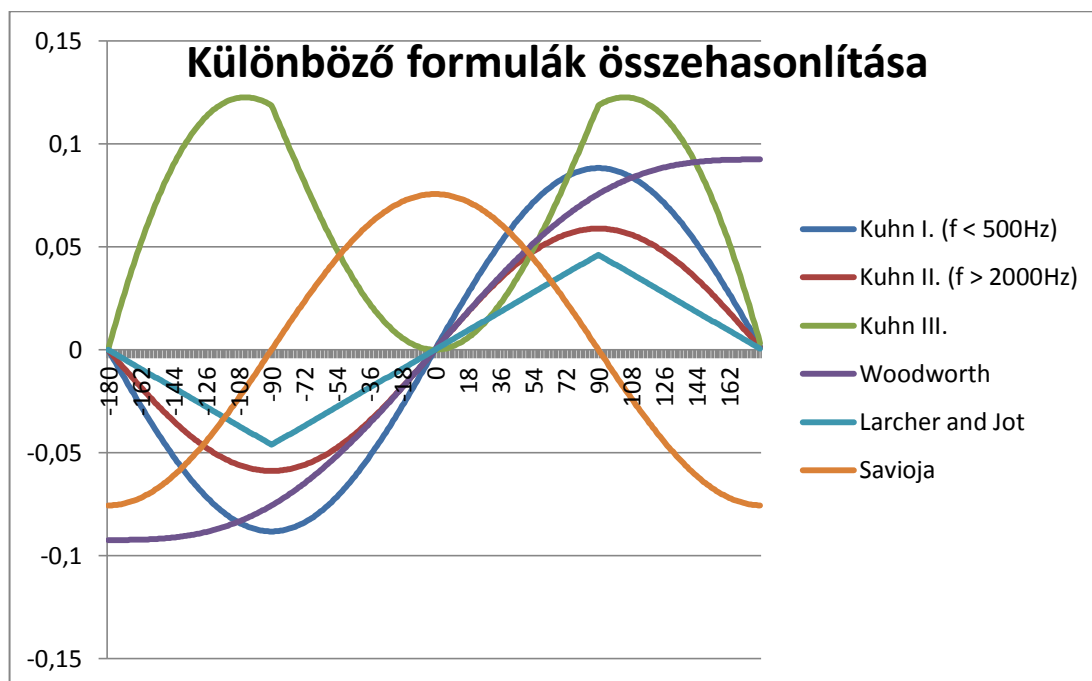
22. ábra: a szögtértékek behelyettesítése a formulákba $-180^\circ - +180^\circ$ -ig

Ennél a diagramnál a következő megállapítások tehetők. Mind a négy formula „szimmetrikus eredményt” ad. Woodworth formula esetében, a képlet csak a $-90^\circ - +90^\circ$ -ig adhat jó eredményt, mert -90° alatt, és $+90^\circ$ felett az ITD értékének csökkennie kell. $0^\circ - 90^\circ$ -ig a Kuhn I., Kuhn II., és Woodworth görbék jellege hasonló, csökkenő meredekséget mutat, míg a Kuhn III. görbe meredeksége növekvő. Kuhn III. görbétől eltekintve a görbék előjelhelyes eredményt adnak.

A két diagramból az is megállapítható, hogy szinte mindegyik formulával más-más szögekhez tartozó ITD értékek számolhatók ki. Ezekből az eredményekből további egyszerű számolásokkal meghatározhatóak a „kimaradó” szögekhez tartozó ITD

értékek is. Ezeket az eltérő számolási módszereket be kell építeni a szimulációs programba, annak érdekében, hogy a különböző formulák pontossági sorrendje megállapítható legyen.

6.5 További kettő db formula: Larcher and Jot, Savioja



23. ábra: a szögértékek behelyettesítése a formulákba -180° - $+180^\circ$ -ig, kiegészítve a Larcher and Jot és a Savioja formulákkal

A Larcher and Jot formula meglehetősen egyszerű végeredményt ad. 0° - 90° -ig a görbe lineáris, monoton növekszik. 90° - 180° -ig is lineáris, monoton csökken. Ez, negatív szögtartományban is megállapítható, ráadásul előjel helyesen.

A Savioja formula ránézésre rossz eredményeket ad. Azonban a görbe „alakja” megegyezik a Kuhn I. és Kuhn II. görbékkel. Ha a Savioja formulába φ szög helyett $\varphi - 90^\circ$ -ot helyettesítünk be akkor a görbe null helye is a megfelelő, $\varphi = 0^\circ$ fokhoz kerül.

6.6 Az ALICE programban eredetileg használt formula

Az ALICE program jelenleg, egy adott módon képes elvégezni ezeket a számolásokat. A számolás menete kódolva van a programban. Jelenleg, ha a számolás mikéntjén változtatni szeretnék, akkor a program forráskódjában kell megkeresni a megfelelő részt és átírni. Ez a módszer meglehetősen lassú és nehézkes. Ezért a vizsgálat meggyorsítása érdekében át fogom írni a programot. A célom, hogy egyszerűen és gyorsan lehessen váltani az ITD számoló formulák közt a program grafikus felületének segítségével. Első lépésként megkeresem a forráskódban azt a kódrészt, amelyikben az ITD értékének számolása történik. Sokáig a `delayAux.m` függvényben kerestem ezt a kódrészletet. Részben meg is találtam, viszont nagyon sok próbálkozás után sem működött, úgy ahogyan kellett volna. Ennek több oka is van egyszerre. Az ITD számolása nem ebben a részben kezdődik. A `Process.m` függvényben is vannak részeredmények, amiket az ITD számolásnál is fel kell használni. A `Process.m` függvényben van meghívva az `aplDelay.m` függvény. Itt a `delay.m` függvény van meghívva, majd ebben pedig a `delayAux.m` függvény. Ez a gondolatmenet, bizonyos körülmények között érthető, de felesleges ennyire többszintű függvényrendszert alkalmazni. Végül a `delayAux.m` függvényben egy olyan, ITD számolási módszer található, ami egyik általam vizsgált formulára sem hasonlít. Ez a formula a következő:

$$abs\left(\frac{\frac{d}{2} * \sin(angle)}{\sin\left(\frac{angle}{2}\right)}\right) * \frac{Fs}{340} - abs\left(\frac{\frac{d}{2} * \sin(180 - angle)}{\sin\left(\frac{180 - angle}{2}\right)}\right) * \frac{Fs}{340}$$

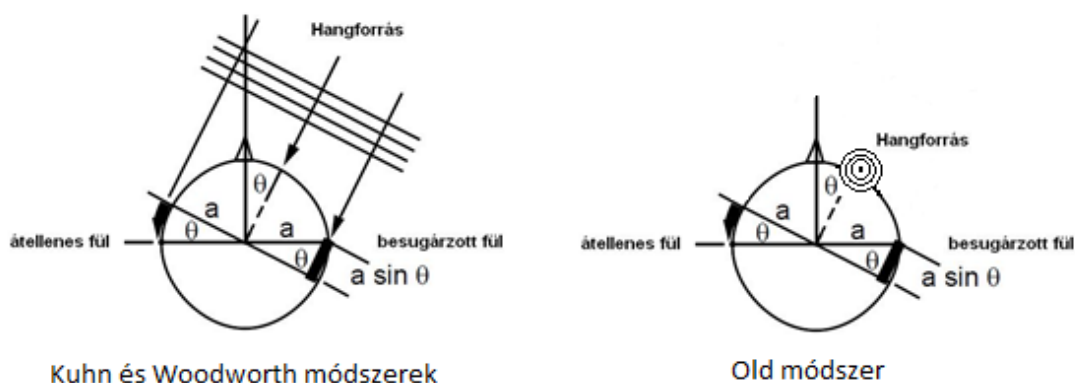
24. ábra: az Old formula vizsgálata

Ezt a formulát Old formulának, a módszert pedig old módszernek neveztem el, azért, hogy a későbbiekben ezen a néven hivatkozhatassak rá. Ebben a formájában nem található meg a programban. Ebben a formájába én állítottam össze, annak érdekében, hogy jól átláthatóan szemléltetni tudjam, a fentebb említett négy darab MATLAB függvény működését, és célját.

A formula vizsgálatakor több szembetűnő hibával is találkoztam. Az első, legfontosabb az F_s nevű változó (aminek konstans értéke van) jelenléte. A változó értéke 44100, ami a bemeneti jel mintavételi frekvenciájával egyezik meg. Visszakövetve a változó értékeit kiderül, hogy ez a változó, ténylegesen ezt a kezdeti bemenő információt hordozza. Így ebben a formulában, a bemeneti hangjel mintavételi frekvenciája befolyásolja a jobb és bal fül között fellépő időbeli eltérés nagyságát. Ez biztosan helytelen megoldáshoz vezet.

A formulából nem derül ki, azonban a vizsgált programrészlet arra enged következtetni, hogy a késleltetés értékét két különálló terjedési idő különbségéből számolja ki. Ezek, a bal és a jobb fülhöz tartozó, terjedési idők, a referenciapont, pedig az orr pontja.

Összehasonlítva az Old módszert a Kuhn és Woodworth módszerekkel szintén egy hibát lehet felfedezni. A Kuhn és Woodworth módszerek egy távoli hangforrást feltételeznek, ami a gyakorlatban azt jelenti, hogy síkhullámokkal számolnak. Az Old módszer viszont akkor adhat helyes eredményt, ha abból a feltételezésből indul ki, hogy a hangforrás a fej kerületénél helyezkedik el.



25. ábra: A hangforrások feltételezett helye

Ez az esetek többségében nem igaz, és a mi vizsgálataink sem erre irányulnak. A módszer további feltűnő tulajdonsága, hogy bonyolult. Ezt nem nevezem hátránynak,

ugyanis miután implementálva van a programba, ez nem hátráltatja a felhasználót a program használatában.

A formula további és talán legfontosabb hibája, hogy nullához nagyon közeli számokat ad vissza ITD értékeként, ami azt jelenti, hogy a formulával kiszámolt késleltetés olyan kicsi, hogy a fül már nem tud időbeli különbséget tenni a két jel között.

Ezért úgy döntöttem, hogy ezt az ITD számolási módszert, mint lehetőséget benne hagyom a programban, mert a későbbiekben további vizsgálatok elvégzésére még alkalmas lehet, miután kijavítottuk a hibáit. További vizsgálat lehet például az az eset, amikor azt szeretnénk szimulálni, hogy a hangforrás valóban nagyon közel helyezkedik el a fejhez, legalább annyira, hogy a hanghullámokat ne lehessen síkhullámoknak tekinteni.

6.7 A különböző ITD számoló formulák implementálása a programba

A különböző módokon kiszámolt ITD értékeket ábrázoltam, és megvizsgáltam, hogy melyik formulákkal fogok további méréseket folytatni. Választásom a Woodworth és a Kuhn III. formulára esett. A Kuhn I. és a Kuhn II. formulák jellegre ugyan olyanok. Hátrányuk azonban, hogy csak bizonyos frekvenciatartományokra ajánlottak.

A Kuhn I. formula például csak 500Hz alatti frekvenciákra ajánlott. Ez azt jelenti, hogy valószínűleg egy főleg alacsony frekvenciás összetevőket tartalmazó hangforrás jelét lehetne, egy ilyen módon számolt ITD értékkel késleltetni, a hangforrástól távolabb elhelyezkedő fül számára. Ha a hangforrás tartalmaz 500Hz-nél magasabb frekvenciás összetevőket is, akkor azok késleltetési idejét nagyobb pontatlansággal fogjuk kiszámolni. Ez, zavart kelthet a hallgatóban, így akadályozva a lokalizáció pontosságát.

Ugyan ez a kérdés áll fenn a Kuhn II. formulára nézve is. A Kuhn II. formula olyan hangforrásokhoz ajánlott, amelyek főleg 2000Hz feletti frekvenciával rendelkező összetevőket tartalmaznak. A Kuhn II. formula minden horizontális síkbeli kitéréshez 2/3 akkora ITD értéket ad meg, mint a Kuhn I. formula. Ez a megállapítás a két formula képletéből adódik, ugyanis ha a két egyenletet elosztjuk egymással, majd átrendezzük:

$$\frac{ITD_{nagyf.}}{ITD_{kisf.}} = \frac{\frac{2a}{c} * \sin \varphi}{\frac{3a}{c} * \sin \varphi}$$

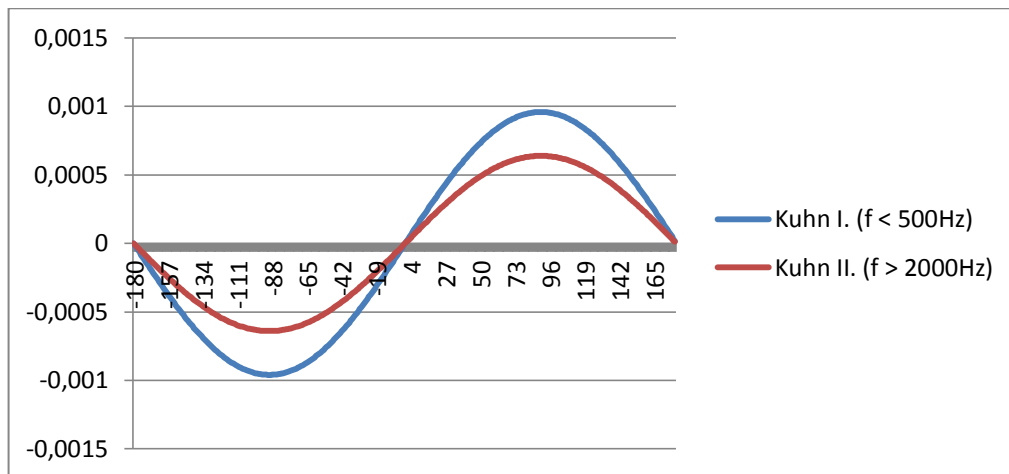
$$ITD_{nagyf.} = \frac{2}{3} * ITD_{kisf.}$$

26. ábra: A Kuhn formulák viszonya $ITD_{nagyf} > 2000$ Hz, $ITD_{kisf} < 500$ Hz

Ez az összefüggés is azt a következtetést igazolja, hogy a nagyobb frekvenciás komponensek gyorsabban terjednek a levegőben, mint a kisebb frekvenciás komponensek. A nagyobb terjedési sebesség miatt, ugyanakkora távolság megtételéhez kevesebb időre van szükség, így a késleltetés mértéke kisebb a nagyobb frekvenciákon.

Általánosságban a vizsgálatok nagy része, és az általam használt bemenő gerjesztőjel fehér zaj. Ennek oka, hogy a fehér zaj, a hallás teljes frekvenciatartományában egyenletes amplitúdó eloszlású. Ebből azonban következik, hogy fehér zaj alkalmazása esetén olyan eszközökkel, képletekkel ajánlott dolgozni, amelyek a teljes, használt frekvenciasávot képesek számolni. Érdekes lehet egy olyan formula kidolgozása, amely magában foglalja a Kuhn I. és Kuhn II. formulákat. Ez a formula Kuhn I.-ként működne az 500 Hz-nél alacsonyabb frekvenciákra, és Kuhn II.-ként a 2000Hz-nél magasabb frekvenciákra. Az ITD növekedésére az alacsonyabb frekvenciák felé, mindenképpen folytonos átmenetet kell biztosítani. Egy ilyen folytonos átmenet megvalósítása nehézkes, és valószínűleg, egy több lépcsős átmenet

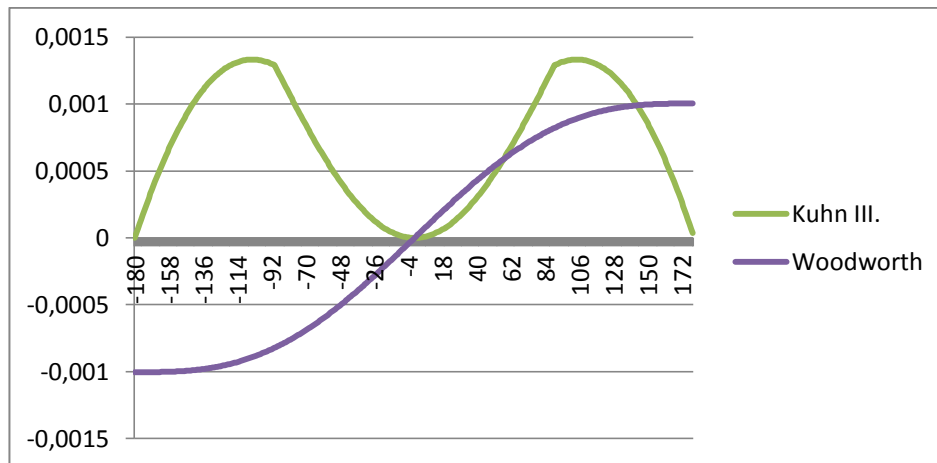
is megfelelő finomsággal közelíti a folytonos átmenetet. Egy ilyen összetett ITD számolási módszer megalkotása valószínűleg jóval egyszerűbb feladat, ha programozási feladatként kezeltem, mint ha matematikai problémaként.



27. ábra: A Kuhn I. és Kuhn II. formulák összehasonlítása

A Kuhn formulák egyesítése remek lehetőséget kínál majd a szimulációs program későbbi fejlesztésére. A felvetett ötletek, problémák és megoldásaik további vizsgálatok megtervezését és kivitelezését teszik szükségessé. Egyelőre azonban a Kuhn I. és Kuhn II. formulák megkötésekkel használhatók. Teljesen elvetni azonban nem akarom, az ALICE programba beépítésre kerülhetnek annak érdekében, hogy a későbbiekben minél több fajta kísérletet el lehessen végezni vele.

A Woodworth és a Kuhn III. formulákra nincsen frekvenciatartomány béli megkötés, azonban ezeknek is megvan a hátrányuk. Jellegre nem ugyan olyanok, ami azt jelenti, hogy a programba való implementálás során el kell különíteni a két számolási utat.



28. ábra: A Woodworth és a Kuhn III formula

A Kuhn III. formula a -180° - $+180^\circ$ -ig terjedő szögtartományból vár bemeneti értéket, és erre ad egy nulla fokos szögértékre szimmetrikus ITD értéket ábrázoló görbét. A szimulációs programban a hangforrás irányát 0° - $+360^\circ$ -ig lehet megadni. A számolás során erre a különbségre figyelni kell. A bemeneti szög értéktől függetlenül pozitív ITD értéket kapunk eredményül, ami azt eredményezi, hogy külön kell majd kiszámolni a késleltetés értékét a nulla foktól nagyobb és kisebb szögekre.

Ezzel szemben a Woodworth formula már negatív és pozitív késleltetési értékeket egyaránt visszaad, tehát ez már azt is jelzi, hogy a jobb fül jele késik-e a bal fül jeléhez képest, vagy siet. A Woodworth görbe 0° - $+180^\circ$ -ig szigorúan monoton növekszik, folyamatosan csökkenő meredekséggel. A görbe növekedése azonban csak 0° - $+90^\circ$ -ig elfogadható. Hiszen ha a hangforrást forgatjuk a fej körül az óramutató járásával megegyező irányban, akkor $+90^\circ$ -os kitérésnél lesz a jobb fül a hangforráshoz a legközelebb, a bal fül pedig a legtávolabb. Ebből adódik, hogy a két fül közötti jelterjedési időkülönbség is $\varphi = 90^\circ$ -os kitérésnél kell, hogy a legnagyobb legyen. Bármelyik ITD értéket ábrázoló görbének pedig maximum pontja kell, legyen, a $\varphi = 90^\circ$ -os kitérésnél. ezzel szemben a Woodworth görbe a 90° - $+180^\circ$ -ig terjedő tartományban is tovább növekszik, ami végeredményben azt eredményezné, hogy a frontális sík mögé szimulált hangforrást is csak jobb oldali irányból vélnénk hallani.

Ebből arra lehet következtetni, hogy a Woodworth formula csak $-90^\circ - +90^\circ$ -ig terjedően ad érvényes értékeket. A $90^\circ - +180^\circ$ és a $-90^\circ - -180^\circ$ -ig terjedő tartományokat pedig a $0^\circ - +90^\circ$ -ig és a $0^\circ - -90^\circ$ -ig terjedő tartományokból kell származtatni az alábbi módon:

$$Ha\ 90^\circ < \varphi < 180^\circ, \text{ akkor } ITD(\varphi) = ITD(180^\circ - \varphi),$$

$$ha\ -180^\circ < \varphi < -90^\circ, \text{ akkor } ITD(\varphi) = ITD(-180^\circ - \varphi).$$

Ugyan ebből a szempontból érdemes elemezni a Kuhn III. formulát is. Jól megfigyelhető a Kuhn III. görbe nem folytonossága a $\varphi = 90^\circ$ pontban. Továbbá a görbe maximumpontja sem a $\varphi = 90^\circ$ hanem a $\varphi = 105^\circ$ -os pontban található. A görbe e tulajdonságai miatt ugyancsak felmerülhet a kérdés, hogy a formulát a $-90^\circ < \varphi < 90^\circ$, vagy a $-180^\circ < \varphi < 180^\circ$ -ig terjedő szögtartományban érdemes-e használni. Azért az érdemes szót használtam, mert szubjektív megítélésem szerint, elméleti úton nem bizonyítható, hogy a Kuhn III. görbe említett tulajdonságai károsan befolyásolnák egy ilyen módon szimulált hangforrás lokalizálhatóságát. Ennek megítéléséhez a legbiztosabb módszer egy további felmérés elvégzése lehet.

7. Alice – A szimulációs program vizsgálata, fejlesztése

7.1 Az Alice futtatása

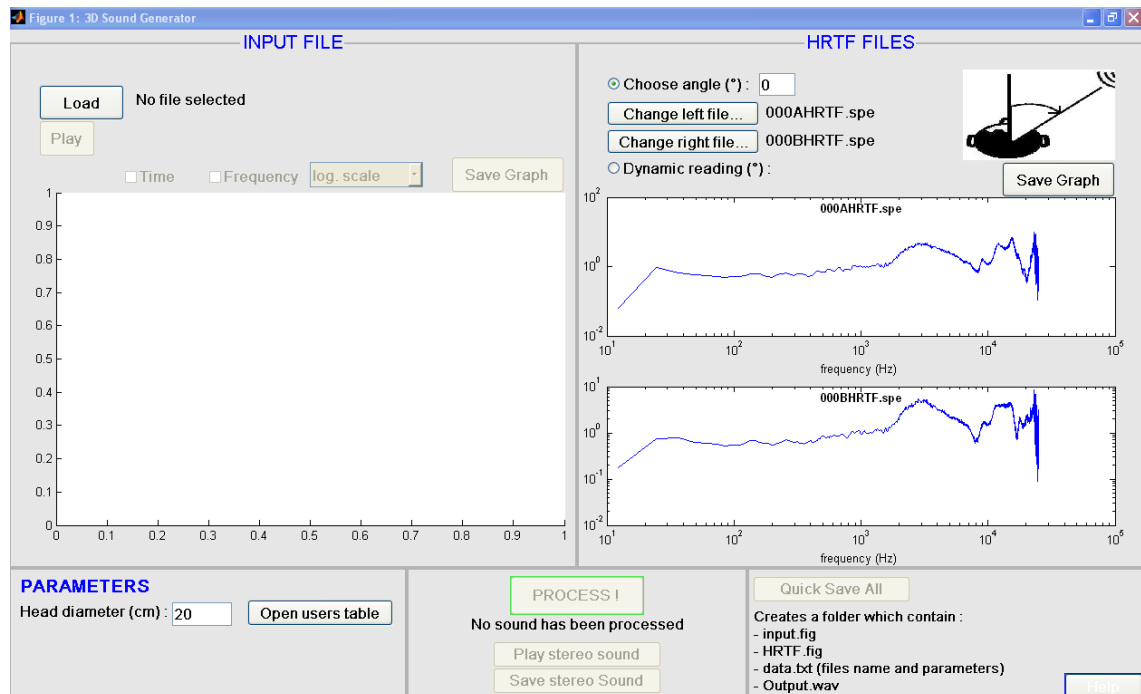
Az Alice nevű program Matlab-ban íródott, mely egy magas szintű programozási nyelv. Parancsszavai matematikai képletek és függvények értelemszerű alkalmazásából adódnak. Parancs vezérelt üzemmódban, vagyis interpreterként dolgozik. „Nyitott” programcsomag, mert újabb funkciók (parancsok) is beépíthetők úgynevezett .m-fájlok formájában. Toolboxszokkal, úgynevezett eszköztárakkal egészül ki, amelyek egy-egy speciális feladatosztály megoldására létrehozott .m-fájlok gyűjteményei. A MATLAB programozási nyelv is! Egy-egy fájl nevének (kiterjesztése nélkül) utasításként való megadása esetén a fájl sorait egymás utáni sorrendben hajtja végre. A % jel után megjegyzések írhatók, ezeket a megjegyzéseket, a program végrehajtása során, figyelmen kívül hagyja.

Minden alkalmazáshoz kijelölhető egy munkakönyvtár melyben a futáshoz szükséges fájlokat lehet elhelyezni. Ezt a project könyvtárat a MATLAB programon belül lehet kijelölni, és az alkalmazáshoz tartozó összes fájl-t célszerű ebben a munkakönyvtárban tárolni. A MATLAB, a Matrix Laboratory elnevezés rövidítése. A programok lényeges jellemzője, amelyre már a neve is utal, hogy a programban használatos változók mátrixok. Ez igen nagy különbség a többi jól megszokott programnyelvhez képest, és nagyon is más gondolkodásmódot igényel. Valamennyi változó globális, a változók típusára nincs külön definíció. A mátrix mérete és típusa a beírt értékek alapján automatikusan képződik. A MATLAB-ot általában, főleg a numerikusszámításokhoz ajánlják, más pl. analitikus számításokra inkább Maple-t ajánlanak [2].

Az Alice program grafikus felületét leíró szöveges fájl a `gui_main.m`. Az ALICE program futtatásának a kulcsa is ez a fájl. Ugyanis ha ezt a fájlt lefuttatjuk Matlabban, akkor megjelenik majd a program grafikus felülete. Ennek a módja pedig a következő:

1. Futtatni kell a Matlab programot.
2. A munkakönyvtárat az éppen aktuálisról át kell állítani arra a *work* könyvtárra, ami a *Final Project* könyvtárat és a benne lévő *.m* fájlokat tartalmazza. Ezt a Matlab fő ablakának a tetején tehetjük meg.
3. A képernyő bal oldalán lévő ablakban megjelenő könyvtárszerkezetben tallózva meg kell keresni a *gui_main.m* fájlt, és meg kell nyitni a Matlab beépített fájlszerkesztőjével.
4. Az 5. funkcióbillentyű (F5) megnyomásával a program futása megkezdődik, és megjelenik a program grafikus felülete.

A program feladata viszonylag egyszerű. Egy adott hangfájlt ellát irányinformációkkal, úgy hogy a hallgató számára az egy jól meghatározható irányból hallatszódjon. Ezt a következő beállításokkal lehet megtenni:



29. ábra: Az eredeti Alice program grafikus felülete

5. A Load gombra kattintva egy felugró ablakban ki kell választani a bemenetként használandó hangfájlt. (Amikor ez megtörtént, akkor a Load gomb alatti

ablakban meg lehet jeleníteni a bemeneti hangfájl spektrumát és időfüggvényét.)

6. A jobb oldali felületen azt lehet kiválasztani, hogy melyik irányhoz tartozó irányinformációkkal szeretnénk ellátni a hangfájlunkat. A dinamikus opciót jelenleg nem használjuk. A vizsgált irány kiválasztása után, az adott irányhoz tartozó HRTF fájl spektrumképe automatikusan ábrázolásra kerül.
7. A következő lépésben a zajlik a hangfájl tényleges feldolgozása. Ezt a Process gombbal kell elindítani. A program számol egy kis ideig, elvégzi a hangfájlon a megfelelő változtatásokat, majd engedélyezi a Play gomb használatát.
8. A Play gomb megnyomására lejátsza az irányinformációkkal ellátott hangot, amit természetesen fülhallgatón érdemes meghallgatni.

A feladatom végrehajtása előtt, program működéséről pontosan ennyit tudtam. A feladatom az volt, hogy ellenőrizsem, hogy a kimeneti hanganyag, az elméletből ismert, szükséges változtatásokon ment-e keresztül. Ez több részből tevődik össze.

- Meg kell vizsgálni, hogy a 6. lépés hatására ténylegesen kiválasztódik-e a megfelelő HRTF fájl. Vizsgálandó, hogy a HRTF fájl tartalma az elvárásoknak megfelel-e.
- Vizsgálandó, hogy a bemeneti hangfájlon a HRTF szűrés végrehajtódik-e.
- Vizsgálni kell, hogy a jobb és bal fülek között megtörténik-e a késleltetés.
- További feladat, hogy az ITD értékek számolása, választhatóan több féle képlettel is, számolható legyen.

7.2 A bemeneti hangfájl kiválasztása

A program grafikus felületének megnyitása után lehetőség nyílik arra, hogy megadjuk a bemeneti paramétereket. Ezen bemeneti paraméterek mind elérhetőek a grafikus felületen keresztül. A sorrendben az első paraméter, amit beállíthatunk az a hangforrás hangja. Ezt egy hangfájl formájában van lehetőség megadni. Az INPUT FILE nevezetű panelen található egy *Load* elnevezésű gomb, aminek a megnyomásával felugrik egy ablak, amiben lehetőség van arra, hogy a számítógépünk mappái között tallózzunk. Ez az ablak úgy van beállítva, hogy csak a wav kiterjesztésű fájlokat jeleníti meg. Ez azért fontos, mert az Alice csak wav fájlokat képes feldolgozni. Ha a forrás valamilyen más formátumban van meg, akkor először egy külső program segítségével át kell konvertálnunk azt, wav kiterjesztésű hangfájlá. Ha ez megvan, akkor már beimportálható a programba.

A bemeneti hangfájl kiválasztásánál egy nagyon fontos dologra kellett figyelni. A bemeneti hangfájl csak mono fájl lehetett. Ez alatt azt értem, hogy csak egy darab hangcsatornával rendelkezhetett a fájl. Ez a kritérium nem a Matlab sajátosságaiból ered. Ezt az állítást le is ellenőriztem. A Matlabban található egy `wavread()` nevezetű függvény, ami arra alkalmas, hogy wav kiterjesztésű fájlokat olvasson be egy változóba. Természetesen ez a változó egy vektor, azonban sztereó hangfájl esetében már mátrix. Ez a mátrix $n \times 2$ nagyságú, mono hangfájl esetében pedig $n \times 1$ nagyságú. A sztereó, tehát a két darab hangcsatornával rendelkező, fájlok használata az Alice miatt volt lehetetlen. Ugyanis a bemenettől egészen a HRTF-ek feldolgozásáig egy csatorna volt kezelve a programban. Ezt mindenképpen meg kellett változtatni, ennek okát a következő bekezdésben magyarázom el. Tehát a program mono részét átírtam úgy, hogy már az elejétől fogva, két darab hangcsatornát kezeljen. További észrevétel, hogy a `wavread()` függvény képes mono és sztereó fájlt is beolvasni, ezért, van lehetőség olyan módon továbbfejleszteni a programot, hogy az, a `wavread()` függvénytől mono és sztereó, azaz $n \times 1$ és $n \times 2$ –es mátrixot is át tudjon venni. Ezt a fejlesztést el is végeztem, a `MakeStereo.m` fájlban írtam meg a programkódot. A `wavread()` a `MakeStereo()` függvénynek adja át a beolvasott mátrixot. A `MakeStereo()` a

futása elején megvizsgálja, hogy a kapott változó egy, vagy kettő hangcsatornát tartalmaz-e. Ha ez a szám kettő, akkor nem változtatja meg a fájlt, ha pedig csak egy csatornát talál akkor a ezt a csatornát duplikálja. Ilyen módon, lesz egy kétcsatornás változónk, ami mindkét csatornájában ugyan azokat az értékeket tartalmazza.

Miért is volt szükség erre a fejlesztésre? Logikáját tekintve, helyesnek látszik, hogy csak mono hangfájlokat volt képes kezelni a program. Hiszen a valóságban ritka jelenség, hogy a hangforrásnak több „hangszórója” legyen, így valószínűsíthetjük azt is, hogy a hangforrás, amit szimulálni szeretnénk egycsatornás. Azonban előfordulhat, hogy valamilyen oknál fogva több csatornás hangfájlt szeretnénk beolvasni a programba. Esetemben ez az ok a fejhallgató-kiegyenlítés volt, amit a következő fejezetben bővebben is kifejtek.

Továbbá lehetőségünk van a kijelölt fájlt lejátszani és meghallgatni a *Play* gomb segítségével. A kijelölt hangfájl időfüggvénye és- vagy frekvenciaspektruma megjeleníthető az INPUT FILE panel grafikonján, amely grafikon el is menthető egy külső fájlba. A függvények ábrázolása választhatóan történhet lineáris és logaritmikus skála szerint.

Az INPUT FILE panelen található egy további checkbox, *Input is not periodic* elnevezéssel, amelyről a [7.7 fejezet](#)ben tesztek említést.

7.3 A fejhallgató kiegyenlítés

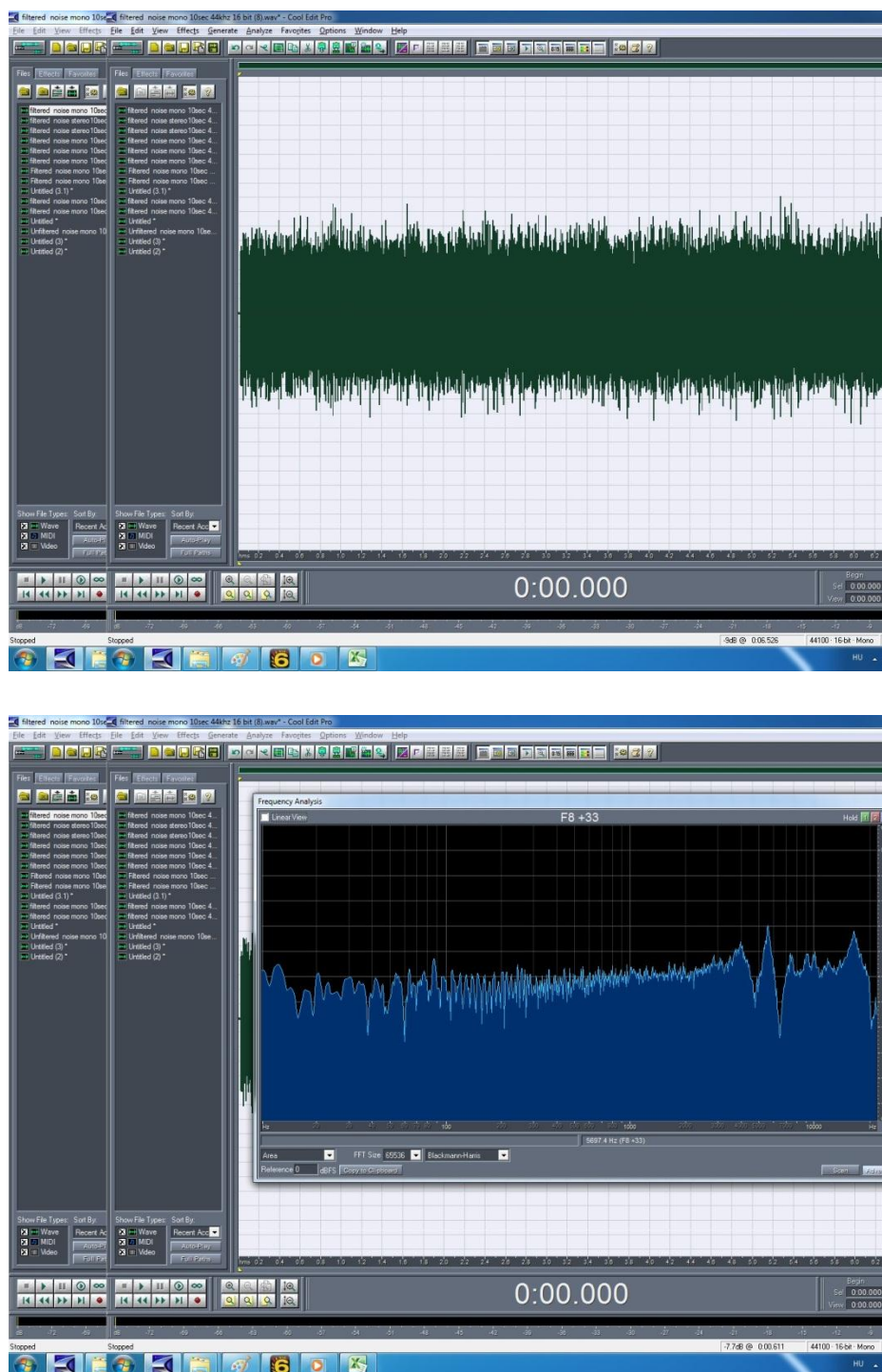
A fejhallgató-kiegyenlítésre azért van szükség, mert a fejhallgató akaratlanul is elvégez egy frekvenciafüggő csillapítást az általa kiadott hangokon. Ez a nem kívánt szűrési jelenség hozzáadódik az általunk, szándékosan végzett szűrési eredményekhez, és így meghamisítja azokat. Ezt elkerülendő, adott egy lehetőség arra, hogy a fejhallgató átviteli függvényének inverz függvényével előszűrjük a bemeneti hangfájlunkat. Ez az inverz függvény viszont eltérő karakterisztikájú a bal illetve a jobb csatornára nézve. Ebből következik, hogy ha ilyen módon előszűrünk egy hangfájlt, akkor az, az előszűrés pillanatától egy sztereó hangfájl lesz.

A program-fejlesztés elején az volt a cél, hogy az előszűrés műveletét legyen lehetőség az Alice programon belül megvalósítani. Ezen okból írtam egy program-modult, az `aplHeadOff.m` fájlba, és a grafikus felületet is kiegészítettem egy gombbal, amivel ezt az Alice fejhallgató-kiegyenlítő funkcióját tudjuk aktívra tenni. A *Use Headphone Offset* elnevezésű gomb megnyomásának hatására a `UseHeadphoneOffset()` függvény megkérdezi annak a kettő darab `.SPE` kiterjesztésű szűrőfüggvénynek a helyét amelyikkel a fejhallgató kiegyenlítést el kell végezni, majd ezeket az információkat külső változóban eltárolja. A szimuláció futtatásakor ezen információk felhasználásával történik a szűrés. A szűrés menete megegyezik a HRTF szűrések menetével, amiről bővebben a [7.8 fejezet](#)ben lesz szó.

A Alice fejhallgató-kiegyenlítő modulja működőképes állapotú, azonban a további fejlesztését egyelőre elvetettük, ugyanis jobb megoldásnak tartjuk, ha a fejhallgató kiegyenlítés nem az Alice programon belül történik. A kiegyenlítés megtörténhet az Alice-tól teljesen függetlenül is. Ekkor, az Alice használatkor már egy ilyen, előszűrt hangfájl kell beolvasni a szimuláció kezdetekor. Ezt megtehetjük, hiszen a frekvenciafüggő szűrések a gyakorlati megvalósításban, frekvenciafüggő szorzást jelentenek, és mint tudjuk, a szorzás művelete kommutatív. A módszer további előnye, hogy nagy mennyiségű wav fájl készítés esetén, a program futása kevesebb időt igényel, ugyanis a számítások mennyisége nagymértékben csökken. Nagy mennyiségű wav fájl készítéséről a [7.10 fejezet](#)ben lesz szó.

Tehát a szimuláció megkezdéséhez szükség lesz egy fejhallgató-kiegyenlített fehér zajra, amit a továbbiakban bemeneti hangfájl, bemeneti wav néven említek. Rendelkezésemre állt kettő ilyen hangfájl, amit Dr. Wersényi György készített. Ezeknek a fájloknak a kiegyenlítése egy Sennheiser fejhallgatóhoz készült, az egyik fájl szűrése FIR2, a másik fájl szűrése pedig INVFRQ eljárással. További célunk volt, hogy megállapítsuk, hogy melyik hangfájl érdemes használni a továbbiakban. Ezért, a BME Akusztikai és Stúdiótechnikai Laboratóriumában, Répás Józseffel végzett vizsgálataink alapján a választásunk a *SH HeadSimulator stereo 44_1 16 bit INVFRQ.wav* fájlra esett, ugyanis ennek a fájlnek az átvitele volt egyenletesebb.

A tesztek alatt azt vizsgáltuk, hogy az előszűrt hangfájl a *Sennheiser* fülhallgatón visszajátszva, mennyire lesz egyenletes az átvitel. A méréseket nagy számban elvégeztük és átlagolással növeltük a pontosságot.



30. ábra: A SH fejhallgatóval lejátszott, fejhallgató-kiegyenlített wav fájl, tipikus idő- és átviteli függvénye, a mérések során

7.4 A hangforrás irányának kiválasztása

A HRTF FILES panelen, lehetett kiválasztani azt a szöveget, ami megadja a szimulálni kívánt hangforrás irányát. Ehhez nyújt segítséget, a szintén ezen a panelen elhelyezkedő ábra, ami jelzi, hogy a szög melyik irányú kitérést jelöli a mediális síktól. Ez a kitérés $0^\circ \leq \text{angle} \leq 359^\circ$ között lehet. A kitérés szögét az `angle` változóval jelölöm a programban, ezért itt is ezt a jelölést fogom használni. Az `angle` megadása után lefut az `editangle()` nevű függvény. Ennek a függvénynek a feladata, hogy a megadott `angle` szögből előállítsa a HRTF fájlok nevét, és elmentse egy külső változóba. A kiválasztott HRTF-függvényeket ezután megjeleníti a HRTF FILES panelen található grafikonokon, logaritmikus skálázással.

Az eredeti Alice program néhány nagyon fontos fejlesztésen esett át ezen a területen. Az egyik legfontosabb észrevétel az volt, hogy az eredeti `editangle()` függvény rosszul választja ki a HRTF fájlokat, felcseréli azokat. Ez azt eredményezi, hogy a `HRTFNameLeft` és a `HRTFNameRight` nevű külső változóba is a helytelen fájlnevek kerültek. A későbbiekben tárgyalt HRTF szűrés is ezen külső változókból nyeri ki a használni kívánt HRTF-függvények neveit, amivel a szűrést kell végezni. Így a jobb fülhöz tartozó HRTF-függvényekkel lettek megszűrve a bal csatornához tartozó amplitúdó értékek, és fordítva. elrontotta a lokalizációt.

További fejlesztés az függvényen az, hogy hogyan választja ki a HRTF-függvények elérési útját. Az előzőekben a HRTF-ek kiválasztása mindig egy adott helyről, adott könyvtárból történt. Ez a könyvtár az alapértelmezett munkakönyvtárból nyílt, azaz relatív útvonala volt. Ebben az alapértelmezett könyvtárban keresett a program egy `\HRTFs\` nevű könyvtárat, amiben a felhasználni kívánt HRTF-eknek lenniük kellett. Ez a megoldás nem tesz lehetővé a különböző HRTF fájlok rugalmas kezelését. Abban az esetben ha másik HRTF készletet szeretnénk volna használni, akkor ezt egy külső fájlkezelő program segítségével lehetett megtenni, úgy hogy a `\HRTFs\` könyvtárban található HRTF-eket felül kellett írni, az éppen aktuálisan szükséges HRTF fájlokkal. Ez meglehetősen nehézkes felhasználást tesz lehetővé.

Ezt az eljárást egy megbízhatóbb módszerrel cseréltem le. A program indulásánál még nincs lényeges különbség. Alapértelmezett mappaként most is a fent említett mappában keressük a HRTF-eket. Ez a mappa a Matlab beállított munkakönyvtárában található, `\HRTFs\HRTFs_Natural\HRTFs_Natural_Horizontal\` mappát jelenti, amelynek a nevét fel is tünteti a program az INPUT PANEL tetején. Ennek a mappának az aktualitását, érvényességét jelzi az a checkbox (szabadfordításban: pipáló gomb), ami mellett a *Use default HRTF directory* elnevezés olvasható. A checkbox nevének a jelentése ugyan az, ami a funkciója is egyben: az alapértelmezett HFRF mappa használata. Ez a mappa tartalmazza a natúr, tehát a haj, szemüveg és egyéb kiegészítők nélkül rögzített, horizontális síkhoz tartozó HRTF-függvényeket. Lehetőség van ezt a mappát megváltoztatni, melynek első lépése, hogy kivesszük a pipát az említett checkbox-ból. Ez a művelet engedélyezi a *Change* elnevezésű pushbutton-t (szabadfordításban: nyomógomb). A *Change*-re rákattintva megjelenik egy ablak, amiben szabadon tallózhatunk a számítógép mappái között, és kiválaszthatjuk azt a mappát, ami tartalmazza a felhasználni kívánt HRTF-függvényeket. További felhasználást segítő funkció, hogy a program megjegyzi az utolsónak kiválasztott mappát, és legközelebb már innen kezdhethetjük a tallózást. Az aktuálisan használt mappa minden esetben kijelzésre kerül, közvetlenül a pushbutton alatti részben, így mindig ellenőrizni tudjuk annak a helyességét.

Az Alice program egyik tervezett felhasználási területe az, hogy a különböző módon előállított HRTF-eket is össze tudjunk hasonlítani egymással. Itt nagyon széles területen és szerteágazó vizsgálatokat lehet lefolytatni. Például, a HRFT rögzítésének sok módja lehetséges, amely módszereket érdemes lehet összehasonlítani. Az egyik ilyen összehasonlítási mód lehet az is, hogy megvizsgáljuk, hogy nagyszámú kísérleti személyen tesztelve melyik eljárásokkal rögzített HRTF-ekkel sikerül jobb lokalizációt elérni az alanyokon. A különböző HRTF-ek különbözősége, azonban nem csak az őket rögzítő eljárások különbözőségén alapulhat. Az előző fejezetekben említettem, hogy HRTF-függvényeket szándékosan is szokás „elrontani”, meghamisítani. Ilyen eset például, amikor egy parókát, vagy szemüveget helyezünk a HRTF rögzítési eljáráshoz használt a műfejre, torzóra. Ezeknek a tárgyaknak a jelenléte befolyásolja a hangteret,

ezáltal megváltoztatja a HRTF-függvények karakterisztikáját. Ezekben az esetekben azt vizsgáljuk, hogy a haj, szemüveg, sapka jelenléte milyen változásokat okoz a HRTF-függvényekben a natúr műfejes HRTF-ekhez képest. Ezek a HRTF béli változások hatással kell, hogy legyenek a lokalizáció sikerességére. A vizsgálatok célja, hogy kiderüljön, hogy ezek a hatások pozitív vagy negatív irányban befolyásolják a lokalizációt, esetleg, mindössze elhanyagolható mértékűek. [18] Elgondolkodtató lehet egy könnyen elvégezhető hétköznapi megfigyelés. Például, mielőtt bemegyünk egy fodrászhoz, és miután kijövünk, nem érzékelünk feltűnő változást a lokalizációnk minőségében, pedig az individuális HRTF készletünk időközben megváltozott. Vajon mi lehet egy szemüveget viselő ember tapasztalata, amikor elfelejti felvenni a szemüvegét? Ezen gondolatmenetek okot adhatnak a feltételezésre, hogy a sapka, szemüveg jellegű hangtér torzító tárgyak zavaró hatása, vagy nem számottevő, vagy a hallórendszerünk képes észrevétlenül is felülemelkedni ezeken a hatásokon.

Ebben a részben szeretném megemlíteni az Alice egy további, általam fejlesztett „tool”-ját, funkcióját és egy további felhasználási lehetőségét. A `HRTF FILES` panelen található, *Save I. R.* elnevezésű pushbutton. Ez az elnevezés az angol *Save Impuls Response* rövidítése. A gomb megnyomásával meghívjuk a `SaveITDifftHRTF()` függvényt, amely első lépésként beolvassa a kiválasztott HRTF-függvények amplitúdóértékeit. A kapott vektorok lesznek a bemenetei az `ifft()` függvénynek, ami egy inverz Fourier transzformáció. Az `ifft()` kimenete pedig szintén két darab vektor lesz, amelyek most már idő-tartomány béli információt hordoznak. A `SaveITDifftHRTF()` függvény megkérdezi, hogy hová szeretnénk menteni az kimenetét. Utolsó lépésként, a `wavwrite()` függvény segítségével elmentjük a kapott hangfájlunkat egy tetszőleges helyre. A wav, amit ilyen módon kapunk, az átviteli rendszerként értelmezett HRTF-függvények impulzus válasza.

A másik említett felhasználási lehetőség az, hogy a „HRTF = 1” elnevezésű checkbox-ra kattintással az alkalmazandó HRTF szűrő értékét konstans „1” értékre állítjuk. Ekkor a HRTF szűrés gyakorlatilag nem fut le, hanem a hangfájl változtatás nélkül átkerül a programkód következő részébe, ahol a további feldolgozás már

megtörténik. Ez az opció, például arra ad lehetőséget, hogy tesztelhessük a szimuláció sikerességét abban az esetben is, ha a bemeneti hangfájlunkat csak ITD információkkal látjuk el.

7.5 A szimuláció személyre szabása

Ebben a fejezetben a *Parameters* panelről lesz szó. A jelenlegi vizsgálatainknál, ez az a funkció, amellyel a szimuláció, a hallgató személy egyedi tulajdonságára szabható. Ez a tulajdonság pedig, a fejátmérő. Valójában nem ez lenne az egyedüli lehetőség, hiszen a program képes arra, hogy különböző HRTF készletekkel dolgozzon, amelyek lehetnének akár individuális HRTF-ek is. A gyakorlatban azonban, ezeknek meglehetősen körülményes az előállításuk, és a felhasználási lehetőségük is sokkal korlátozottabb, mint a műfejes technikával rögzített HRTF készleteké (lásd [5.2 fejezet](#)). Jelen esetben a felhasznált HRTF-függvények minden vizsgálati alanynál megegyeznek. Említettem, hogy a HRTF készletnek csak az amplitúdó értékei kerülnek felhasználásra, a fázis információit elhagyjuk, és más módszerrel helyettesítjük. Ezt megtehetjük, hiszen a rendszerünk minimálfázisú ([5. fejezet](#)). A fázistolás értékeit, az ITD információval helyettesítjük, ami azt adja meg, hogy a síkhullámok milyen időkéselettel érik el a hangforrástól távolabb eső fület, a hangforráshoz közelebb eső fülhöz képest. Az ITD információ előállításához, különböző tapasztalati képletek, formulák állnak rendelkezésre, melyek közül a Kuhn és a Woodworth formulákat implementáltam az Alice programba. Az eredeti Alice-ban használt ITD számoló formula helytelennek bizonyult, így az, már nem elérhető a szimulációban. A program futtatása során lehetőségünk van, hogy kiválasszuk, hogy melyik formulával szeretnénk az ITD információt előállítani. Alapértelmezettként, a Woodworth formulával fog számolni a program, amit ki is jelez úgy, hogy kettő darab radiobutton (választógomb) közül a *Woodworth* elnevezésű lesz aktív.

Ezek a formulák négy bemenő paraméterrel képesek számolni, amelyek a következők:

- horizontális sík béli kitérés, azimuth, δ
- mediális sík béli kitérés, eleváció, ϕ
- hangterjedési sebesség, c
- fej átmérő, head diameter, d

A szimulációs program jelenlegi felhasználásában a horizontális és mediális sík béli elmozdulásokat mindig egymástól elkülönítve használjuk, szimuláljuk. Nem áll elő az eset, amikor az azimuth és elevációs szög értéke is különbözik nullától. Tehát, a horizontális sík béli vizsgálatoknál az eleváció értéke nulla, mediális sík béli vizsgálatoknál pedig az ITD értéke nulla. Ebből adódóan az elevációs szög értékét nem használjuk fel az ITD előállításához, csak a HRTF-függvények kiválasztásánál. A másik három paraméter mindegyike könnyedén beállítható az Alice grafikus felületén. A hangsebesség beállítás és a fejátmérő beállítás közvetlenül is elérhető. Az azimuth szög értékét, egy előzőleg beállított paraméter értékéből olvassa ki a program. Ez a paraméter a HRTF fájlok kiválasztásánál került megadásra. Tehát a HRTF fájlok kiválasztásához, és az ITD érték számolásához, ugyan azt a paramétert használjuk fel. Ezt a paramétert használom fel a Mediális sík béli kitérésekhez tartozó HRTF fájlok kiválasztásához is.

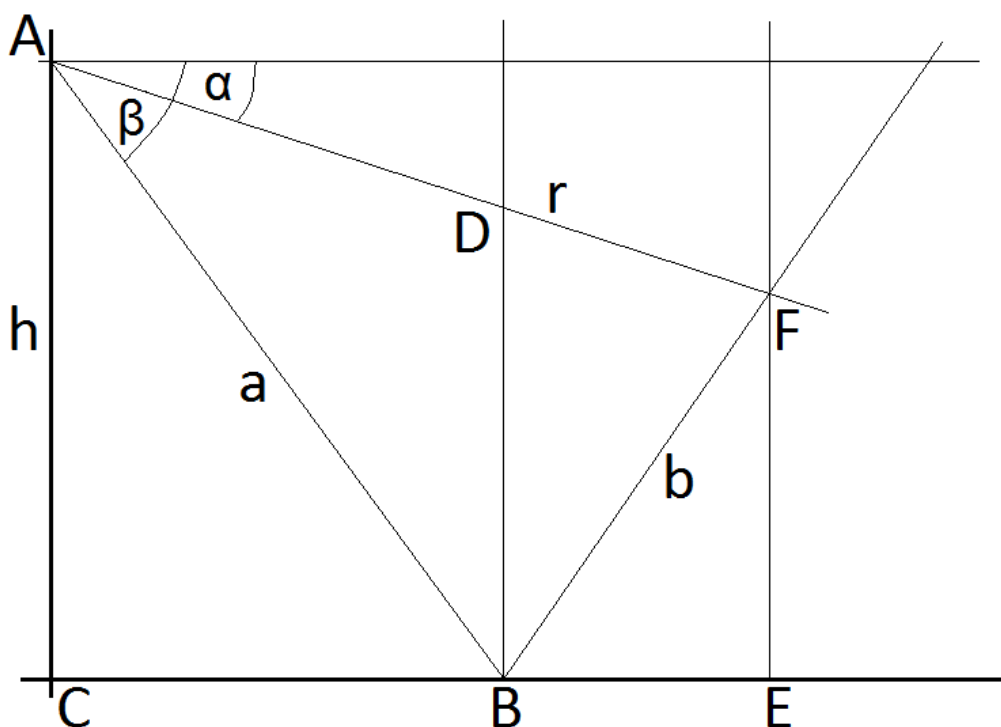
Mindenképp megemlítendő, hogy az ITD számolása és alkalmazása csak akkor történik meg, ha a program horizontális sík béli kitérést szimulál. Ha a programban aktiválva van az *Elevation* elnevezésű checkbox, akkor az IDT számoláshoz tartozó programrész nem fut le, és a hangfájl értékeit tartalmazó mátrix érintetlenül adódik át a következő program-modulnak.

7.6 A reflexiók szimulálása

Az Alice képes arra, hogy szimuláljon olyan eseteket amikor, a hangforrás hangja reflektálódik a hangforrás és a hallgató személy között is húzódó földfelületről, majd a reflektált hangjel ugyancsak eljut a hallgató személyhez. Ekkor azt szeretnénk vizsgálni, hogy a földfelületről érkező reflexió milyen hatással van a lokalizációra. Ezen

irányú vizsgálatoknak, a legfőbb határoló tényezője az a tény, hogy a HRTF-függvényeink mindegyike vagy a horizontális, vagy a mediális síkba tartozik. Ebből következik, hogy a horizontális sík béli kitérés esetén keletkezett, reflektált jelhez, nem tudunk hozzárendelni egy HRTF-függvényt sem. Így az azimuth szöget szükségszerűen nullának kell választanunk, kitérés, csakis a mediális síkban lehet. Éppen ezen megfontolásból, a reflexiók szimulációt úgy építettem bele az Alice-ba, hogy az, opcionálisan, kiegészítse a Mediális sík béli vizsgálatokat. Tehát, a *Reflection* elnevezésű checkbox, akkor válik engedélyezett állapotúvá, ha az *Elevation* checkbox is aktiválva van. Ez önmagában még nem jelenti azt, hogy a program reflexiót is fog számolni, mert ameddig a *Reflection* panelen található objektumok szürke színnel jelennek meg, addig azok inaktív állapotúak. A reflexió szimulálását, alkalmazását, a *Reflection* checkbox aktiválásával lehet elérni. Ekkor a *Reflection* panelen található objektumok színe is szürkéről feketére változik, jelezve az aktív állapotot.

A reflexiót szimuláló programmodul megírása előtt készítettem egy ábrát, amelyen jól megfigyelhető, hogy a jel milyen utakat jár be a hangforrástól a hallgató személyig.



31. ábra: A reflexió: $a+b$, és közvetlen: r jelutak, a mediális síkban

Mielőtt tovább haladnék, íme, egy rövid jelölésmagyarázat a 17. ábrához:

- A pont: a hallgató személy fejének a középpontja
- F pont: a hangforrás helye a térben
- AF szakasz, r: a hangforrást és a hallgató személyt összekötő egyenes
- CE szakasz: a föld síkja, ahonnan a hang, képes a reflektálódásra
- B pont: Az a pont, ahol az F-től az A-ig terjedő reflexiós jelút, érinti a földet
- α szög: a közvetlen jelúthoz tartozó elevációs szög
- β szög: a reflektált jelúthoz tartozó elevációs szög.

A fenti ábráról nem megállapítható, de szimuláció alatt az α és a β szögek nem lehetnek tetszőleges nagyságúak. Ennek leginkább, gyakorlati oka van. A HRTF készlet ugyanis véges, nincs minden szögértékhez HRTF-függvényünk. A rendelkezésre álló HRTF-ek, -45° -tól $+90^\circ$ -ig terjednek 5° -os felbontásban. A HRTF készlethez alkalmazkodva, a reflexió számolásánál is a rendelkezésre álló HRTF-ekhez tartozó szögértékek közül lehet csak választani. Így az ábrán látható α és a β szögek csak adott értékeket vehetnek fel, amelyeket már a szimuláció kezdetén kiválasztunk, ismerünk. Ez az oka annak is, hogy az ábrán csak ezek a szögek kaptak megkülönböztető jelölést. Továbbá kiemelendő még a h szakasz, ugyanis ezt a hosszúságot is bemeneti paraméterként kezelem, ami a kísérleti alanyunk magasságát jelenti, ami a gyakorlatban szintén adott. Használok továbbá egy negyedik paramétert is, ami az a reflexiós tényező, amelyjel a reflektált jel csillapodik a B pontban.

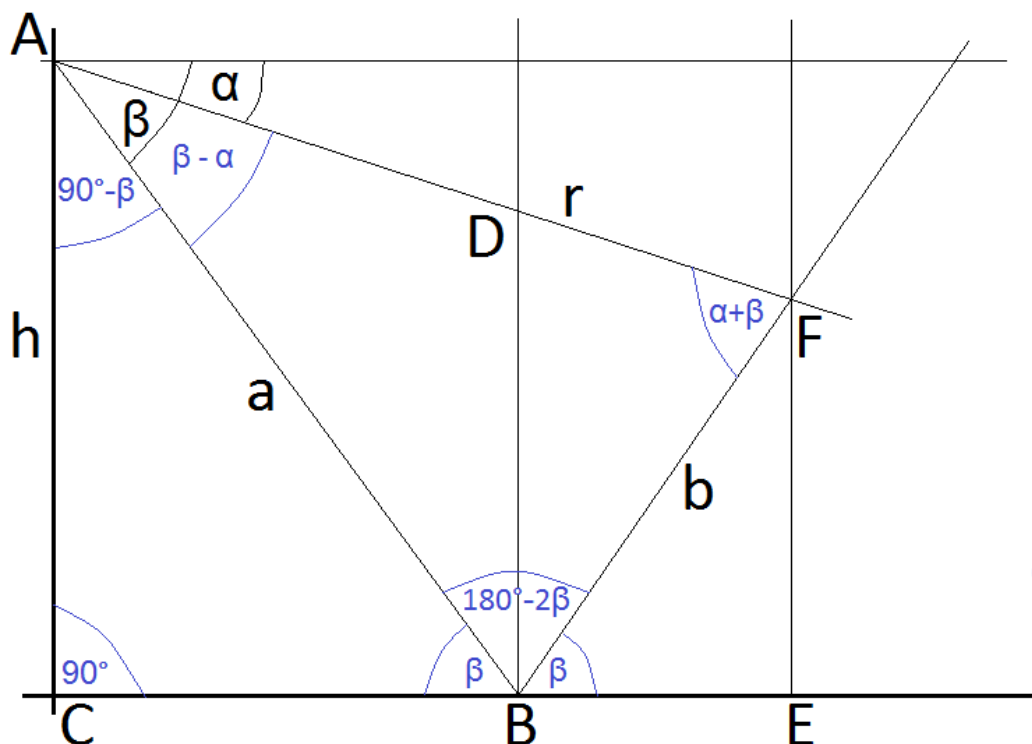
A három bemeneti paraméterből, α , β , h kel kiszámolni a további szükséges adatokat, amelyek a reflexió számolásához szükségesek lesznek. Ahhoz hogy a feladatot elvégezhessem, tudnom kell, hogy milyen jelenség fog történni az A pontban. Az A pontba két külön irányból fog beérkezni egy-egy hangjel, azaz vektor. Az egyik vektort az α szöghöz tartozó, a másik vektort a β szöghöz tartozó HRTF-fel kell megszűrni. továbbá, a két vektor különböző úton érkezik, ezért a megtett út és a beérkezés időpontja is különbözni fog. Az $F \rightarrow B \rightarrow A$ úton érkező vektor késni fog az $F \rightarrow A$ úton érkező vektorhoz képest. A programban ezt a késleltetést úgy fogom

kivitelezni, hogy a reflektált vektor elejére, és a közvetlen vektor végére, egy t hosszúságú, azaz n elemű vektort fűzök. Az n értékének meghatározásához először is a terjedési időt, ahhoz pedig a reflexió jelút, azaz, $a+b$ szakasz hosszát kell meghatározni.

Első lépésként felírom az ismert szögeket, majd kiszámolom a további számolásokhoz szükséges szögeket. A három betű egymás után, szögeket fog jelölni, ahol a két szélső betű a szög szárait jelenti. Sajnos, a Word nem ismeri a $\sphericalangle$ jelölést, mint karaktert. Tehát, az ABC szög szárai AB szakasz és BC szakasz.

- $BAC = 90^\circ - \beta$, ugyanis pótszögek
- $ACB = 90^\circ$, ugyanis derékszög
- $ABC = 180^\circ - ACB - BAC = 90^\circ - \beta$, ugyanis a háromszög szögeinek összege 180°
- $FBE = ABC = 90^\circ - \beta$, a visszaverődés geometriai szabályosságai miatt
- $ABF = 180^\circ - ABC - EBF = 180^\circ - 2\beta$, kiegészítő szögek
- $BAF = \beta - \alpha$, itt kell tennünk egy kikötést, miszerint: $\beta > \alpha$
- $AFB = 180^\circ - BAF - ABF = 180^\circ - (\beta - \alpha) - (180^\circ - 2\beta) = \alpha + \beta$

A fenti eredményekkel kiegészítem a 17. ábrát, a könnyebb átláthatóság érdekében:



32. ábra: a kiszámolt szögértékek megjelenítése

A szögek ismeretében, alkalmaztam a koszinusz szögfüggvényt az ABC derékszögű háromszögre, az a oldal hosszának kiszámításához:

$$\cos(90^\circ - \beta) = \frac{h}{a}$$

Ezt átrendezve:

$$a = \frac{h}{\cos(90^\circ - \beta)}$$

A b szakasz kiszámításához a szinusz-tételt alkalmaztam az ABF általános háromszögre.

A szinusz-tétel kimondja, hogy: Egy általános háromszögben, két tetszőleges szög szinuszainak aránya megegyezik a szögekkel szemközti oldalak arányával. Tehát:

$$\frac{\sin(\alpha + \beta)}{\sin(\beta - \alpha)} = \frac{a}{b}$$

Ezt átrendezve:

$$b = a * \frac{\sin(\beta - \alpha)}{\sin(\alpha + \beta)}$$

Az r szakasz kiszámításához a szinusz-tételt alkalmaztam az ABF általános háromszögre:

$$\frac{\sin(\alpha + \beta)}{\sin(180^\circ - 2\beta)} = \frac{a}{b}$$

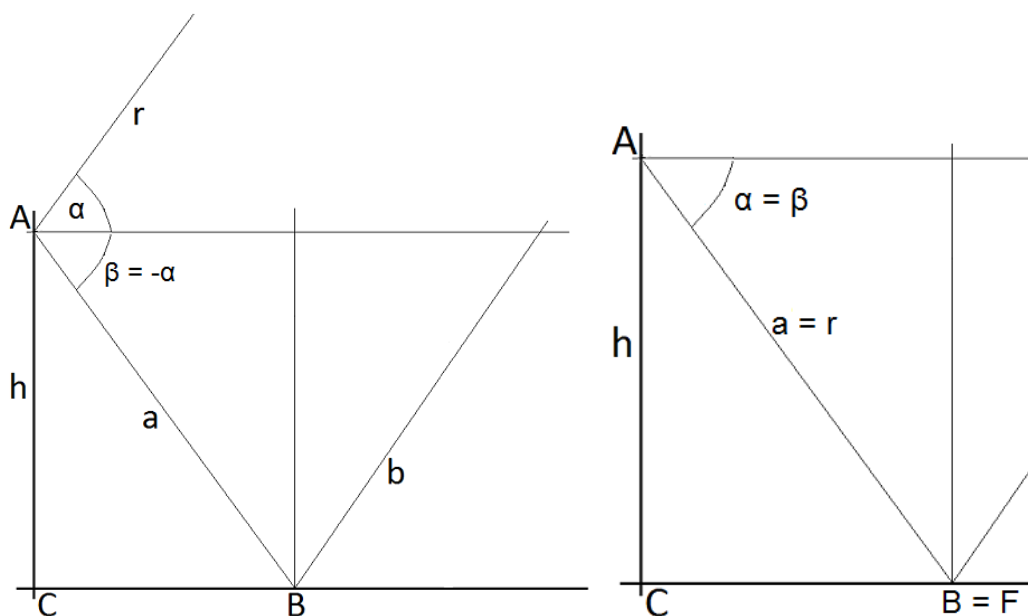
Ezt átrendezve:

$$b = a * \frac{\sin(180^\circ - 2\beta)}{\sin(\alpha + \beta)}$$

A késleltetési időt megkapjuk, ha a direkt és a reflexiós útvonalak hosszának különbségét, elosztjuk a hang terjedési sebességével:

$$t_{\text{késleltetés}} = t_{\text{ref.}} - t_{\text{dir.}} = \frac{a + b - r}{c}$$

A fenti képletek és összefüggések mindegyikét implementáltam az Alice programba, a `calculaterefdata()` függvénybe. A programban használt összefüggések annyiban térnek el a fentebb bemutatott összefüggésektől, hogy az α szög helyére $-\alpha$ szögértéket helyettesítettem. Ennek a műveletnek, programozás béli előnyei voltak. Amikor, az Alice grafikus felületén megadjuk a reflexiószámításhoz megadjuk a szükséges paramétereket, a program azonnal lefuttatja a `calculaterefdata()` függvényt, amely a számítások végén kijelzi a kiszámolt információkat, amelyek tartalmazzák például, a hangforrás távolságát is. Ezek az információk segítenek eldönteni, hogy az adott paraméterekkel szimulált hangforrás, vajon milyen mértékben lehet „hiteles” a hallórendszerünk számára. Ugyanis, ha nem elég körültekintően választjuk meg a bemeneti paramétereket, akkor előfordulhat, hogy a szimulált hangforrás valószínűtlenül közel, illetve távol fog elhelyezkedni a hallgató személytől.



33. ábra: A hangforrás a végtelenben, és a földön

Továbbá, lehetséges a bemeneti paraméterek olyan kombinációja, amely lehetetlen helyzetet követel meg. Ilyen például, az $\alpha = +15^\circ$, $\beta = -15^\circ$. Az előjelek a vízszintes síktól történő eltérés irányát jelölik, nem a 17. illetve a 18. ábra szerint. A β

szög nyilván csak negatív értékű lehet, hiszen a reflexió a föld felől, azaz lefele irányból érkezik. Beláthatjuk, hogy ezen szögeértékek mellett, a direkt útvonal, és a reflexiós útvonal, reflexió előtti szakasza, egymással párhuzamosnak kell legyenek. Ez pedig nem lehetséges, mert a két útvonal metszetének kellene kijelölni a szimulált hangforrás helyét. A helytelen paraméterválasztás elkerülése érdekében, a hibavédelmet is beépítettem a `calculaterefdata()` függvénybe. A függvény leellenőrzi a beírt adatok helyességét. Súlyos hiba esetén azonnal javítja, valószínűtlen adatok esetén pedig figyelmeztet, de megengedi a szimuláció elvégzését. Ilyen valószínűtlen adat, például ha $\alpha = \beta = -15^\circ$. Ez nem lehetetlen, de a gyakorlatban azt jelenti, hogy a szimulálni kívánt hangforrás a földön helyezkedik el, feltéve hogy pontszerűnek tekintjük a forrásunkat.

7.7 Process - a szimulációs számítás indítása

Az Alice indítása után, a grafikus felületen, a sok más objektum között megjelenik a *Process* elnevezésű pushbutton is. Ez a gomb, alapértelmezetten nincs engedélyezve, mert a program, az indulási állapotában hibára futna a gomb használatával. A bemeneti paraméterek értékeinek megfelelő beállítása után a *Process* pushbutton engedélyeződik. Megnyomásával a `Process()` függvényt hívjuk meg. Ebben a fejezetben, ennek a függvénynek a lefutását fogom részletezni.

A `Process()` függvény az, amely lineáris lefolyást ad a szimulációs eljárásnak. A bemeneti hangfájl beolvasásától a kész hangfájlra ez a függvény hívja meg a többi függvényt. A változók értékei ebből adódnak át a többi program-modulnak és ebbe a függvénybe adódnak vissza. Már a `Process()` függvény elején egy elágazás van beépítve. A továbbhaladás iránya attól függ, hogy a *Choose angle* és a *Dynamic Reading* radiobutton-ok közül melyik van kiválasztva. A *Dynamic Reading* feladata, hogy nem egy statikus hangforrást szimulál, hanem egy folyamatosan mozgásban lévő forrást. Ez a folyamatos mozgás egy köríven történik, vagy akár egy teljes körön. Ezen tulajdonságok, a *Dynamic Reading* radiobutton aktiválására megjelenő szövegdobozokon keresztül állíthatók be. A funkció csak a HRTF szűrést használja fel a

szimuláció során, az ITD információk hozzáadása pedig nem történik meg. A HRTF szűrést az `aplHRTFDynamic()` függvényen keresztül végezzük el. Itt a bemeneti hangfájlt időben felosztjuk annyi részre ahány különböző HRTF-függvénnyel szeretnénk megszűrni a bemeneti hangfájlt. Ezeken a részeken egyesével elvégezzük a szűrést a különböző HRTF-ekkel, majd újra összeillesztjük azokat. természetesen az összeillesztéseknél egy „térbeli ugrás” lesz hallható. Ennek az ugrásnak a mértéke függ attól, hogy az összeillesztett fájlokat, egymástól mennyire „távoli” HRTF-ekkel szűrtük meg. A feldolgozott hangfájl minőségét nagymértékben lehet javítani a HRTF-ek számának növelésével

Ha a *Process* indítás előtt nem a *Dynamic Reading* funkciót választjuk, hanem a *Choose angle* –t akkor a program az egész hangfájlt egy darab HRTF-függvény felhasználásával fogja megszűrni. Ekkor úgy halljuk, hogy a hangforrás végig egy adott irányban helyezkedik el. Ehhez a szűrési módszerhez is az `aplHRTF()` függvény fog lefutni. A vizsgálataink során mindig fehér zajjal dolgozunk, ami ideális esetben, a nulladik másodperctől az utolsó másodpercig, minden frekvencián azonos amplitúdójú rezgést biztosít. Ilyen módon a fehér zajunkat periodikus jelnek tekintem, így majd a `fft()` elvégzésekor az integrálási idő tarthat a hangfájl elejétől a végéig. Erről az FFT függvény vizsgálata fejezetben bővebben is írok.

A HRTF szűrések elvégzése után az ITD információk hozzáadása történik a hangfájllhoz. Ezt a `aplDelay()` függvény végzi el. Ennek a függvénynek a lefutása függ attól, hogy a programot mediális, vagy horizontális sík béli vizsgálatokhoz szeretnénk-e használni. Az `aplDelay()`, ugyanis csak az utóbbi esetben fut, le mert a mediális síkban az ITD értékét mindig nullának választom meg, így további számításokra nincsen szükség.

Mediális sík béli szimulációk esetén, a HRTF szűrés elvégzése után, a reflexiószámítást végző program-modult fogja meghívni a `Process()` függvény. Ennek a meghívás teljesülésének az a feltétele, hogy a *Reflection* checkbox aktiválva legyen.

7.8 A HRTF szűrés vizsgálata

A HRTF szűrés elvégzése kulcsfontosságú lépés a szimulációs számításoknál, ezért ennek a függvénynek az alapos vizsgálata is nélkülözhetetlen. A vizsgálódásom alatt ki is derült, hogy komoly probléma volt a függvény lefolyásában. Ennek a hibának az okát nem tudom. Lehet, hogy egy félreértelmezett elmélet sikerült implementálni, de lehetséges, hogy csak egy elírás következménye volt az `aplHRTF()` függvény hibás működése. Azonban a hibajavítás szempontjából a fenti kérdés nem is fontos, azonban a hibát részletezni fogom, ebben a fejezetben.

Az `aplHRTF()` függvényből kimásoltam azt a néhány sort ami a működéséhez a legfontosabb, és egyben az említett hibajelenségért is felelős:

% A hangfájl amplitudó értékeinek szorzása a megfelelő HRTF szűrőértékekkel

```
for i=1:(leng/2)
    freqFiltered(i,1)= freq(i)*sqrt(LHRTF(floor((i/(leng/2)*Fs/2)/12.2)+1));
    freqFiltered(leng-i+1,1) = freq(leng-i+1)*sqrt(LHRTF(floor((i/(leng/2)*Fs/2)/12.2) +1));
    freqFiltered(i,2)= freq(i)*sqrt(RHRTF(floor((i/(leng/2)*Fs/2)/12.2)+1));
    freqFiltered(leng-i+1,2) = freq(leng-i+1)*sqrt(RHRTF(floor((i/(leng/2)*Fs/2)/12.2)+1));
end
```

% Inverz Gyors Fourier Transzformáció

```
waveFiltered = zeros(leng,2);
waveFiltered(1:leng,1) = ifft(freqFiltered(1:leng,1), leng);
waveFiltered(1:leng,2) = ifft(freqFiltered(1:leng,2), leng);
end
```

A függvény bemenete `wave`, `Fs` és kimenete, `waveFiltered` valójában egy-egy változó, ami tömb típusú. A Matlab ezeket vektorként kezeli. Megemlítendő, hogy ezt a vizsgálatot egy `test.wav` nevű hangfájl segítségével végeztem. Ez a fájl megtalálható lesz a diplomamunkámhoz mellékelte CD lemezen. Ebben az esetben a `wave` egy 76191×1 nagyságú vektor lesz, ami egy időfüggvény amplitúdójának pillanatnyi értékét tartalmazza egy-egy időpillanatban. Ehhez, tartozik még egy elengedhetetlen információ, a mintavételi frekvencia (F_s), ami 44100 (Hz). Ebből

adódik, hogy a hangfájl 1,72s időtartamú ($76191 / 44100 = 1,72$). A `length(wave)` megadja, hogy hány darab mintából áll a `wave` változó. A `nextpow2(x)` megadja, azt a legkisebb kitevőt, amelyre 2-t emelve x-nél nagyobb számot kapunk. Így a `leng = 2^nextpow2(length(wave))` kifejezés értéke 131072 lesz.

A `freq = fft(wave, leng)` kifejezés egy valós számokat tartalmazó változóból készít, egy `leng` hosszú komplex számokat tartalmazó változót. Ez egy Fourier transzformáció, ezzel kapcsolatban még további vizsgálatokat végeztem, amiket Az FFT függvény vizsgálata című fejezetben fogok kifejteni.

A `LHRTF = AmpHRTF(nameLHRTF, pathLHRTF)` kifejezés egy adott HRTF fájlból kimásolja a HRTF szűrőre jellemző erősítésértékeket a frekvencia függvényében. A `LHRTF` nevű változó egy 2048 elemű, valós számokat tartalmazó vektor lesz.

A következő két kifejezés ugyan azt a tömböt tölti fel értékekkel, az egyik előlről, a másik hátulról. Ezt az `fft()` függvény sajátosságai miatt szükséges, ilyen módon véghezvinni, két irányból. A `freqFiltered()` vektor már a HRTF szűrő megfelelő értékeivel megszorozott, szűrt hangfájl értékeit tartalmazza, amik komplex számok.

```
for i=1:(leng/2)
    freqFiltered(i,1)= freq(i)*sqrt(LHRTF(floor((i/(leng/2)*Fs/2)/12.2)+1));
    freqFiltered(leng-i+1,1) = freq(leng-i+1)*sqrt(LHRTF(floor((i/(leng/2)*Fs/2)/12.2) +1));
end
```

$$x = \text{egészrész} \left\{ \frac{\frac{i}{leng} \times \frac{Fs}{2}}{12,2} \right\} + 1$$

A művelet lényege: `freqFiltered(i,1)=freq(i)*sqrt(LHRTF(x))`. Az FFT-t hangfájl minden egyes elemét megszorozzuk, szűrjük a HRTF vektor x-edik

elemének négyzetgyökével. Az x értékének kiszámolása pedig bonyolítja a képletet. Ezt a képletet a program forráskódjából olvastam ki. Ennek a képletnek több hiányossága is van. Míg $i=0$ -nál megadja az $x = 1$ értéket addig $i = i_{\max}$ -nál nem adja vissza az $x = 2048$ helyes értéket. Ugyanis, a képlet szerint, ha $i = i_{\max} = \text{leng}/2$, akkor $x = 1808$. Ez biztosan rossz megoldás, hiszen a HRTF függvény felső kétszáz értékét biztosan nem használjuk fel a szűrési eljárásnál.

A fent leírtak után, a helyes forráskód a következő:

```
for kkk=1:(leng/2)
    freqFiltered(kkk,1)= freq_left(kkk)*LHRTF(floor((length(LHRTF)-1)/(leng/2)*kkk)+1);
    freqFiltered(leng-kkk+1,1) = freq_left(leng-kkk+1)*LHRTF(floor((length(LHRTF)-1)/(leng/2)*kkk)+1);
    freqFiltered(kkk,2)= freq_right(kkk)*RHRTF(floor((length(RHRTF)-1)/(leng/2)*kkk)+1);
    freqFiltered(leng-kkk+1,2) = freq_right(leng-kkk+1)*RHRTF(floor((length(RHRTF)-1)/(leng/2)*kkk)+1);
end
```

Az általam megírt forráskód helyessége a program futása közben könnyedén leellenőrizhető, amit természetesen megtettem. Arra a megállapításra jutottam, hogy az új `apLHRTF()` függvény a megfelelő HRTF értékekkel szorozza meg a Fourier-transzformált hangfájl minden egyes elemét.

7.9 Az FFT függvény vizsgálata

A bemeneti hangfájl irányinformációkkal való ellátása több lépcsőből tevődik össze. Ezek közül a HRTF fájlokkal való szűrés kritikus fontosságú. Ahhoz, hogy a HRTF szűrést megtehessük, először is Fourier transzformálni kell a bemeneti hangfájl időfüggvényét. Ezért is nagyon fontos ez a lépés, hiszen a többi művelet ez után következik. Ha ebben a lépésben hiba keletkezik, akkor az összes többi későbbi lépés hibás lesz, mert az alkalmazott függvények hibás bemeneteket kapnak. Ezért döntöttem a Matlabban található FFT függvény alapos vizsgálata mellett.

A Matlabban található egy beépített mintafeladat, ami egy biztos támpontot adott a függvény vizsgálatokor. Itt egy viszonylag egyszerű, két darab szinuszos jel

összegzésével kapott jelet vizsgálunk. Ezek a jelek egységnyi amplitúdóval rendelkeznek, az egyik 50Hz-es, a másik pedig 120Hz-es ().

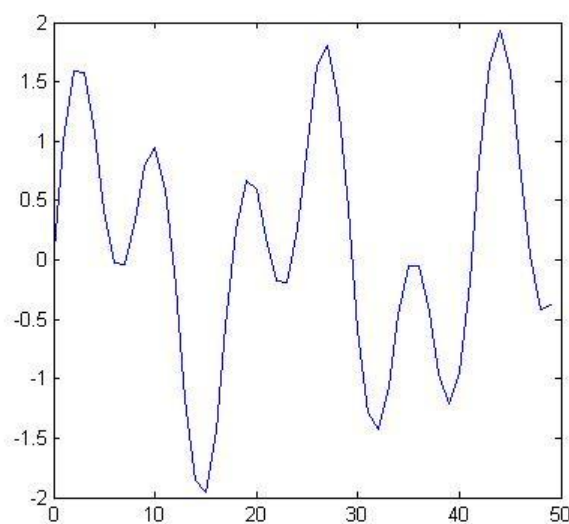
A következő sorokban a t vektort feltöltjük értékekkel. A vektor 601 elemű lesz. Elemei növekvő számtani sorozatot alkotnak, a szomszédos elemek különbsége 0,001. A legkisebb eleme nulla, a legnagyobb 0,6. A t lesz az a változó, amiben azokat az időpontokat tároljuk, amik a további vizsgálatok szempontjából fontosak lesznek. Ebből is látszik, hogy az időt nem folytonos, hanem diszkrét függvényként fogja kezelni.

```
t = 0:0.001:0.6;  
x = sin(2*pi*50*t)+sin(2*pi*120*t);  
plot(1000*t(1:50),x(1:50))
```

Az első két sor funkcióját egy „kitalált” programnyelvvvel lehet a legjobban szemléltetni:

```
for i = 0 to 600  
    t[i] = i * 0.001  
    i = i + 1  
next  
for i = 0 to 600  
    x[i] = sin(2*pi*50*t[i])+sin(2*pi*120*t[i])  
    i = i + 1  
next
```

A jól szemlélhetőség kedvéért az alkalmazott jel időfüggvényét ábrázoltam is:



34. ábra: a két szinuszjel összegének időfüggvénye

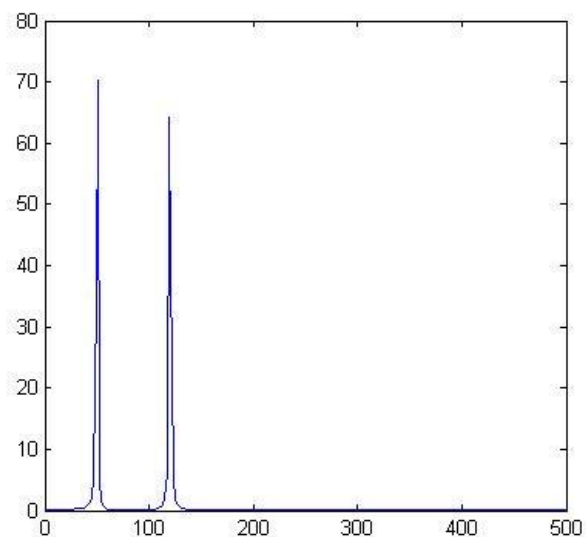
Ezek után folytattam a jelek vizsgálatát a Fourier transzformációval. Az FFT alkalmazása, önmagában még nem elég, hiszen ekkor még csak komplex számok állnak rendelkezésre az Y változóban. Ezeknek a számoknak az abszolút értékét akartam ábrázolni. A komplex számok abszolút értékét úgy is ki lehet számolni, hogy az adott számot szorozzuk a konjugáltjával, majd az eredményből gyököt vonunk. Ugyanez képlettel:

$$Y * \text{conj}(Y) = (a + jb) * (a - jb) = a^2 + jab - jab + b^2 = a^2 + b^2$$

4. ábra: a konjugálttal való szorzás azonossága

A példaprogramban a gyökvonás nem történik meg. Ez nem hibás ábrázolás, viszont tisztában kell lenni vele, hogy így a jel energiáját, teljesítményét lehet ábrázolni. Ideális esetben, a jelalakok közt nincs különbség, hiszen az amplitúdó és a teljesítmény spektrum is kettő-kettő nulla vastagságú vonalból áll. Magasságukban azonban különbségnek kell lennie, amik között az összefüggés négyzetes.

```
t = 0:0.001:0.6;
x = sin(2*pi*50*t)+sin(2*pi*120*t);
plot(1000*t(1:50),x(1:50))
Y = fft(x,512);
Pyy = Y.* conj(Y) / 512;
f = 1000*(0:256)/512;
plot(f,Pyy(1:257))
```

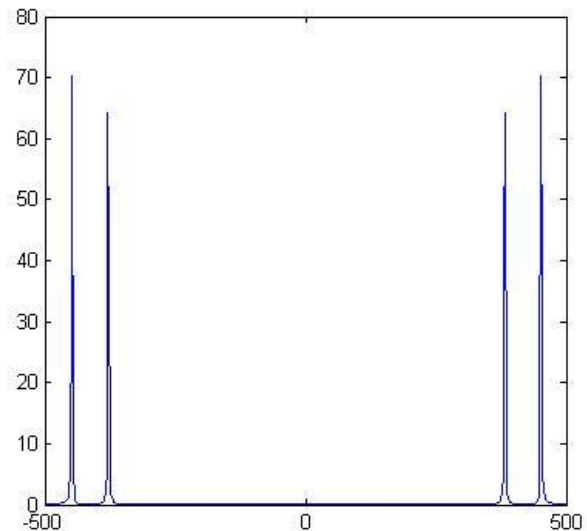


35. ábra: : Az alkalmazott jel spektrumösszetevői

Y, és Pyy ugyancsak egy-egy vektor, amik 512 eleműek. Azonban ha megfigyeljük az előző két ábrát és a hozzájuk tartozó parancssorokat, akkor látszik,

hogy Y-nak, és Pyy-nak csak az első feleit azaz, az első 256 értékét ábrázoljuk. Azonban, ha a Pyy -nak nem csak az első, hanem a második felét is ábrázolom, akkor a következő eredményt kapom:

```
t = 0:0.001:0.6;
x = sin(2*pi*50*t)+sin(2*pi*120*t);
Y = fft(x,512);
Pyy = Y.* conj(Y) / 512;
f = 1000*(-255:256)/512;
plot(f,Pyy(1:512))
```



36. ábra: Az alkalmazott jel spektrumösszetevői az egész frekvenciatartományban ábrázolva.

A `conj(Y)` komplex konjugáltat jelent, amit a komplex szám képzetes részének -1-gyel való szorzásával kapunk meg. A `.` szorzást jelent. Mivel a Matlab mátrix-okkal dolgozik (mindegyik változó egy-egy mátrix) ezért több féle szorzást kell megkülönböztetni. A `*` mátrixszorzást jelent, ami jóval összetettebb, mint két szám egyszerű szorzata. A `.` azt jelenti, hogy a két mátrixot elemenként kell összeszorozni. `Pyy`-t úgy is megkaphatjuk, ha alkalmazzuk a Matlabba beépített `abs()` függvényt, ami a zárójelbe írt kifejezés abszolút értékét adja vissza, és működik komplex számokra is. Komplex számokat tartalmazó vektor esetén is, az elemeken egyenként végzi el az abszolút érték képzést.

Az `fft()` függvény vizsgálatából a következő megfigyeléseket vontam le. A Fourier transzformáció minden esetben egy olyan frekvencia spektrum béli jelalakot hoz létre, aminek az értékei szimmetrikusak a jelalak közepére nézve. Tehát, az ábrákon látható jelalaknak csak a bal fele hordoz valós információt. A spektrum béli

jelalakon a frekvencia balról jobbra haladva, az ábra közepéig, növekszik (0 Hz-től 500 Hz-ig). Az ábra közepétől, balról jobbra haladva csökken (500 Hz-től 0 Hz-ig).

Ezt a tényt figyelembe kell venni a HRTF szűrések megvalósításakor. Ugyanis, ha a HRTF szűrő értékeivel rendre megszorozom a Fourier transzformált jel spektrumának értékeit, balról-jobbra, akkor hibás eredményt kapok. A helyes számítási mód az, ha a HRTF szűrő alacsony frekvenciákhoz tartozó értékeivel a Fourier transzformált jel „elejét” és „végét” is megszorozom. Majd a HRTF szűrő egyre nagyobb frekvenciákhoz tartozó értékeivel kell szorzást végezni, ahogy haladok a Fourier transzformált jel „közepe” felé. Ez leprogramozva így néz ki (feltéve, hogy a `freq` és a HRTF vektor is `leng` hosszúságú):

```
for i=1:(leng/2)
    freqFiltered(i)= freq(i)*HRTF(i);
    freqFiltered(leng-i+1) = freq(leng-i+1)*HRTF(i);
end
```

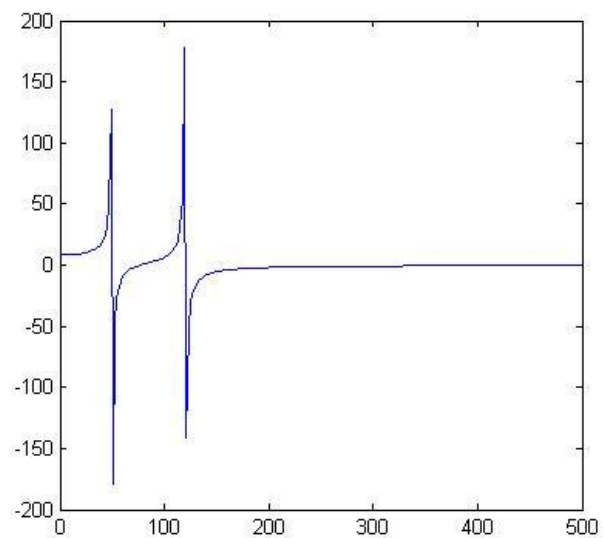
Ez a programrészlet az $f = \pm x$ Hz-hez tartozó `freq` értékekhez a HRTF szűrő ugyanazon értékét rendeli hozzá.

Érdekesség képen megnéztem, hogy milyeneredményt kapok, ha nem a Pyy-t, hanem az Y-t ábrázolom. Erre azért volt szükségem, mert nem tudtam, hogy pontosan milyen változások történnek a példaprogram negyedik sorában, és a plot függvény működését is teszteltem komplex számok esetére.

```

t = 0:0.001:0.6;
x = sin(2*pi*50*t)+sin(2*pi*120*t);
Y = fft(x,512);
Pyy = Y.* conj(Y) / 512;
f = 1000*(0:256)/512;
plot(f,Y(1:257))

```



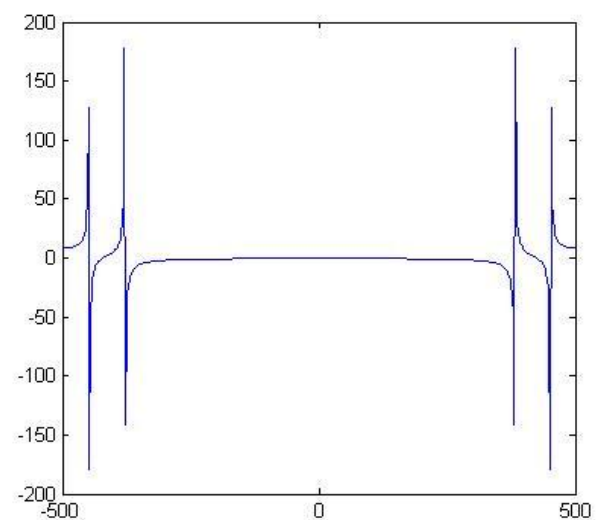
37. ábra: az Y jel alakja a frekvenciatartományban

Ugyancsak, ha a nem P_{yy} -nak, hanem Y -nak ábrázolom nem csak az alsó, hanem a felső felét is, akkor a következő eredményt kapom:

```

t = 0:0.001:0.6;
x = sin(2*pi*50*t)+sin(2*pi*120*t);
Y = fft(x,512);
Pyy = Y.* conj(Y) / 512;
f = 1000*(-255:256)/512;
plot(f,Y(1:512))

```



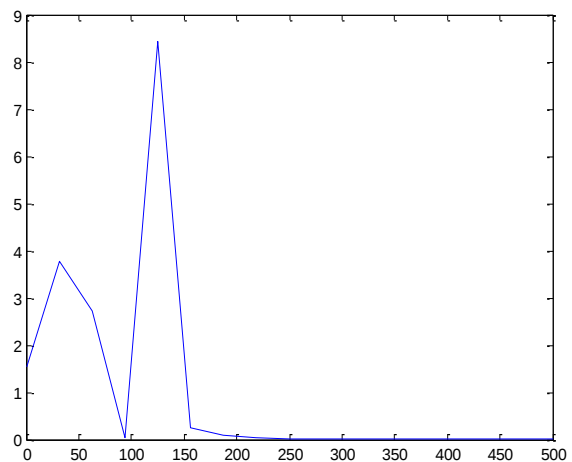
38. ábra: A teljes Y jel alakja a frekvenciatartományban

Y függvény értékei azoknál a frekvencia értékeknél mutat kitérést ahol a P_{yy} is. A Plot függvény komplex számok esetére is működik, ekkor azonban a valós és képzetes elemek külön vektorokká válnak. Ábrázolásra a képzetes számok kerülnek, a

valós számok függvényében. Tehát ha Y komplex, akkor:
`plot(Y)=plot(real(Y), imag(Y))`.

A következő vizsgálat célja, hogy kiderüljön mi történik, ha az FFT pontszámait nagymértékben csökkentem, először 32 pontos FFT-re:

```
t = 0:0.001:0.6;  
x = sin(2*pi*50*t)+sin(2*pi*120*t);  
Y = fft(x,32);  
Pyy = Y.* conj(Y) / 32;  
f = 1000*(0:16)/32;  
plot(f,Pyy(1:17))
```



39. ábra: 32 pontos FFT esetén a spektrum elkenődik

Ahogy az ábráról is látható a FFT pontszámainak a csökkentése a jel spektrumképében elkenődésként jelentkezik.

Az `aplHRTF()` függvényben, az `fft()` használata után, nem végzek további átalakításokat a jelen. Az `fft()` kimenete egy komplex számokból álló vektor. Ezen a vektor elemein fogom elvégezni a HRTF szűrést. Ez a gyakorlatban azt jelenti, hogy az `fft()` után kapott komplex számokat, `freq`, megszorozom a HRTF-függvény megfelelő elemével, ami minden esetben valós szám. Az `aplHRTF()` függvény esetében nincs szükség arra, hogy a `freq` komplex számok értékeinek algebrai alakjából kinyerjük az abszolút érték és fázis információt. Az `ifft()` függvény komplex számokat is képes kezelni. Az `ifft()` függvény tulajdonsága, hogy amennyiben komplex konjugált szimmetrikus a bemenete, akkor a kimenetén valós számokat fog eredményül adni. A valóságban ezek a valós számok tartalmazhatnak

nagyon kis értékkel rendelkező képzetes részeket, amik elhanyagolható nagyságúak. Ezeket a további felhasználás előtt levágom a `real()` függvény segítségével.

7.10 A „wav gyártás” felgyorsítása

Az Alice programot alapvetően arra lett tervezve, hogy a megfelelő paraméterek megadása után, azoknak megfelelően, szimuláljon egy adott hangforrást. A számítások elvégzése után a meg lehet hallgatni a kapott eredményt, és igény esetén el lehet menteni a számítógép egy tetszőleges könyvtárába. Az eredeti tervek szerint ez pontosan így is történt, azaz a mérések alatt szimuláljuk a hangforrást, lejátsszuk a kísérleti alanynak, az észrevételeket feljegyezzük, majd továbblépünk a következő szimulálni kívánt irányra, hangra, formulára, személyre.

Ezzel a megoldással több gond is adódott. Az egyik, hogy a program átültetése másik számítógépre, illetve operációs rendszerre, nem várt hibákat okozott. Ezek persze mind megoldhatóak, de felhívják a figyelmet arra a tényre, hogy az operációs rendszerek közötti átjárás, nem mindig problémamentes. Sőt egy-egy operációs rendszer időbeli stabilitása sem a mindig optimális. Ez csak az egyik szempont.

A másik észrevétel, hogy a paraméterek megfelelő megadása, ténylegesen nagy körültekintést igényel. Ennek tudatában is könnyen előfordultak hibák a tesztek során, amelyek sokszor megtévesztettek, akár a fejlesztés menetében is. Ennek elkerülése érdekében több hibavédelmi funkciót is beépítettem a programba, ilyen például a `PokeJoke()` függvény. Ezen hibavédelmi funkciók nagyban segítettek a felhasználhatóságot, de végül, egy ennél is biztosabb megoldás mellett döntöttünk.

Az új elképzelés szerint, előre meg kell tervezni, hogy melyik hangfájlok fognak kelleni az elkövetkező méréseken, és azokat az éles tesztek előtt legyártjuk. Ez rengeteg hangfájl legyártását jelenti. Egy megtörtént példa szerint:

*(16 horizontális irány + 6 mediális irány) * 6 fejátmérő * 4 HRTF készlet = 528 darab*

Ekkor még csak egy alkalomról tettünk említést. Azonban, ha bármelyik másik paraméter változik (gerjesztőjel, egyik HRTF készlet jobbnak bizonyul a másiknál, stb...), akkor a wav-gyártást élőről kell kezdeni, és az elvégzendő munka sokszorozódik. A wav-gyártás folyamatának gyorsítása érdekében, írtam egy függvényt, ami kisebb mértékben automatizálja ezt a folyamatot. Ez a függvény a `QuickSaveHorizontalWav()`.

Használata előtt be kell állítani a bemeneti paramétereket, ugyan úgy, mint amikor hagyományos szimulációt használunk, plusz be kell állítani, hogy melyik fejátmérő tartományt kívánjuk legyártatni. A `QuickSaveHorizontalWav()` függvényt a `QuickSaveWavs` pushbuttonnal kell meghívni. A függvény megkérdezi, hogy melyik az a célmappa, amelyikbe a kimeneti hangfájlokat menteni szeretnénk. Ezután, létrehozza a célmappában, a használt fejátmérők értékeivel elnevezett mappákat, amelyekbe a kimeneti wav-okat menteni fogja. Mindegyik mappába, annyi wav fájl fog kerülni, ahány szögértékhez tartozó HRTF fájlt talál a függvény, a kiválasztott, HRTF fájlokat tartalmazó mappában. A wav fájlok elnevezése egy háromjegyű szám lesz, amelynek az értéke megegyezik a szimulált hangforrás irányával.

Ezzel a módszerrel, már relatív gyorsan lehet wav fájlokat létrehozni. Ennél a megoldásnál viszont fontos, hogy egy nagyon jól strukturált könyvtárszerkezetben tároljuk a fájljainkat, hogy azok nehegy összekeveredjenek. A módszer előnye, hogy ha egyszer már legyártottuk jól, a kívánt hangfájlokat, a helyes paraméterekkel, akkor a legközelebbi felhasználásnál is biztosak lehetünk abban, hogy nem rontjuk el a szimulációt. További előny, hogy a hangfájlok lejátszásánál, például térhallás szimulációs vizsgálatoknál, nincs feltétlenül szükségünk Matlab-ra, vagy egy adott operációs rendszerre, mindössze egy jó minőségű, wav fájlokat lejátszani képes eszköze.

7.11 További hibajavítások, „debug-olás”, hangerősség-ingadozás

Az Alice fejlesztése, és tesztelése során rengeteg hibát kellett kijavítanom, melyeknek egy része a program forráskódjának hibájából származott, egy része viszont nem. Általánosságban, az utóbbi hibafajtákat volt nehezebb felfedezni, és így megoldani is. Ebben a fejezetben néhány ilyen hibát, és hibajavítást szeretnék megemlíteni. A sort egy hangerősség-ingadozás problémával nyitom. Azért ezt az esetet választottam, mert mindegyik közül ez volt a leglátványosabb, illetve a leginkább hallható.

A hibajelenség a következő volt. Legyártottam 16 darab wav-ot, sorban egymás után. Ezek a wav-ok, különböző hangforrás irányokat szimuláltak, de mindem más paraméterben megegyeztek. A hangfájlok hangerősségei csak olyan mértékben különbözhetek volna, amely különbségek még magyarázhatóak a HRTF-ek amplitúdó menetei közötti különbségekkel. Azonban, ennél sokkal nagyobb hangerősség-különbségek voltak hallhatóak. Ennek a jelenségnek az okát. először is a program forráskódjában kerestem. Itt szeretnék különbséget tenni hibakeresés és debug között. A hibakeresésnek általánosabb értelme van, mint a debug-nak. A debug, szó szerinti fordításban bogármentesítést jelent. Azonban, programot debug-olni, azt jelenti, hogy programfutást hibamentesíteni. Ekkor, a program már futtatásra kész állapotban van, azaz szintaktikai hiba már nincsen benne.

Tehát a fenti hibajelenség felderítését a program debug-olásával kezdtem. Ezt igen alaposan, sorról-sorra végeztem. Azonban minden függvény megfelelően működött, a változók hossza megfelelők, a változók értékei pedig helyesek voltak. Más okból kifolyólag, tanulmányoznom kellett a wav fájlok felépítését. Ekkor, két lényeges információ tűnt fel. A Matlab nem ír bele a wav-ba olyan adatokat, mint például a dátum, és hogy a wav az értékeket 0 és 1 között várja. Az utóbbi információ adott egy új ötletet. Az Alice-ban volt, van egy programrész, ami normalizálja a kimeneti változó értékeit 0 és 1 közé. Ez a programrészlet a következő:

```
m = 1.0001*max(max(abs(Outsignal)));  
Outsignal = Outsignal/m;
```

Ez a két sor, röviden, a következő műveleteket végzi. Kiválasztja a legnagyobb elemét az `Outsignal` vektornak, és ezzel a legnagyobb értékű elemmel, elosztja az összes elemet. Ez a művelet elméletben jó is lenne, de ha van egy-két, hibásan nagy elem, az `Outsignal`-ban, akkor, a hibásan túl nagy elemmel osztjuk végig a vektort, és a hangerősség túl kicsi lesz. Egy ilyen, hibásan nagy értékű elem, akármilyen zajból, illetve HRTF és bemeneti wav rögzítési hibából adódhat. Gyakorlati tapasztalat azt mutatja, hogy ilyen dirac-impulzusra emlékeztető zajok, tüskék rendszerin előfordulnak, még a leginkább professzionális hangrendszerekben is. Ezért éltem a feltételezéssel, hogy ha az `Outsignal` -ból eltávolítom az n darab legnagyobb értékű elemet, akkor ez javulást fog eredményezni a hangerősség-ingadozás problémájában.

Tehát megírtam egy függvényt, a `PutValuesBetweenland1()`-t ami képes arra hogy a bemenetére kapott $m * 2$ -es mátrix mindkét csatornájából, eltávolítsa az n darab legnagyobb elemet. A tapasztalataim, kísérletezéseim szerint, elég az első 15 legnagyobb ilyen értéket levágni. Ekkor a 15. és a 16. legnagyobb elemek értékei között már nincs nagy változás. Legkedvezőtlenebb esetben, ezek a levágott értékek valós információt hordoztak, amik így elvesznek. Ezt az információvesztést csökkentendő a 16. legnagyobb elem értékével helyettesítem a levágott értékeket, előjel helyesen. Ez, az esetlegesen hibásan elvégzett művelet, a legrosszabb esetben $15 * 1 / 44100 = 0,0003401$ sec. hosszúságú részletet érint, ami még így is kevésbé zavaró, mint a hangerősség ingadozás, amit kijavítok vele. Az általam ellenőrzött fájlokon, ez a 15 darab legnagyobb elem nem egymás után következő minták voltak, hanem elszórva helyezkedtek el a mátrixon belül. Ez a tény is azt igazolja, hogy nem a hangfüggvény egy-egy maximumhelyét találta meg a `PutValuesBetweenland1()` függvényem, hanem elszórt zajokat vágtam le vele. A legfőbb eredmény, ami igazolja az eredeti feltételezésem helyességét az, hogy a

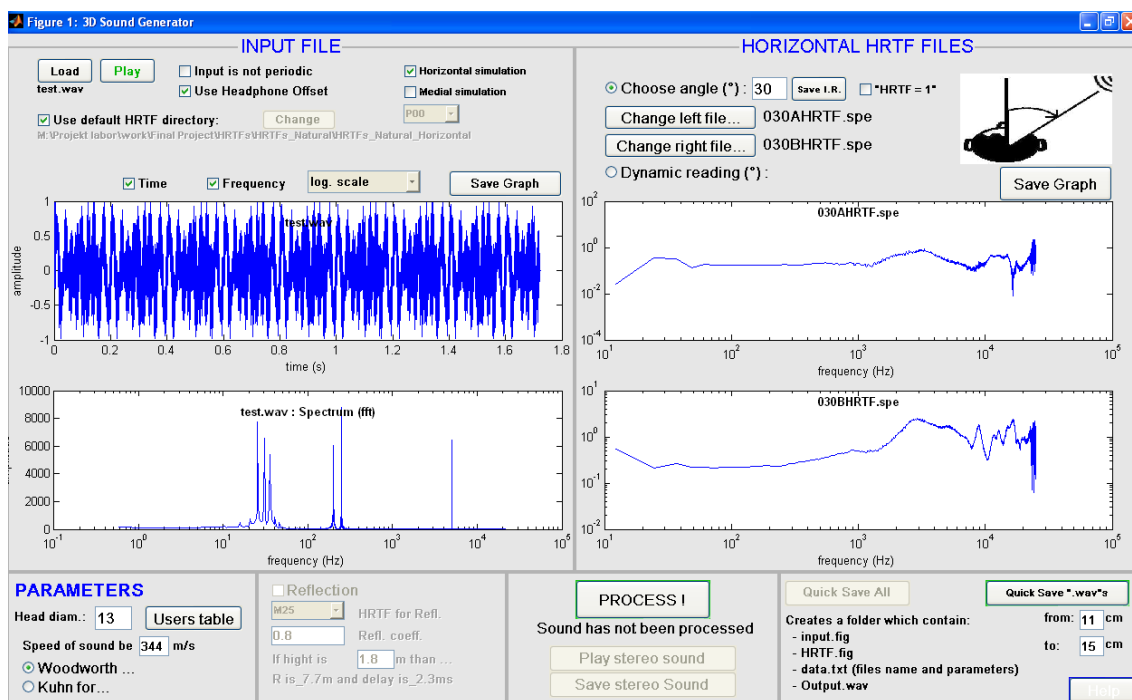
`PutValuesBetween1and1()` függvény megoldotta a hangerősség ingadzás problémáját, ugyanakkor nem okoz hallható zavart a wav-okban.

Egy másik hibajelenség az volt, hogy a az előzőekben ismertetett módon legyártott wav-oknál, néha véletlenszerű volt, a hangforrás helyének az érzete. Debugolás közben itt sem látszódott a hiba pedig az `EditAngle()` nevű függvény rossz helyen volt lefuttatva. Ez nem is lett volna feltétlenül helytelen, ha mindig ugyanolyan alapállapotból indul a program. Az `EditAngle()` többek között azt a feladatot is ellátja, hogy a beírt szögérték alapján egy "külső" változóba menti a szükséges HRTF-ek elérési útját. Viszont úgy, hogy a program nem mindig ugyanabból az állapotból indult, gyakran előfordult az az eset, hogy a `Process()` függvény előbb futott le, mint az `EditAngle()`. A probléma megoldásában most is az jelentette a nehézséget, hogy a hiba jelenléte nem akadályozta a program futását, így nem volt semmilyen visszajelzés a hiba helyére vonatkozólag.

Egy harmadik esetben az volt a hibajelenség, hogy a wav-ok sorrendben történő meghallgatásakor ellentétes irányba indult el a hangforrás. Itt a probléma oka az volt, hogy Egy-egy HRTF készlet rosszul volt elnevezve. Ezeket átneveztem. A programban nem lett volna célszerű ezt kezelni, mert nem elég csak megcserélni a jobb- és bal oldalhoz tartozó HRTF-eket. Ugyanis ez csak akkor tehető meg, ha pl.: 090AHRTF = 270BHRTF. Ekkor lett volna felcserélhető a két függvény. A hasonlóság a két függvény között, biztosan nagy, de nem megegyezőek.

7.12 az Alice grafikus felületének végleges kinézete

Az Alice program rengeteg fejlesztésen és változáson ment keresztül. Ezeknek a legnagyobb része a program forráskódjában történt, de ezeknek egy része kihatott a grafikus felület kinézetére is. a fejlesztés során egy-egy gomb, objektum nagyon sokszor helyet változtatott, mire elnyerte a végleges helyzetét. A fejlesztés során végig szempont volt, hogy a logikailag is összetartozó objektumok, funkciók, egy helyre legyenek csoportosítva, a könnyebb megtalálhatóság és a felhasználhatóság érdekében. Ez kissé zsúfoltta tette a felületet, azonban ez csak a kezdetekben lehet zavaró, a rendszeres programhasználat mellett könnyen megtanulható, megszokható a program kezelése.



40. ábra: Az Alice grafikus felületének végleges változata

Irodalomjegyzék:

- [1] Wersényi Gy, *Pszichoakusztika és az emberi térhallás alapjai*, UNIVERSITAS-GYŐR Nonprofit Kft. Győr, 2012, 116.
- [2] Hős Csaba: *Matlab hogyan*, 2005
http://www.vizgep.bme.hu/~hoscsaba/matlab/matlab_hogyan.pdf
- [3] T. T. Sandel, D.C. Teas, W. E. Feddersen, and L. A. Jeffress, „Localization of sound from single and paired sources,” *J. Acoustical Soc. Am.*, vol 27, o. 5, pp. 842-852, September 1955.
- [4] J. Blauert, *Spatial Hearing*, The MIT Press, MA, USA, 1983.
- [5] Wersényi Gy, *Pszichoakusztika és az emberi térhallás alapjai*, UNIVERSITAS-GYŐR Nonprofit Kft. Győr, 2012, 69-70.
- [6] S. Mehrgart, and V. Mellert, „Transformation characteristics of the external human ear,” *J. Acoustical Soc. Am.*, vol. 61, 6, pp. 1567-1576, June 1977.
- [7] R. L. Martin, K. I. McAnally, and M. A. Senova, „Free-Field Equivalent Localization of Virtual Audio,” *J. Audio Eng. Soc.*, vol. 49, no. 1-2, pp 14-22, 2001.
- [8] D. D. Rife, and J. Vandrkooy, „Transfer-function measurements with maximum-length sequences,” *J. Audio Eng. Soc.*, vol. 37, no. 6, pp. 419-444, 1989.
- [9] N. Matsunaga and T. Hirahara, „Fast nar-field HRTF measurements using reciprocal method,” in *Proc. of the 20th Congress of acoustics 2010 (ICA 2010)*, Sydney, Australia, 2010, 5 pages.
- [10] A. Kulkarni, S. K. Isabelle, and H. S. Colburn, „Sensitivity of human subjects to head-related transfer functions phase spectra,” *J. Acoustical Soc. Am.*, vol 105, no. 5, pp. 2821-2840, May 1999.

- [11] P. Minnaar, J. Plogsties, and F. Christensen, „Directional resolution of Head-related transfer functions required in binaural synthesis,” *J. Audio Eng. Soc.*, vol. 53, no. 10, pp. 919-929, 2005.
- [12] P. Berényi, and A. Illényi, „What does it mean for an HRTF not to have the minimal phase property?,” in *Proc. of InterNoise96*, Liverpool. UK, 1996, pp. 2127-2130.
- [13] S. Hwang, Y.-J. Park, and Y.-S. Park, „Costumization of Spatially Continous Head-Related Impulse Responses in the Median Plane,” *Acta Acustica united with Acustica*, vol. 96, no. 2, pp. 351-363, 2010.
- [14] J. Usher, and W. L. Martens, „Perceived naturalness of speech sounds presented using personalised versus non-personalised HRTFs,” in *Proc. of the 13th Int. Conf. o Auditory Display (ICAD 07)*, Montreal, Canada, 2007, pp. 10-16.
- [15] <http://www.isvr.co.uk/graphics/hatsgroup.jpg>
- [16] <http://research.microsoft.com/en-us/projects/audioprocessing/micarrayheadset.jpg>
- [17] H. Moller, „On the quality of artifical head recording systems, ” in *Proc. of InterNoise97*, Budapest, Hungary, 1997, pp. 1139-1142.
- [18] Wersényi Gy, *Pszichoakusztika és az emberi térhallás alapjai*, UNIVERSITAS-GYŐR Nonprofit Kft. Győr, 2012, 162-170.
- [19] Wersényi Gy, *Pszichoakusztika és az emberi térhallás alapjai*, UNIVERSITAS-GYŐR Nonprofit Kft. Győr, 2012, 101-102.
- [20] http://visisonics.com/preview/images/visisonics/HRTF%20Playback%20Grid-1_HRTF.jpg

Mellékletek:

A mellékletek elektronikus állományokat tartalmaznak, melyek megtalálhatók a diplomamunkához csatolt DVD lemezen, melyet a dokumentum utolsó oldalához ragasztottam.

A DVD a diplomamunkán túl, tartalmazza az:

- ALICE szimulációs program forráskódját.
- Egy öt másodperc hosszúságú, fehér zajt tartalmazó wav fájlt, amit a szimulációs program bemeneti hangfájljaként használtam.
- Az ALICE szimulációs programmal készített mintafájlokat, melyek különböző HRTF készletekhez, fejátmérőkhöz tartozó, hangforrás irányokat szimulálnak.