



**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Méréstechnika és Információs Rendszerek Tanszék

Horváth Gergely Zoltán

# **AKUSZTIKUS NYALÁBFORMÁLÁS**

Konzulensek

**Dr. Sujbert László**

**Dr. Balogh László (ProDSP Technologies Kft.)**

BUDAPEST, 2014



# Tartalomjegyzék

<b>Tartalomjegyzék .....</b>	<b>3</b>
<b>Összefoglaló .....</b>	<b>6</b>
<b>Abstract.....</b>	<b>7</b>
<b>1 Előszó .....</b>	<b>8</b>
<b>1.1 A feladat értelmezése, pontosítása.....</b>	<b>8</b>
<b>1.2 A tervezés célja.....</b>	<b>8</b>
<b>1.3 A tervezés részletes menete .....</b>	<b>9</b>
<b>2 Akusztikus nyalábformálás.....</b>	<b>11</b>
<b>2.1 Elméleti áttekintés.....</b>	<b>11</b>
2.1.1 Hullámokkal kapcsolatos fizikai alapfogalmak ([1] alapján).....	11
2.1.2 Hullámok szuperpozíciója ([1] alapján) .....	12
2.1.3 A delay-and-sum nyalábformálás alapötlete.....	13
2.1.4 A szükséges fázistolás meghatározása.....	14
2.1.5 Mikrofonjelek fázistolása .....	15
<b>2.2 Nyalábformálás a gyakorlatban .....</b>	<b>17</b>
2.2.1 Az akusztikus nyalábformálás története ([4] és [5] alapján) .....	17
2.2.2 Repülőgép .....	19
2.2.3 Vonatszerelvény.....	20
2.2.4 Mikrofon elrendezések ([3] és [6] alapján).....	20
<b>3 Előzetes kísérletek.....</b>	<b>23</b>
<b>3.1 Kísérleti rendszer .....</b>	<b>23</b>
3.1.1 Mikrofonsor .....	24
3.1.2 Előfeszítő áramkörök.....	25
<b>3.2 Kísérleti mérések.....</b>	<b>26</b>
3.2.1 Mérési eredmények.....	26
3.2.2 Mérőrendszer korlátai .....	26
<b>3.3 Analóg multiplexeres próbálkozás .....</b>	<b>27</b>
3.3.1 Hardver-összeállítás .....	27
3.3.2 Multiplexer áramkör .....	28
3.3.3 Multiplexeres elképzelés korlátai .....	29
<b>4 Akusztikus nyalábformáló rendszer .....</b>	<b>31</b>

<b>4.1 Rendszerszintű terv .....</b>	<b>31</b>
<b>4.2 Mintavevő kártyák.....</b>	<b>32</b>
4.2.1 Mintavevő eszközök .....	32
4.2.2 A DAQ-6062 kártya bemérése .....	33
4.2.3 Az USB-6363 típusszámú A/D kártya.....	35
<b>4.3 Mikrofontömb bővítése .....</b>	<b>36</b>
4.3.1 Jeltovábbítás, vezetékezés .....	36
4.3.2 Lokális előfeszítő áramkör.....	38
4.3.3 Mikrofonelrendezés .....	39
<b>4.4 Jelkondicionáló fokozat .....</b>	<b>40</b>
4.4.1 Szűrők és erősítés kiválasztása .....	40
4.4.2 Áramkörtervező-program kiválasztása .....	44
4.4.3 Áramkörtervezés menete .....	45
4.4.4 Sallen-Key tagok, előerősítő fok .....	46
4.4.5 Tápellátás, vezetékezés .....	48
4.4.6 Áramkör rajzolat .....	50
4.4.7 Jelkondicionáló kártya bemérése, hibák javítása.....	52
<b>5 Akusztikus nyalábformálás mérések .....</b>	<b>58</b>
<b>5.1 Szoftverrendszer .....</b>	<b>58</b>
5.1.1 Mintavevő kártya illesztése .....	58
5.1.2 Delay-and-sum nyalábformálás algoritmus.....	61
5.1.3 Szimulációs szoftver .....	61
5.1.4 Hangtérkép.....	64
<b>5.2 Süketszobás bemérés .....</b>	<b>66</b>
5.2.1 Mérési elrendezés, mérési körülmények.....	66
5.2.2 Az új és a régi nyalábformáló rendszer összehasonlítása .....	67
<b>6 Összegzés.....</b>	<b>71</b>
<b>Irodalomjegyzék.....</b>	<b>72</b>
<b>Mellékletek .....</b>	<b>74</b>

# HALLGATÓI NYILATKOZAT

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

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

Kelt: Budapest, 2014. 05. 26.

.....  
Horváth Gergely Zoltán

# Összefoglaló

Az iparban manapság igen elterjedt módszer az akusztikus források irányának akusztikus kamerával történő meghatározása. Ezzel a technikával lehetőség nyílik nehezen megközelíthető hangforrások iránykarakterisztikájának bemérésére, aránylag nagy pontossággal. Az akusztikus kamera rendszerint egy előre megtervezett kialakítású mikrofontömbből, egy jelkondicionáló fokozatból és egy jelfeldolgozó egységből áll. A hangkamera által rögzített jeleket az úgynevezett *akusztikus nyalábformálás* módszerével dolgozzák fel.

A diplomatermben a szakirodalomból kiindulva részletesen bemutatom egy egyszerű akusztikus nyalábformáló rendszer hardver- és szoftverelemeinek megtervezését és azok realizálását. Mérésekkel igazolom, hogy a létrehozott akusztikus kamera alkalmas a 200 Hz és 8 kHz közötti frekvenciasávba eső hangforrások aránylag pontos helymeghatározására. A kialakított akusztikus feldolgozó rendszer fő korlátai a feldolgozó algoritmus viszonylag hosszú futási ideje és a valós idejű feldolgozás lehetőségének hiánya, melyek kiküszöbölésére továbbfejlesztési tanácsokat is adok.

# Abstract

Nowadays acoustic camera systems are utilized industry-wide in order to locate sources of acoustic signals. This technique provides a means to measure the characteristics of otherwise hardly approachable objects that act as acoustic sources. The acoustic camera system usually consists of a carefully designed microphone array, a stage fit for signal shaping, and a signal processing unit. The method of processing the signals of an acoustic camera is usually referred to as acoustic beamforming.

The present thesis describes the designing and realization process of a basic acoustic beamforming system. It also presents measurements as proof that the implemented hardware-software assembly is capable of precisely locating acoustic sources in the frequency range of 200 Hz and 8 kHz. The main limitations of the systems are the slightly long runtime of the processing algorithm, and the fact that it is not capable of processing signals in a real-time manner. The thesis also provides some ideas how to improve the acoustic camera in these directions.

# 1 Előszó

Ebben a fejezetben értelmezem és pontosítom a feladatkiírásom, megindokolom az elvégzett munka szükségességét, majd röviden felvázolom a feladatom megoldásához végrehajtott tervezési és implementációs lépéseket.

## 1.1 A feladat értelmezése, pontosítása

A diplomaterv-feladatkiírásom alapján részletesen át kell tanulmányoznom az akusztikus nyalábformálás szakirodalmát, kiemelt figyelmet áldozva a gyakorlati felhasználási területekre és az elterjedt hardver összeállítások megismerésére. Az összegyűjtött elméleti ismeretek alapján terveznem kell egy saját rendszert, mely képes 16 mikrofon jelének egyidejű felvételére. A megtervezendő hardvernek tartalmaznia kell magát a 16 mikrofonból álló tömböt, egy jelkondicionáló fokozatot és egy mintavevő egységet. A tervezés során ügyelnem kell a megfelelő jel-zaj viszony biztosítására és a jelek egyidejű mintavételezésére. A megtervezett rendszert össze kell állítanom, majd méréseket kell vele végeznem, melyeket később felhasználva igazolni tudom a munkám során szintén implementálandó *delay and sum* nyalábformáló algoritmusom irodalmi eredményeknek megfelelő működését.

A kiírásban szereplő 16 mikrofon egy közelítő szám, amellyel már elegendően jó érzékenység érhető el a nyalábformálás során. A diplomatervezésem során azonban technikai okokból csak egy 15 mikrofonból álló tömböt tudtam létrehozni. Ez a változtatás a kiírásomat megalapozó elveket nem sérti, mivel az elérhető maximális érzékenységet csak kis mértékben befolyásolja. Ebből következően a feladatkiírást – a konzulensem beleegyezésével – szükségszerű pontosítanom, hogy nem 16, hanem csak egy 15 mikrofonból álló tömböt kell elkészítenem.

## 1.2 A tervezés célja

Az akusztikus nyalábformálás egy régóta ismert, sikerrel alkalmazott technika. Ennek ellenére a Méréstechnika és Információs Rendszerek Tanszék profiljában eddig nem szerepelt erre épülő eszközök fejlesztése, tanulmányozása.



A diplomatervem célja egy olyan rendszer megalkotása, melynek segítségével a legegyszerűbb, úgynevezett *delay and sum* nyálábformáló algoritmus viselkedése bemutatható. A nyálábformáláshoz szükséges mikrofontömb és a jelek felvételére szolgáló hardverelemek alkotta összeállítás a későbbiekben lehetőséget nyújthat más, összetettebb algoritmusok elméleti és gyakorlati tanulmányozására is a tanszéken.

### **1.3 A tervezés részletes menete**

A tervezés első fázisában széles körben áttanulmányoztam az akusztikus nyálábformálás irodalmát. Részletesebben megismerve annak elterjedt felhasználási módjait és a gyakorlatban rendszerint előforduló mikrofonelrendezéseket. Ezután utánajártam az akusztikus nyálábformálás pár érdekesebb felhasználási területének.

A munkámat ezután egy 15 mikrofon jeleit érzékelő, majd az azokat digitalizáló hardverrendszer inkrementális tervezésével és összeállításával folytattam. Ehhez biztos és használható alapot nyújtott a korábbi egyetemi tanulmányaim során létrehozott kísérleti berendezés és az ezzel végzett mérések. Először felvázoltam magamnak a rendszer előzetes tervét, majd elkezdtem az egyes részegységeinek tervezését, kivitelezését és tesztelését.

Először az újonnan rendelkezésemre bocsátott mintavevő kártyákat mértem be. Miután ezek alkalmasnak bizonyultak a kitűzött feladatuk ellátására, a 15 elemű mikrofontömb kialakítását, jeleinek továbbítását és a mikrofonok tápellátását terveztem meg. A tervezett egység összeállítása után mérések segítségével megállapítottam, hogy a mikrofontömből induló jelvezetékek távolabbi végén nem megfelelő jel-zaj viszony.

A jelminőség javításának érdekében, a mérési eredményekből kiindulva terveztem egy jelkondicionáló fokozatot, mely zajszűrő és előerősítő funkciókat valósít meg. A 16 csatornát kezelő fokozat előtt próbapanelen elkészítettem egy kevesebb bemenetet kezelő prototípust is, melyen ellenőriztem az előzetes elvárásoknak megfelelő működést.

Következő lépésként egy CAD programban összeállítottam a 16 csatornás jelkondicionáló fokozat áramköri tervét, majd ennek alapján legyártattam az Elektronikai Technológia Tanszéken a nyomtatott áramkört, amire később saját kezűleg ültettem be az alkatrészeket. A nyomtatott áramkör gyártási hibáinak javítása és az alkatrészek beültetése után, a jelkondicionáló egység bemérése során fényt derítettem két, a tápellátást érintő tervezési hibára. E hibák kiküszöbölése érdekében pár helyen megszakítottam a vezetékeezést, és beépítettem egy plusz tápszűrő részegységet. Ezután megismételtem a funkcionalitást ellenőrző méréseket, melyek alapján a jelkondicionáló fokozatot az eredetileg kitűzött célra alkalmasnak találtam.

A jelkondicionáló áramkör összeállításával minden részegység rendelkezésemre állt, ami az előzetes rendszerterv alapján a 15 bemeneti csatornás nyalábformálás hangfelvevő hardverrendszeréhez szükséges. A teljes rendszer összeállítását és laboratóriumi bemérését követően a helyes működés bemutatására megterveztem egy süketszobás méréssorozatot. Ez idő alatt egyben javítottam és kibővítettem a korábbi nyalábformáló kísérleteim során használt Matlab programot is. A süketszobás méréseknél ügyeltem arra, hogy a mérési eredmények a korábbi kísérleti és az irodalomban szereplő eredményekkel összehasonlíthatóak legyenek. A süketszobás méréseken felül elvégeztem egy tantermi méréssorozatot is, hogy a hardver-összeállítás és a nyalábformáló algoritmus felhasználási korlátait megismerjem.

Az előzőekben említett mérések során begyűjtött adatokat kiértékeltem, a kapott eredményeket összehasonlítottam a korábbi, 8 mikrofonnal végzett kísérleti eredményeimmel és az irodalommal egyaránt. A munkám zárásaként a levont tapasztalatok alapján felvázoltam a témám pár továbbfejlesztési lehetőségét.

## 2 Akusztikus nyalábformálás

### 2.1 Elméleti áttekintés

Ebben a fejezetben részletesen tárgyalom az akusztikus nyalábformálás megismeréséhez szükséges fizikai és jelfeldolgozási alapismereteket.

#### 2.1.1 Hullámokkal kapcsolatos fizikai alapfogalmak ([1] alapján)

A rugalmas közegben létrehozott deformáció a térben tovaterjed. Az ilyen közegek egy pontjában létrehozott rugalmas deformáció egy idő után a pontban eltűnik, és másik hely(ek)en jelentkezik egy bizonyos idővel később. Ezt a jelenséget mechanikai hullámnak nevezzük. A hanghullám a térben terjedő, háromdimenziós hullámok csoportjába sorolható.

A térben terjedő hullámoknál az azonos rezgési állapotú pontok alkotta felületet hullámfrontnak nevezzük. A pontszerű hullámforrások által kibocsátott hullámok hullámfrontjai homogén izotrop közegben gömbfelületeket alkotnak. A síkhullám olyan térben terjedő hullám, melynek hullámfrontjai párhuzamos síkok. A hullámforrástól elegendő nagy távolságban a közeg kis tartományán áthaladó hullám jó közelítéssel síkhullámnak tekinthető. A hullám terjedési iránya a hullámfrontra merőleges egyenes, azaz a hullámfront normálisa.

A hullámforrás egy bizonyos rezgésállapota a hullámforrástól  $x$  távolságra lévő pontban csak  $t$  idő múlva jelenik meg. Ennek a távolságnak és az időnek a hányadosa a terjedési sebesség, melynek jele:  $c$ .

Periodikus egy hullám, ha a hullámforrás rezgése periodikus. A periodikus hullámot négy mennyiség jellemzi: az előbb definiált terjedési sebesség, az amplitúdó, a frekvencia és a hullámhossz. Egy periodikus hullámot harmonikus (szinuszos) hullámnak nevezünk, ha a hullámforrás kitérése az idő szinuszos függvénye. A harmonikus, pontszerű hullámforrástól  $x$  távolságban lévő pont pillanatnyi kitérését  $t$  időpillanatban az alábbi képlettel határozhatjuk meg.

$$\xi(x, t) = A \sin\left(\omega \left(t - \frac{x}{c}\right) + \varphi\right) = A e^{j\left(\omega \left(t - \frac{x}{c}\right) + \varphi\right)}$$

A képletben  $A$  az amplitúdót,  $\omega$  a szinuszos gerjesztés körfrekvenciáját,  $\varphi$  pedig a gerjesztés kezdőfázisát jelöli.

A közeg két részecskéjének rezgési állapota akkor egyezik meg, ha a fázisszögük (a szinusz függvény argumentuma) egyenlő, vagy  $2\pi$  egész számú többszörösével tér el. Ebben az esetben azt mondjuk, hogy a két részecske fázisa megegyezik, azonos fázisban vannak.

### 2.1.2 Hullámok szuperpozíciója ([1] alapján)

A közeg két különböző pontjából indított hullámok zavartalanul áthaladnak egymáson. A találkozásuk helyén az eredő kitérés minden időpillanatban a két hullám által létrehozott kitérések vektori eredője lesz. Ezt a jelenséget a hullámok szuperpozíciójának nevezzük.

Észlelhető interferenciakép jön létre azonos frekvenciájú, időben állandó fáziskülönbségű és azonos rezgési síkú hullámok találkozásakor. Az előbb felsorolt tulajdonságokkal rendelkező hullámokat koherens hullámoknak nevezzük.

Koherens hullámok szuperpozíciója során a közeg egy adott pontjában az eredő rezgés amplitúdója akkor lesz maximális azaz a szuperponált hullámok amplitúdóinak összege, ha a hullámok fáziseltérése  $2\pi$  egész számú többszöröse. Ezt az esetet nevezik maximális erősítésnek. Ha két szuperponált koherens hullám fáziseltérése  $(2k + 1)\pi$ ,  $k \in \mathbb{Z}$  alakú azaz  $\pi$  páratlan számú többszöröse, a kialakuló hullám eredő amplitúdója az amplitúdók különbségének abszolút értéke lesz. Ez a maximális gyengítés. Ha a két hullám amplitúdója megegyezik, az eredő kitérés 0. Ezt nevezzük kioltásnak.

Általános esetben a tér egy adott pontjában találkozó két koherens hullám szuperpozíciója során kialakult eredő hullámforma amplitúdóját az alábbi összefüggés szerint számíthatjuk, ahol  $\varphi$  a két hullám közötti fáziskülönbség:

$$A_e = A_1 + A_2 \cos(\varphi)$$

Ez a képlet általánosítható kettőnél több, egymással interferáló hullám esetére is oly módon, hogy először kiválasztunk egy referencia hullámot, a többi fázishelyzetét ehhez képest nézzük, majd az egyes amplitúdókat a szuperpozíció során normáljuk a fáziskülönbségek koszinuszával.

### 2.1.3 A delay-and-sum nyalábformálás alapötlete

Az akusztikus nyalábformálás célja egy előre kijelölt forrásból érkező hang kiemelése a más, ettől eltérő forrású hangok elnyomásának segítségével. A nyalábformálás alapja az előző fejezetben tárgyalt koherens mechanikai hullámok interferenciája.

A nyalábformálás során több mikrofon segítségével vesszük fel a közegben terjedő hangokat. Mivel a mikrofonok a tér különböző pontjaiban helyezkednek el, a különböző mikrofonokon minden időpillanatban egymástól eltérő fázisban látjuk a hangforrások jeleit. A hang terjedési sebességének és a mikrofonok helyzetének ismeretében kiszámítható, hogy adott forrásból érkező hanghullámok épp milyen fázishelyzetben érkeznek a mikrofonokhoz:

$$\varphi_i(f, t) = \varphi_0 + 2\pi f \frac{d_i}{c}$$

Egy kiválasztott pontból érkező hangot úgy tudunk kiemelni, hogy először meghatározzuk a fenti képlet alapján, hogy melyik mikrofonhoz milyen fáziskésleltetéssel érkezik a választott pontból származó hanghullám. Ezután egyik mikrofont referenciának választva, a többi mikrofonnál vett hullámok fázisát úgy toljuk el, hogy azok megegyezzenek a referencia mikrofon által vett jel fázisával.

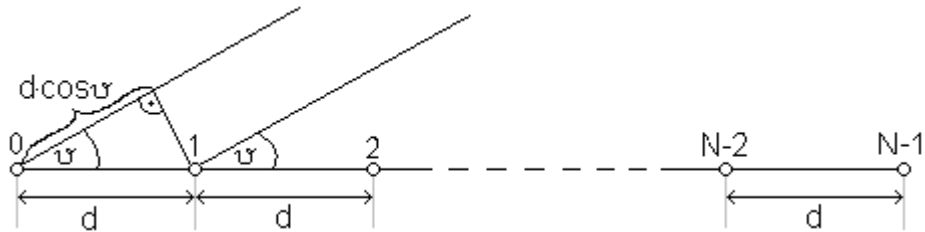
Mivel az így kapott jelek koherensnek tekinthetők, szuperponálva őket érvényesül az előzőekben tárgyalt képlet az eredő jel amplitúdójára. A kijelölt forrásból származó jelek fázistolás után azonos fázisba kerülnek, így összeadva őket a teljes erősítés jelenségét figyelhetjük meg. A kijelölttől eltérő forrásból származó jelek a fázistolás után nem kerülnek azonos fázisba, így szuperpozíció után az eredőjüknek kisebb amplitúdója lesz, mint az azonos fázisú esetben. Ezzel a technikával lehetséges a kiszemelt hangforrás jelének kiemelése a többi forrás jelei közül.

A nyálábformáláshoz először tehát meg kell határoznunk, hogy a pontszerűnek tekintett mikrofonokhoz milyen fázisban érkezik a kiválasztott forrásból származó hang, ez után úgy kell késleltetni a felvételeket, hogy a kijelölt forrásból érkező jelek azonos fázishelyzetbe kerüljenek, majd a vett jeleket összegezzük és az így kapott időfüggvényt normáljuk (elosztjuk) a mikrofonok számával.

#### 2.1.4 A szükséges fázistolás meghatározása

A mikrofonok által vett jelek szükséges fázistolásának meghatározására két módszer terjedt el a gyakorlatban. Az elkülönítés alapja a kiemelendő hangforrás helyének megadása.

Az első algoritmus csak a hangforrás mikrofonokkal bezárt szögét veszi figyelembe. Ez akkor alkalmazható, amikor a forrás nagyon messze található, ebből adódik a végtelen fókusz távolságú nyálábformálás elnevezés. Ilyenkor a mikrofonokra érkező gömbhullám a nagy görbületi sugár miatt síkhullámnak tekinthető, a beesési irány egyértelműen megadható a hullámfront normálisa és a mikrofonok egyenesé között bezárt szöggel. A végtelen fókusz távolságú nyálábformáláskor feltételezett geometriát mutatja az alábbi ábra homogén, lineáris mikrofonelrendezés esetén:



**2.1 ábra:** Végtelen fókusz távolságú nyálábformálás (forrás: [2])

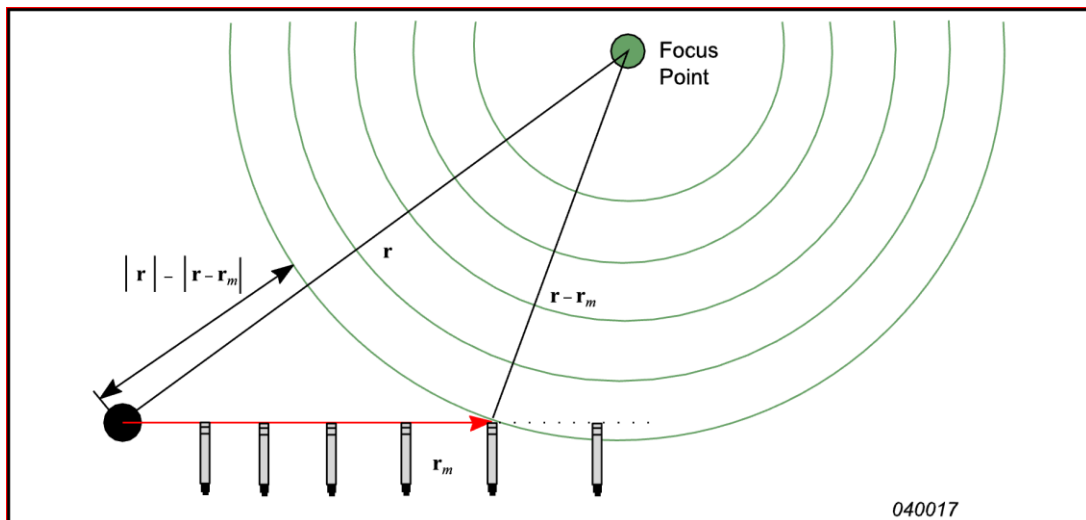
Az  $i$ -edik és a  $k$ -edik mikrofonok által érzékelt hanghullámok út- és fáziskülönbsége az alábbi képletek alapján számíthatók ebben az esetben:

$$\Delta d_{i,k} = d(k - i) \cos \vartheta$$

$$\Delta \varphi_{i,k} = 2\pi f \frac{\Delta d_{i,k}}{c}$$

A másik elterjedt nyalábformálási algoritmus a szükséges fázistolás meghatározására az úgynevezett véges fókusz távolságú nyalábformálás. Mint a nevéből is adódik, a fókuszpont (kiemelendő forrás) ekkor nem végtelen távolságra van a mikrofonoktól. Ekkor minden pontszerűnek tekintett mikrofon fókusztól való távolságát meg kell határozni a szükséges fázistolások kiszámítása érdekében. Ehhez a fókuszpont helyét annak koordinátaival célszerű megadni.

A véges fókusz távolságú nyalábformálás távolságának meghatározását a következő ábra szemlélteti:



2.2 ábra: Véges fókusz távolságú nyalábformálás (forrás: [2])

A mikrofonok és a fókuszpont koordinátáit (helyvektorait) ismerve az  $i$ -edik mikrofon jelének a referencia mikrofonhoz viszonyított fáziseltérése az alábbi képlet alapján számítható.

$$\Delta d_{0,i} = |\bar{r}_0| - |\bar{r}_0 - \bar{r}_{0,i}|$$

$$\Delta \varphi_{0,i} = 2\pi f \frac{\Delta d_{0,i}}{c}$$

### 2.1.5 Mikrofonjelek fázistolása

Az előző fejezetekben csak az egy spektrális komponensű, tisztán szinuszos, koherens hullámok szuperpozícióját tárgyaltam. Ilyen tulajdonságú hullámok a gyakorlati életben csak ritkán fordulnak elő, így felmerül a kérdés, hogy a nyalábformálással fel lehet-e dolgozni a több spektrális komponensű harmonikus, illetve a tranzienst jeleket is?

Amennyiben a rugalmas közeg, amiben a hang terjed, homogénnek tekinthető, az összetett jelek feldolgozására is lehetőség nyílik. Ebben az esetben mivel a nyalábformálás során egy legalább rövid ideig stacionárius hangforrásra koncentrálnunk megállapítható, hogy a fókuszról induló, több spektrális komponenset tartalmazó hullámok fáziskülönbsége frekvenciakomponensenként időben állandónak tekinthető. Tehát a mikrofonoknál észlelt jelek spektrális komponensei egymással koherensek, így komponensenként megfigyelhetőek a koherens hullámok eddig tárgyalt tulajdonságai.

A mikrofonjelek késleltetése egy  $-\pi < \Delta\varphi < \pi$  intervallumba transzformált, frekvenciatartományban elvégzett fázistolással megoldható abban az esetben, ha a hangforrás időben állandónak tekinthető. Amennyiben a kijelölt hangforrásra nem teljesül ez a feltétel, a jelek feldolgozáshoz ki kell jelölni egy maximálisan nagy időablakot, amelyben a kiszemelt hangforrás stacionáriusnak tekinthető. Ebben az esetben az egész mintaszámnyi késleltetéseket az időtartományban kell elvégezni, a tört késleltetések pedig megfeleltethetők egy frekvenciatartománybeli fázistolásnak. Az utóbbi, tranziens jelek esetén használandó módszer alkalmas időben állandó, harmonikus összetevőkből álló hangforrás jeleinek feldolgozására is.

A frekvenciatartománybeli fázisváltoztatás megvalósítható egy szűrő segítségével, vagy az egyes komplex frekvenciakomponensek egyenként történő forgatásával.

A szűrős megoldáshoz egy lineáris fáziskarakterisztikájú FIR szűrő a megfelelő választás. A szűrő együtthatóinak meghatározása számításigényes, de amíg a fókuszpont helye nem változik, elegendő ezeket egyszer kiszámolni.

A frekvenciakomponensek egyenként való forgatása nagyobb számítási igényű, mint a FIR szűrős feldolgozás. Ezt abban az esetben érdemes alkalmazni, amikor a fókusz helyzete időben nem állandó.



## 2.2 Nyalábformálás a gyakorlatban

### 2.2.1 Az akusztikus nyalábformálás története ([4] és [5] alapján)

Az akusztikus nyalábformálás ősenek tekinthető akusztikus antenna gyakorlati felhasználásának története az 1800-as évek közepére nyúlik vissza. Alfred M. Mayer, a New Jersey-i Műszaki egyetem professzora, 1880-ban szabadalmaztatta az Egyesült Államokban az úgynevezett topofont, mely a ködös időben segítette a tájékozódást.

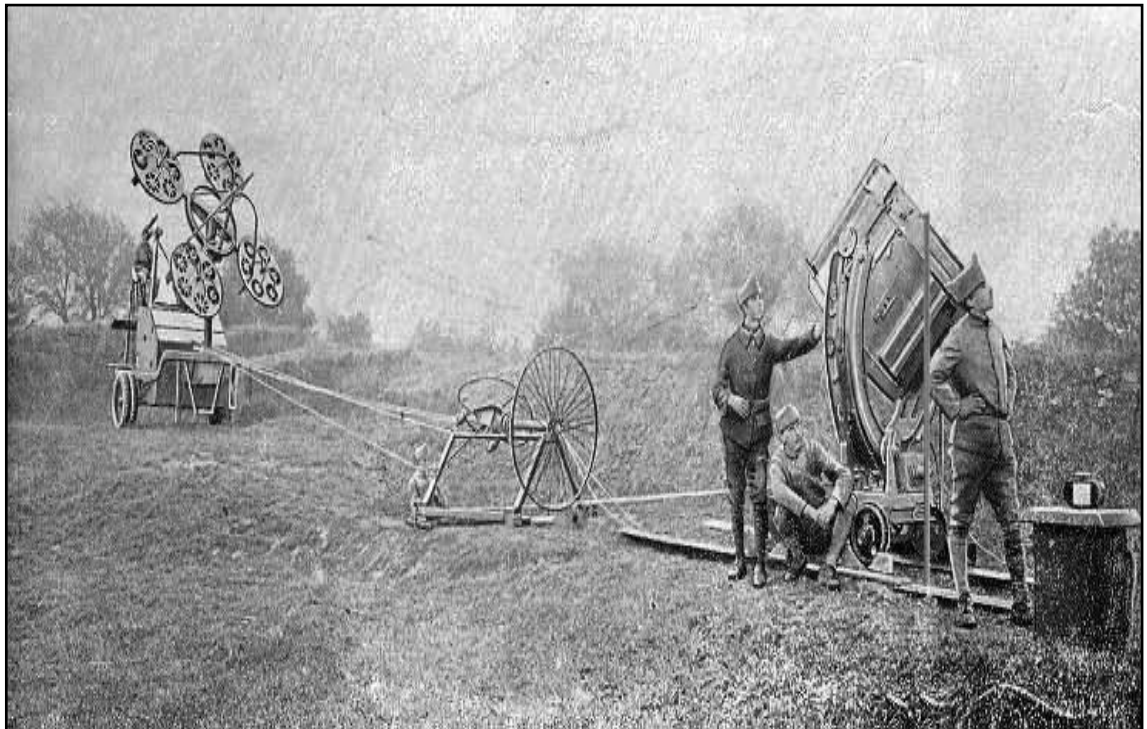


**2.3 ábra:** Mayer professzor topofonja (forrás: Scientific American, 3rd July 1880)

Ez a mai szemmel nézve megmosolyogtató látványt nyújtó szerkezet két, vállra szerelhető tölcsérből állt, melyek által összegyűjtött hangot azonos hosszúságú csövek juttatták el a felhasználó füléig. A tölcsérek orientációját a rögzítésük alatt található tárcsákkal lehetett állítani. Mint az ábrán látható, a csövek közepén közös légtérben folytatódnak, itt jön létre a tölcsérekből érkező hanghullámok szuperpozíciója. A szerkezet a tölcsérek irányából érkező hangokat felerősíti, a más irányból érkezőeket pedig elnyomja.

A szakirodalom beszámol még két, 1880-nál korábban elfogadott szabadalomról is, melyek egy-egy, a fentihez hasonló kialakítású eszköz használati jogára nyújtanak oltalmat. Ezek közül a legkorábbi szabadalom Benjamin R. Smith nevéhez fűződik, és 1859-ben fogadták el.

Az akusztikus antennák az első világháború alatt éltek a hőskoruk. Mind az antant, mind a központi hatalmak oldalán felsorakoztak különféle hangfeldolgozó eszközök, melyek fő rendeltetése az ütegek és a repülőgépek helymeghatározása volt. A **2.4 ábra** négy hatszög alakban elrendezett tölcserendszerből álló repülőgép észlelő rendszert mutat be. A tölcsérekből egyenlő hosszú csöveken érkezik a hang a reflektort irányító francia katona füléhez. A hangok összegzését itt is egy közös torkolati cső látja el.



**2.4 ábra:** Nyalábformáláson alapuló, francia repülőgép észlelő rendszer (1916)

### 2.2.2 Repülőgép

Bár az alapelv már az 1800-as években ismert és alkalmazott volt, az első mikrofontömbre épülő nyalábformáló rendszer terve egészen 1974-ig váratott magára, amikor is John Billingsley előállt egy elektronikus elven működő akusztikus teleszkóp ötletével. 1976-ra Billingsley társával, Kinns-szel már implementált is egy valós idejű feldolgozásra alkalmas, 14 kondenzátor mikrofont tartalmazó akusztikus teleszkóppal. A mikrofonok jelét aránylag hosszú, 100 méter hosszú kábeleken vezették el egy személyi számítógépig, ahol két darab, 8 bites A/D átalakítóval, 20 kHz mintavételi frekvenciát használva digitalizálták az érkező jeleket. A rendszert a Concorde sugárhajtóművének (Rolls-Royce – SNECMA) diagnosztikai vizsgálatára használták. Billingsley már 1976-ban felvetette a korrelált források elkülönítésének ötletét, de a Concorde hajtóműveknél ez nem volt használható, így sokáig feledésbe merült. Manapság a modern szakirodalom kezdi ismét elemezni ezt az ötletet, mely tűzként terjed a modern orvosi és idegtudomány diagnosztikában is.

Míg Billingsley csak a földön vizsgálta a rögzített sugárhajtóműveket, 1986-ban Howell kidolgozott egy módszert a Lockheed Tristar repülőgépek repülési zajának mérésére. A mozgó járművek vizsgálatakor a Doppler effektus kiküszöbölése is szükséges műszaki követelmény. A repülőgép pályájának meghatározására Howell egy állványra szerelt kamerát használt. Mivel akkoriban a képfeldolgozás még nagyon gyerekcipőben járt, Howell a gép pályáját az azt pontosan követő kamera forgásszög-idő és dőlésszög-idő függvényeiből határozta meg a repülőgép pontos méreteinek ismeretében. A repülési irányban így 2.16 méter, arra merőlegesen 0.15 méter pontosságot tudott elérni. A repülőgép magasságát 0.88 méteres négyzetes hibával határozta meg. Howell ezen felül vizsgálta a túl- és az alulmintavételezés jel-zaj viszonyra gyakorolt hatását is, és arra jutott, hogy legalább kétszeresen túl kell mintavételezni a jeleket, hogy legalább 35 dB-es jel-zaj viszont lehessen elérni.

A későbbi repülőgépekkel végzett kísérletek a mikrofontömb elemszámának bővítésére, irreguláris elrendezésekkel való kísérletekre és a Monte Carlo módszer hatékonyságának vizsgálatára összpontosítottak. A nyalábformálást a mai napig használják a repülőgépiparban minőségbiztosítási és hibadiagnosztikai célokra.

### 2.2.3 Vonatszerelvény

A mozgó vonatszerelvények tengelyvizsgálatakor a kezdeti időkben (1970-es években) a legnagyobb kihívást az jelentette, hogy aránylag rövid, körülbelül 5 méteres szakaszon kellett méréseket végezni 80-100 km/h sebességű járművek hangjáról. Az első méréseket 15 elemű lineáris mikrofontömbökkel végezték. Egy tömböt vízszintesen, egyet függőlegesen helyeztek el a sín mellett, hogy mindkét tengely mentén reprezentatív méréseket tudjanak végezni. A mikrofontömbök segítségével keskeny-sávú spektrumanalíziseket hajtottak végre, mozgó fókuszponttal. A térbeli átlapolódás következtében megjelenő oldalnyalábokat úgynevezett beágyazott, irreguláris tömbökkel próbálták kiküszöbölni, mely egyre csökkenő távolságban elhelyezett mikrofonokból állt. G. Élias (1999) ötlete alapján egy speciális, spirál alakú elrendezéssel sikerült ezt a hatást a lehető legkedvezőbb szintre csökkenteni.

### 2.2.4 Mikrofon elrendezések ([3] és [6] alapján)

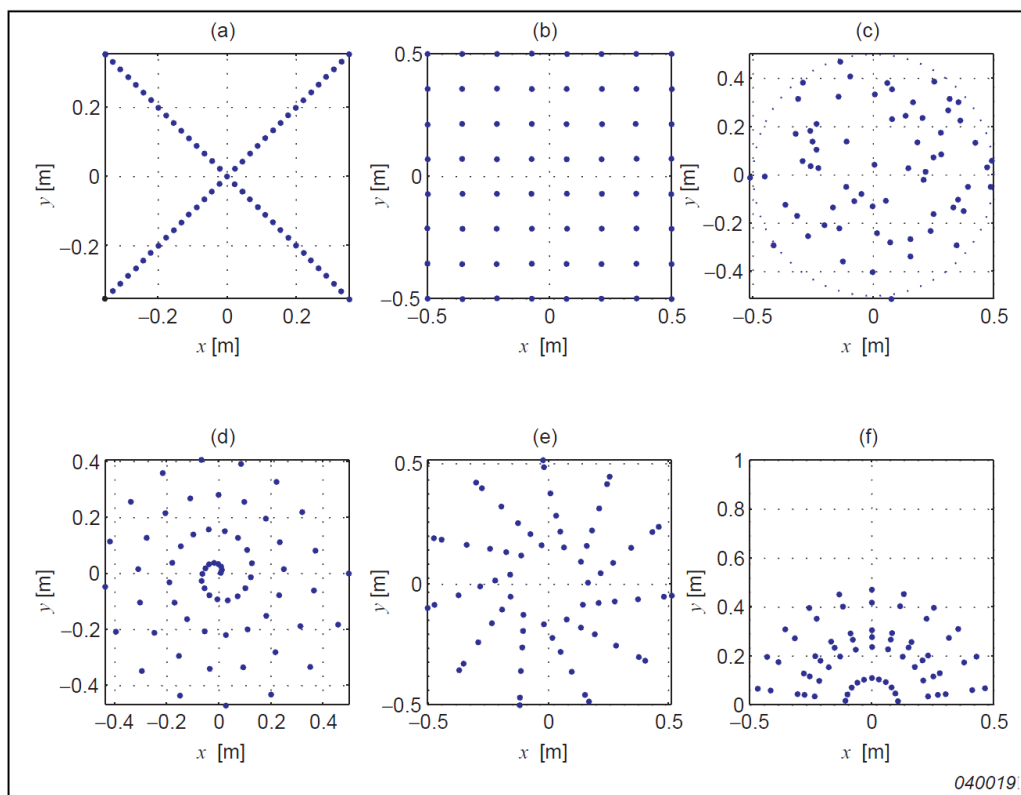
A nyalábformálás során a felhasznált mikrofonok elrendezésének fontos szerepe van. Ettől függ az algoritmus frekvencia érzékenysége és a fókusz kiemelésének mértéke.

A mikrofonelrendezések két nagy csoportba sorolhatók: hagyományos és rendhagyó elrendezések. Hagyományosnak nevezzük mindazon elrendezéseket, melyekben a mikrofonok azonos távolságra találhatók a legközelebbi szomszédjuktól. Irreguláris minden olyan elrendezés, amire az előző definíció nem teljesül.

A hagyományos elrendezések aránylag kis költséggel kialakíthatók, és kevesebb tervezési időt igényelnek az irreguláris elrendezéseknél. Hátrányuk, hogy a térbeli átlapolódás miatt a főnyalábok mellett úgynevezett melléknyalábok alakulnak ki a hangtérképen. Ez meghamisítja a hangtérképet, így sok alkalmazásban megpróbálják csökkenteni ezt a hatást.

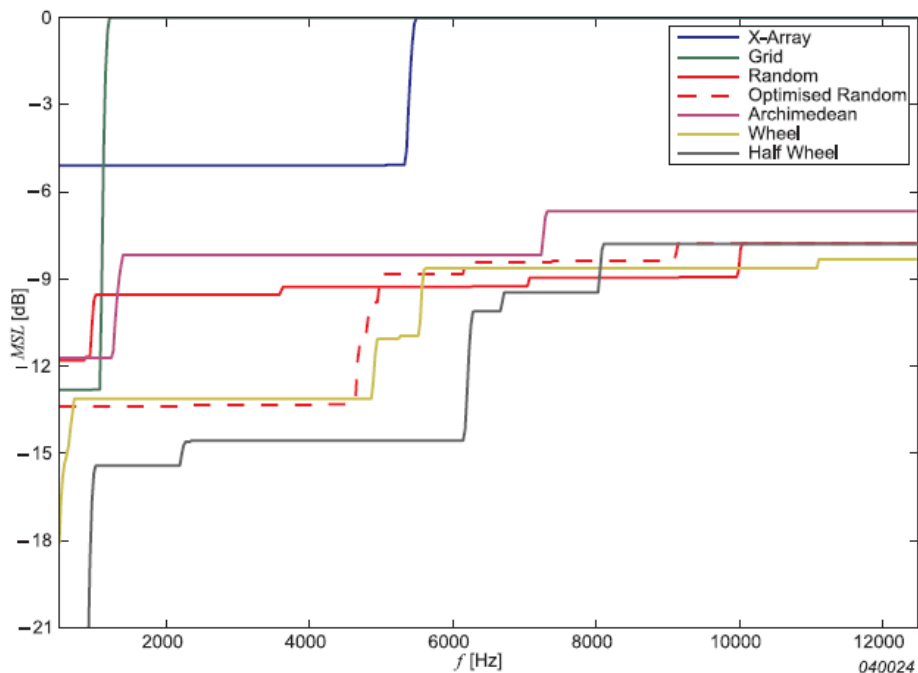
Az irreguláris elrendezések előnye a hagyományosokkal szemben pont a kedvező oldalnyaláb elnyomásban rejlik. Ezeket az elrendezéseket úgy alakítják ki, hogy a mikrofonok helyzetében a lehető legkisebb legyen a redundancia. Ilyen elrendezés például a kerék, vagy a spirál alakzat. Ezek hátránya az időigényes tervezés és a viszonylag költséges kialakítás.

Az oldalnyaláb elnyomás mértéke az úgynevezett *Maximal Side-Lobe Level* (MSL), mely a legnagyobb melléknyaláb főnyalábhoz viszonyított szintjéből határozható meg. A következő két ábrán a gyakorlatban elterjedt reguláris és irreguláris mikrofon elrendezéseket, és azok MSL értékét mutatom be.



**2.5 ábra:** Elterjedt mikrofontömb elrendezések

- a.) Kereszt; b.) Mátrix;
- c.) Gauss-i; d.) Arkhimédészi;
- e.) Optimált kerék; f.) Fél-kerék



2.6 ábra: Jellegzetes MSL értékek

A 2.6 ábráról leolvasható, hogy a Nyquist-frekvencia (átlapolódás határa) felett a kereszt és a mátrix elrendezések esetében a legnagyobb oldalnyalábok szintje megegyezik a főnyaláb szintjével. Ezzel szemben az irreguláris elrendezéseknél sokkal kisebb mértékben emelkedik a melléknyalábok relatív szintje a mért frekvenciához képest, és soha nem éri el az 1-et (0 dB-t) az MSL értéke.

A gyakorlati életben a mikrofontömb alakzatán kívül fontos tulajdonság az egyes mikrofonokon vett jelek súlyozása is. Legegyszerűbb esetben minden mikrofon jelét egységnyinek vesszük az összeadáskor. T. Betlehem és R. C. Williamson kimutatta, hogy ez nem minden esetben helyes megközelítés. Az emberi hang feldolgozásakor kimutatták, hogy a hangforrások (emberek) mögött elhelyezkedő mikrofonokat kisebb súllyal érdemes figyelembe venni, így természetesebb hatás és jobb érzékenység érhető el.

Másik érdekes ötlet a súlyok adaptív, lágy számításokon alapuló hangolása valós időben, mely a [8] irodalom alapján különböző zajcsökkentő alkalmazásoknál nagyon kedvező zajelnyomást eredményezhet.

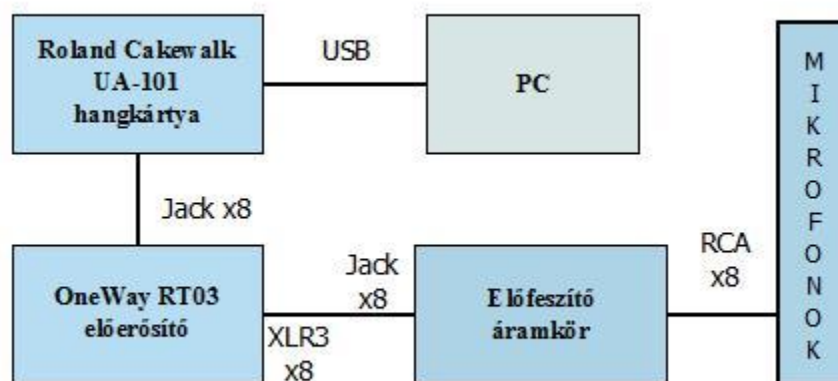
### 3 Előzetes kísérletek

Mivel a Diplomamunkám alapjául az Önálló laboratórium 1-2 tárgyak során elvégzett munkám szolgált, ebben a fejezetben röviden ismertetem az előzetes kísérletezésekhez összeállított rendszert, a kísérleti eredményeimet, majd ezekből kiindulva ismertetem az irányt, ami felé a diplomatervezés készítése során elindultam.

#### 3.1 Kísérleti rendszer

Az Önálló laboratórium 1-2 tárgyak keretén belül összeállítottam egy kísérleti rendszert, amely segítségével alapvető nyálábformáló kísérleteket végeztem. Ebben az alfejezetben a rendszer hardverfelépítését mutatom be részleteiben.

A DSP laborban rendelkezésemre állt egy OneWay RT03 Octal Mic Preampfier típusú előerősítő és egy 8 bemeneti csatornával rendelkező, Roland Cakewalk UA-101 típusú hangkártya, mely USB-n keresztül csatlakoztató a PC-hez. Ezeket kiegészítve egy mikrofontömbbel és a mikrofonokhoz szükséges előfeszítő áramkörökkel lehetőségem nyílt valós jeleken tanulmányozni a nyálábformálást.



3.1 ábra: A kísérleti hardverrendszer blokkdiagramja

### 3.1.1 Mikrofonosor

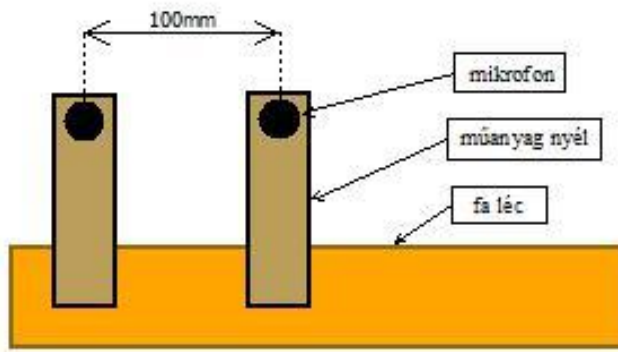
A rendelkezésre álló hangkártya és az előerősítő is 8 bemeneti csatornát támogat, így kézenfekvő volt egy 8 darab mikrofont tartalmazó felvevő rendszer létrehozása. A mikrofonok elrendezésére két alapvető megoldás a 3x3-as, közepén hiányos mátrix és az 1x8-as lineáris beállítás. Mérnöki becslés alapján a konzulensemmelet azt gondoltuk, hogy a lineáris elrendezés az egyik térbeli irányra való érzéketlenség árán jobb felbontást biztosít egy, a mikrofonok egyenesével párhuzamos síkban. Ezt a feltételezésünket későbbi szimulációk igazolták. Mivel a mikrofonok elrendezése a későbbiekben aránylag egyszerűen megváltoztatható, a döntés támogatására nem fektettünk be több erőforrást, és az 1x8-as lineáris elrendezést választottuk.

A mikrofonok közötti távolság megválasztásakor az egyszerűséget és átláthatóságot szem előtt tartva egyenlő távolságban elhelyezett mikrofonokban gondolkodtunk. Az irodalomban szereplő mérési eredményekkel való összehasonlíthatóság érdekében a 10 cm-es távolság mellett döntöttünk. A térbeli mintavétel szabálya szerint ilyen mintavételi távolság mellett a legalább 20 cm hullámhosszú jelek állíthatóak vissza egyértelműen. Ha a mintavett jel hullámhossza kisebb, mint 20 cm, fellép az úgynevezett térbeli átlapolódás. Az átlapolódás-mentesen visszaállítható jel maximális frekvenciája:

$$f_{\max} = \frac{c}{2d}$$

A mi esetünkben a maximális frekvencia a hangsebességet  $330 \frac{m}{s}$  értékűnek tekintve 1650 Hz-nek adódik, e fölött számolnunk kell az átlapolódás megjelenésével. (Forrás: [3].)



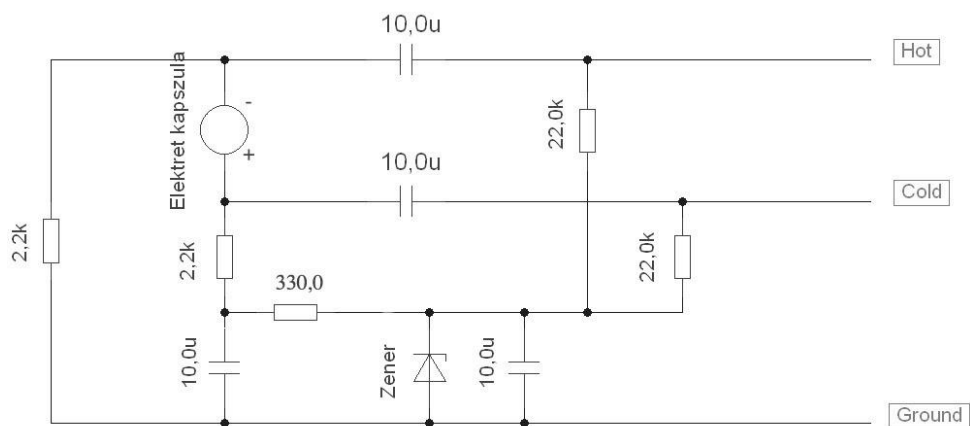


**3.2 ábra:** Mikrofonsor

A mikrofonok típusát illetően a választásunk az electret mikrofonokra esett, mivel a nagyon kedvező árak mellett megfelelően jó karakterisztikával rendelkeznek és elegendően kicsi méretben kaphatóak.

### 3.1.2 Előfeszítő áramkörök

Az electret mikrofonok a működésükhöz 2-12 V-os előfeszítést igényelnek. A rendelkezésünkre álló előerősítő csak 48 V-os fantom tápot tudott biztosítani, ezért szükség volt egy olyan áramkör összeállítására, mely képes külön-külön előfeszíteni a 8 darab mikrofont. Az áramköröket kézzel forrasztottam meg egy 200x100 mm<sup>2</sup> méretű próbapanelen az alábbi kapcsolási rajz szerint. A rajzon szereplő Zener dióda letörési feszültsége 12 V.



**3.3 ábra:** Előfeszítő áramkör (forrás: [7])

Az elkészített áramköröket bedobozoltam, a mikrofonok összeköttetését RCA kábelen, RCA csatlakozókkal oldottam meg. Az előerősítő és az előfeszítő áramkörök között egy egyedi XLR3-jack csatlakozást valósítottam meg. Az XLR bekötése szabványos, a jack csatlakozó csúcsa a Cold, a gyűrű a Hot, míg a köpenye a Gnd pólusoknak megfelelően lett bekötve. Az egyedi csatlakozó megválasztásánál az ár és az időtállóság volt a két figyelembe vett szempont.

Az összeállított előfeszítő áramkör mind a 8 csatornáját funkcionálisan teszteltem az éles laboratóriumi mérések előtt. A PC-s hangfelvételhez az Audacity ingyenesen letölthető hangrögzítő programot használtam.

## **3.2 Kísérleti mérések**

### **3.2.1 Mérési eredmények**

A létrehozott hardverrendszerrel több mérést végeztem. Ezek részletes tárgyalása az Önálló laboratórium beszámolómban olvasható. A jelenlegi dokumentum szempontjából azonban fontos megemlítenem a Híradástechnikai Tanszék süketszobájában végzett mérésből levonható következtetéseket. A mérési elrendezést és a mérés során a nyalábformálás segítségével készített hangtérképeket a **2. Melléklet** tartalmazza.

A süketszobás mérés eredményei arra engedtek következtetni, hogy az implementált nyalábformáló algoritmusom az előzetes szimulációknak megfelelő módon működik a használt lineáris mikrofonelrendezés esetén.

### **3.2.2 Mérőrendszer korlátai**

Bár a mérési eredmények megfeleltek az előzetes várakozásoknak, a mérőrendszer és a mérési elrendezés olyan korlátokat állítottak elénk mind a mérések időigénye, mind az eredmények kiértékelése és a nyalábformáló algoritmus hatékonysága terén, melyek megoldását mindenképp szükségesnek láttuk a konzulensemmel.

A mérések nagy időigénye a mérési elrendezés aránylag nagy összeállítási idejéből adódott. A 3.3. ábrán jól látszik, hogy minden mérés megkezdése előtt 24 darab, egyenként átlagosan 4 méter hosszú kábelt kellett a megfelelő aljzatokhoz csatlakoztatni. Mérések után pedig ugyanezt a rendszert kellett szétszedni.

Ennek megoldására elterveztük, hogy a következő mérőrendszernél minimálisra csökkentjük a mikrofontömb és a mintavevő kártya (jelenleg hangkártya) közötti vezeték- és kábelszámot.

A mérőrendszer másik két nagy hátránya a mikrofontömb kialakításának volt köszönhető. A lineáris elrendezésből adódóan a nyalábformálás kimenetétül szolgáló hangtérkép mindenképpen hengerszimmetrikus lesz, mert a mikrofonoktól egyenlő távolságra lévő pontok egy köríven helyezkednek el a mikrofonosor körül. A másik zavaró jelenség, mely szintén az elrendezés hibája, hogy az akusztikus nyalábformálás alsó frekvenciakorlátja 200 Hz körül van, így a nagyon alacsony frekvenciás hangforrások feldolgozásánál egyáltalán nem jellemző a várt térbeli szelektivitás. E két jelenség kiküszöbölésére a mikrofonelrendezés módosítása mellett döntöttünk, ami egyben a mikrofonok számának növelését is igényelte.

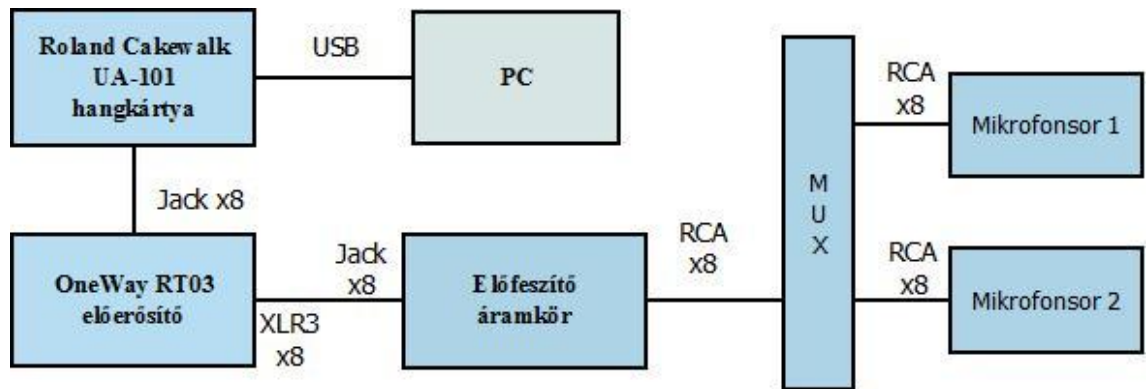
A mikrofonszám növeléséhez elengedhetetlen a hangfelvételi eszközt is felkészíteni több bemeneti csatorna kezelésére. Erre két lehetőséget vettünk számításba. Egyik ötletünk a már eddig is használt Roland Cakewalk UA-101 hangkártya bemenetén a mikrofonjelek multiplexelése volt, amellyel egy-egy előre meghatározott időablakban különböző mikrofonnyolcasok jelét lehet felvenni. A másik számításba vett megoldás a mikrofonszámmal párhuzamosan a bemeneti A/D átalakítók számának növelése volt. A tanszéken azonban ekkor még nem állt rendelkezésünkre 8 csatornánál több bemenetet kezelő mintavevő kártya, így először az első módszer megvalósíthatóságát ellenőriztem kísérletekkel, hogy megismerjem annak korlátait.

### **3.3 Analóg multiplexeres próbálkozás**

#### **3.3.1 Hardver-összeállítás**

A felvevő csatornák számának analóg multiplexerrel történő bővítéséhez a már rendelkezésre álló kísérleti rendszert egészítettem ki egy analóg kapcsoló fokozattal. Először 16 mikrofon jelének feldolgozását tűztem ki célul. Ehhez a kapcsoló áramköröket egy 50%-os kitöltési tényezőjű négyszögjellel célszerű vezérelni, így egységnyi idő első felében az egyik, második felében a másik 8 darab mikrofonból álló tömb jelét kapcsolja a hangkártya bemenetére a

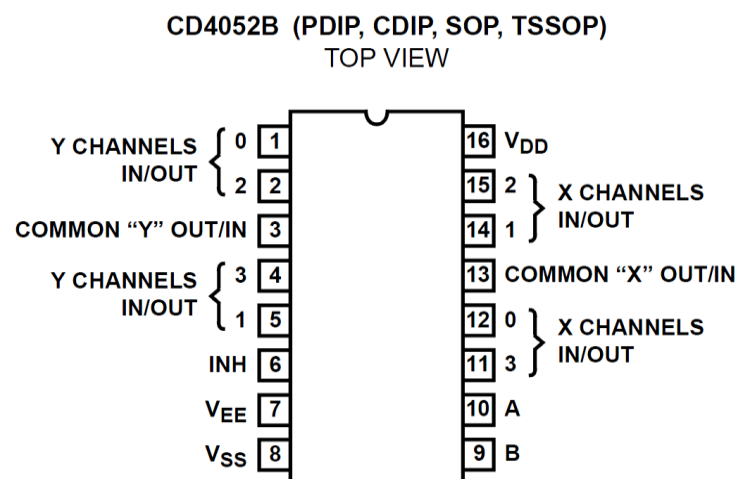
multiplexer áramkör. Az időablak megválasztásánál meghatározó szerepet játszik, a feltételezett fókuszban található hangforrás mennyi ideig tekinthető állandónak (stacionáriusnak). A tervezett hardver-összeállítást a következő ábra szemlélteti:



3.4 ábra: Analóg multiplexeres rendszer-összeállítás

### 3.3.2 Multiplexer áramkör

Az analóg multiplexer feladatra az adatlapján megadott paraméterek alapján a Texas Instruments CD4052B típusszámú integrált áramkört találtam megfelelőnek.



3.5 ábra: Texas Instruments CD4052B

Az IC képes az A és B vezérlőjelektől függően egy-egy közös analóg vonalra (Common X és Y) négy-négy csatorna (X0-3; Y0-3) jelét kapcsolni. Mivel az alkalmazásunk szempontjából az analóg MUX egy kritikus elem, az adatlapon szereplő kapcsolási paramétereket mérésekkel ellenőriztem.

A multiplexer adatai közül az alkalmazásunk leginkább az átkapcsolási késleltetésre (vezérlőjel váltás esetén a csatorna kimenetének átállási ideje) és a jelterjedési késleltetésre érzékeny. A worst-case tervezési elv alapján az átkapcsolási késleltetés várható legnagyobb értéke fontos számunkra, mert a kapcsolások során az ez idő alatt vett mintákat a későbbi feldolgozás során külön kell kezelni. A mért eredményeim alapján az átkapcsolási késleltetés minden esetben kisebb volt, mint az adatlapon feltüntetett legkedvezőtlenebb érték, azaz 720 ns. Csak ezt a paramétert figyelembe véve legfeljebb durván 1 MHz-es vezérlőjelet érdemes az IC kapcsoló lábára adni. Később látni fogjuk, hogy igazából nem ez a paraméter korlátozza a kapcsolási frekvenciát.

A jelterjedési késleltetés esetében a mért érték időbeli állandósága és a kapcsolt csatornáktól való függetlensége lényeges. Méréseim alapján a multiplexer IC minden csatornán kevesebb, mint 60 ns jelkésleltetést okozott, továbbá a kimeneti jelalakon nem figyeltem meg zavaró nemlineáris hatásokat. A választott integrált áramkört ezek alapján alkalmasnak találtam a mikrofonjelek multiplexelésére.

### 3.3.3 Multiplexeres elképzelés korlátai

Az analóg multiplexer áramkör bemérése után próbapanelen összeállítottam a 2 darab, egyenként 8 mikrofont tartalmazó tömb közötti átkapcsolásra alkalmas áramkör prototípust. A kapcsolójelet egyelőre függvénygenerátorral állítottam elő, a későbbiekben ennek szerepét egy, a hangkártya kimenetéről hajtott trigger áramkör vette volna át. Kulcsfontosságú szempont a kapcsolójelek egyenlő késleltetéssel való eljuttatása az összes (esetünkben még csak 2 darab) multiplexer IC-hez. Ez szimmetrikus jel utak használatával megvalósítható, de külön odafigyelést igényel.

Az áramkör prototípusának kapcsolt kimeneti jeleit a már ismert hangkártyával próbáltam felvenni. A hangkártya által felvett jelek nem egyeztek a hangkártya bemenetein megjelenő, oszcilloszkóppal ellenőrzött jelekkel. Ennek okát mérésekkel a hangkártyában lévő szigma-delta A/D átalakítók általunk előzetesen feltételezettnél nagyobb beállási késleltetésében azonosítottuk. A feltételezett 4-5 mintányi beállási késleltetés helyett az A/D átalakítók legrosszabb esetben akár 30 mintányi késést is okoztak a digitalizált jelben az eredetihez képest. Ennek következményeképp a hangkártya maximális mintavételi frekvenciája (96 kHz)

effektíve harmincadára csökkent ( $\sim 3$  kHz), így a rendelkezésre álló felvevő eszközünket a célunkra alkalmatlannak minősítettük. Tehát az analóg multiplexeres megoldáshoz más A/D átalakítóra van szükség.

## 4 Akusztikus nyalábformáló rendszer

Ebben a fejezetben ismertetem, hogy az előzetes kísérletekből leszűrt tapasztalatokból kiindulva milyen megfontolásokat vettem alapul a diplomatervem témájában szereplő akusztikus nyalábformáló rendszer tervezése során. Részletesen leírom, hogy a kísérleti tapasztalatokat felhasználva kigondolt előzetes rendszertervtől milyen tervezői és implementációs döntések vezettek a 15 bemeneti csatornát kezelő mintavevő hardverig.

### 4.1 Rendszerszintű terv

A diplomakiírásom második pontja alapján terveznem kellett egy 16 mikrofon befogadására alkalmas tömböt, és egy, a mikrofonjelek felvételére szolgáló jelkondicionáló és mintavételező egységet. A rendszer tervezése során az előző fejezetben tárgyalt kísérleti összeállításból indultam ki.

A 3.2.2 alfejezetben részletezett előzetes kísérleti tapasztalatok alapján a tervezés első fázisában az alábbi három célt tűztem ki:

- A nyalábformálás irányérzékenységének növelése.
- A mikrofon-mintavevő egység közötti jelúton megjelenő additív zajok csökkentése.
- Mikrofontömb robusztussága, mérések során a mérőrendszer gyors össze- és szétszerelhetősége.

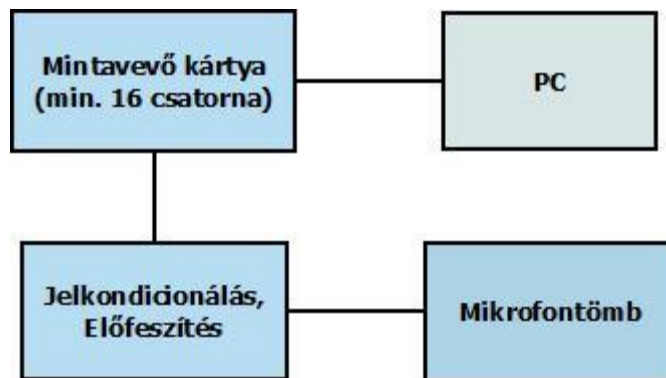
A *delay-and-sum* nyalábformáló algoritmus irányérzékenységét a [3] irodalom alapján a mikrofonszám növelésével és a mikrofonelrendezés optimális kiválasztásával lehet javítani. Előbbi egy univerzális módja a felbontás finomításának, míg az utóbbi módszer hatékonysága az éppen aktuális felhasználási igény függvénye. Ezek alapján az előzetes kísérletekhez képest egy 16 mikrofonból álló tömb alkalmazása várhatóan javítja majd az algoritmus irányérzékenységét.

A mikrofontömb robusztusságának növelése és az összeszerelési idő csökkentése elérhető a jelvezetékek és a csatlakozók számának minimalizálásával.

Tehát a cél egy kompakt, a lehető legkevesebb egységből álló hardver-összeállítás tervezése és implementálása.

A harmadik megjelölt cél az additív zajok csökkentése, melyet érdemes már a tervezés legelején szem előtt tartani, és minden implementáció-kiértékelési fázisban megvizsgálni.

A kezdeti célkitűzéseimet figyelembe véve összeállítottam egy rendszerszintű tervet, melynek feladata a 15 mikrofon jelét digitalizálni, és így a későbbi személyi számítógépes feldolgozás bemenetét szolgáltatni. A tervezett rendszer egy 15 elemű mikrofontömbből, egy előfeszítő és jelkondicionáló áramkörből, egy legalább 16 darab bemeneti csatornát kezelő A/D egységből és a felvételt irányító PC-ből állt. Ezt szemlélteti a következő ábra:



4.1 ábra: Rendszerszintű terv vázlata

## 4.2 Mintavevő kártyák

### 4.2.1 Mintavevő eszközök

Az előzetes kísérletek alapján nyilvánvalóvá vált, hogy 15-16 bemeneti csatorna egyszerre történő mintavételezése nem triviális feladat. Mint már a 3. fejezetben részletesen tárgyaltam, az előzetes kísérletekben használt, 8 csatornás hangkártya még analóg multiplexer kiegészítéssel is alkalmatlan 16 csatorna kezelésére.

A hangkártyán kívül még rendelkezésemre állt a DSP laborban egy elég régi, National Instruments DAQ-6062 típusú mintavevő kártya, mely egyszerre 16 analóg bemenet digitalizálására képes.



	<b>Hangkártya</b>	<b>Mintavevő kártya</b>	<b>Mintavevő kártya</b>
<b>Típus</b>	Roland UA-101	NI DAQ-6062	NI USB-6363
<b>Csatolófelület</b>	USB	PCMCIA	USB
<b>Csatornaszám</b>	8	16	32
<b>Mintavételi frekvencia</b>	96 kHz (csatornánként)	500 kS/s (aggregált)	max 2000 kS/s (aggregált)
<b>Felbontás</b>	24 bit	12 bit	16 bit
<b>A/D típus</b>	Szigma-Delta	Szukcesszív approximációs	Szukcesszív approximációs

**4.1. táblázat: A hangkártya és a mintavevő kártyák összehasonlítása ([12], [13] és [14] alapján)**

A táblázatban megjelenő National Instruments USB-6363 kártya egy később jelentkező igényt kiszolgáló alternatívája a DAQ-6062 mintavevő egységnek.

Az adatai alapján a DAQ-6062 mintavevő kártya alkalmas a hangkártya leváltására 16 bemeneti csatorna esetén. Az aggregált 500 kS/s mintavétel maximális csatornaszám esetén csatornánként 31.25 kHz-es mintavételi frekvenciát jelent, ami a mi céljainkra elegendő. A PCMCIA csatolófelület valamennyire korlátoz, hisz ez általában csak a régebbi típusú laptopokon van jelen. A 12 bites felbontásba egyelőre bele kellett nyugodnunk. Az első technikai értékelés után úgy döntöttem, hogy érdemes a fenti mintavevő kártya pontos paramétereinek megismerésére méréseket végezni.

#### 4.2.2 A DAQ-6062 kártya bemérése

A mérőkártya meghajtó programja biztosít a Matlab felé egy magas szintű kommunikációs protokollt, mely választhatóan lehet lekérés-válasz és munkafolyamat jellegű.

Én az egyszerűsége miatt az előbbi kommunikációs módot használtam, mivel a felhasználás szempontjából nem járt volna nélkülözhetetlen előnnyel a kommunikáció munkafolyamatba szervezése. Egy bemeneti csatorna konfigurációját és arról adatok beolvasását a következő Matlab kóddal valósítottam meg:

```
ai = analoginput('nidaq', 'Dev1');

chan = addchannel(ai, 0, '0');
set(ai, 'InputType', 'SingleEnded');
Fs = setverify(ai, 'SampleRate', 16000);
N = setverify(ai, 'SamplesPerTrigger', 16000);

set(chan, 'SensorRange', [-0.5 0.5]);
set(chan, 'InputRange', [-0.5 0.5]);
set(chan, 'UnitsRange', [-0.5 0.5]);
set(chan, 'Units', 'V');

start(ai);
wait(ai, 2*N/Fs);
data = getdata(ai);
stop(ai);
delete(ai);
clear ai;
```

Mint a programkódból egyértelműen látszik, a mintavevő kártya meghajtó programja lehetőséget biztosít a mintavételi frekvencia, a méréshatárok és az egy mérés során rögzítendő minták számának beállítására. A mintavevő kártya két mérési módot támogat: a földfüggő, úgynevezett *SingleEnded*; és a földfüggetlen, *Differential* módot. Utóbbinál egy jel mérése a mintavevő egység két bemenetét igényli, így a rendelkezésre álló kártyával csak 8 bemenetet tudtam volna differenciálisan kezelni, ezért a *SingleEnded* módot kellett választanom.

A DAQ-6062 mérési adatgyűjtő kártya specifikációjában szereplő paraméterek kielégítő mértékben megfeleltek a valóságnak, azonban az adatlap nem tartalmazott két, számomra alapvető fontosságú információt:

- A különböző csatornákon elméletileg egyszerre történő mintavételek között tapasztalt valós időkülönbség.
- A mintavételi késleltetés átlagos értéke, jittere és e két paraméter homogenitása a különböző csatornákon.

Célirányos mérések segítségével - ugyanazokat a jeleket kötve a bemenetekre - megfelelő biztonsággal állíthatom, hogy a mintavevő kártya kvázi azonos időpontban tárolja el mindegyik bemeneti csatorna mintáit, majd egy csatornaszámtól függő idő elteltével a felhasználói program rendelkezésére bocsátja azokat. Ez a működés számomra megfelelő, hisz így a csatornák mintái körülbelül ugyanahhoz az időpillanathoz köthetők, nem kell rajtuk külön-külön időeltolást végezni.

Külön méréseket végeztem annak meghatározására, hogy az ismételt mintalekérések között mennyi idő elteltével szolgáltat a mintavevő kártya adatokat. Tehát ha egymás után többször veszünk N darab mintát, azok mennyire tekinthetők folyamatosnak. Tapasztalatom alapján ez a mintacsomagok közötti késleltetés nem állandó, a mintavétel során használt operációs rendszer ütemezőjétől függ. Ez a tulajdonság a későbbi valós idejű feldolgozást megnehezíti, de a Matlab keretrendszerét (Data Acquisition Toolbox) megkerülve, közvetlenül a National Instruments MAX szoftverén keresztül mintát véve valószínűleg megfelelő mértékben csökkenthető ez a hatás.

#### 4.2.3 Az USB-6363 típusszámú A/D kártya

A későbbiek során National Instruments külön kérésünkre kölcsönzött nekünk egy komolyabb, USB-6363 típusjelzésű mintavevő kártyát, amely legfeljebb 2000 kS/s mintavételre képes, 16 bites felbontással és 32 bemeneti csatornával rendelkezik. Ezzel az új kártyával sokkal jobb minőségű felvételeket tudtam készíteni és lehetőségem nyílt a későbbiekben differenciális módban kezelni a 16 bemeneti csatornát.

Nem elhanyagolható az sem, hogy az új kártya USB interfésszel rendelkezik a PCMCIA helyett, így univerzálisan csatlakoztatható bármilyen személyi számítógéphez.

Az előbbieken ismertetett méréseket az USB-6363 típusú kártyával is megismételtem, és a várt eredményeket tapasztaltam. A továbbiakban csak ezzel az új mintavevő egységgel végzett méréseket fogom tárgyalni annak ellenére, hogy nagy részüket elvégeztem még a régi, DAQ-6062 kártyával is.

### **4.3 Mikrofontömb bővítése**

A 4.1 fejezetben megfogalmazott, nyálábformáló algoritmus érzékenység növelési célkitűzés elérése érdekében a felvevő tömb mikrofonszámának növelése mellett döntöttem, mivel a [3] irodalom alapján ez már önmagában is javítja a hangtérkép felbontását, és egyben többféle mikrofonelrendezést tesz megvalósíthatóvá. Az új, 15 mikrofont tartalmazó tömb tervezése során hasznosnak bizonyultak a régi, 8 elemű tömb használatakor felmerült problémákból levont tapasztalatok, továbbfejlesztési igények. Ezek többsége gyakorlati jellegű, és a mikrofontömb mechanikai kialakítását érinti, de funkcionális szempontból nem elhanyagolhatók. A mikrofontömb kialakításának tervezésekor szem előtt tartottam a mikrofontömbből, előerősítő- és jelkondicionáló fokozatból és a mintavevő kártyából álló hardverrendszer későbbi egyszerű, gyors összeszerelhetőségét.

#### **4.3.1 Jeltovábbítás, vezetékezés**

A mikrofonjelek digitalizálására rendelkezésünkre álló DAQ-6068 mintavevő kártya összesen 16 bemeneti csatornát támogat. Ezzel lehetőséget biztosít 16 jel közös analóg referenciaponthoz viszonyított, vagy 8 jel differenciális mintavételezésére. Bár számunkra utóbbi működési mód sokkal kedvezőbb lett volna - mivel érzéketlen a vezetékeken indukált zavarokra -, a csatornaszámból adódó korlát miatt a másik, közös referenciapontú mintavételezést kellett használnom. Így 15 darab mikrofon kezeléséhez, közös táp- és referenciavezeték használva, legalább 17 vezeték szükséges.

A 3. fejezetben tárgyaltak alapján a Diplomatervem kiindulási alapjául szolgáló, 8 felvételi csatornát támogató hardverrendszer a mikrofontömb előfeszítésére használt megoldás és a differenciális jelátvitel miatt sok kábelt tartalmazott, ezért a rendszer aránylag sok helyet foglalt és összeállítása időigényes volt.



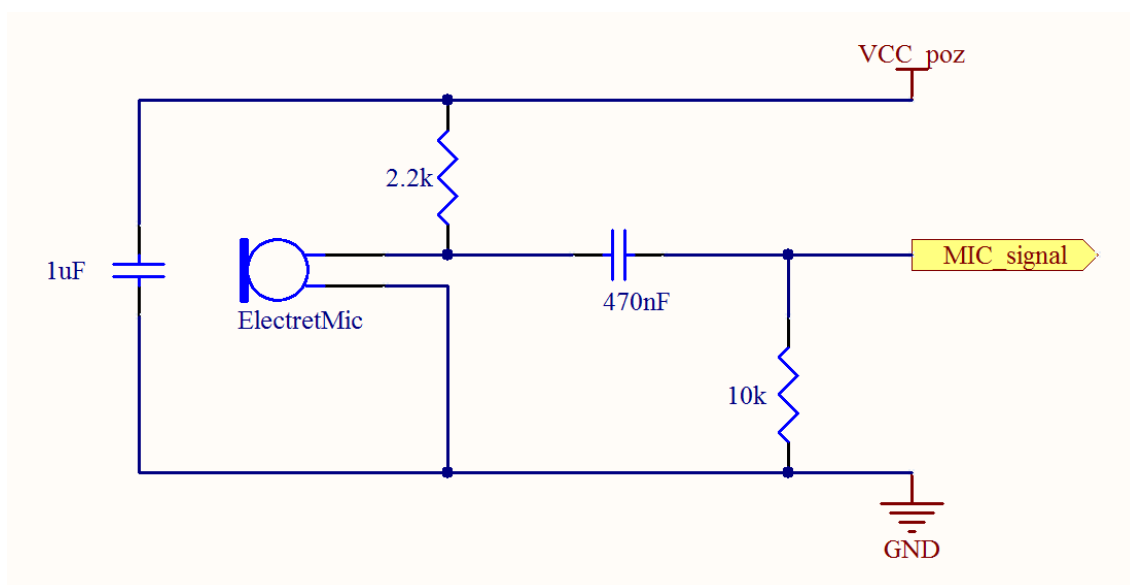
**4.2 ábra:** A korábbi mérési összeállítás

A 15 mikrofonból álló tömb kialakításánál törekedtem arra, hogy a lehető legkevesebb kábelt használjam. A differenciális jelátvitel hiánya miatt a jelvezetékek érzékenyebbek a zajokra, ezt a hatást a vezetékek árnyékolásával lehet csökkenteni. A választásom végül az Cat 5e fajtájú FTP (Foil-shielded Twisted Pair) kábelre esett. Ez a kábeltípus 8 darab, páronként sodort, réz eret és egy földet tartalmaz, melyek kívülről egy alumínium fóliával vannak árnyékolva. Ezen felül a sodrások térbeli periodicitásának változtatásával a vezetékek között minimális áthallást értek el, mely a mi alkalmazásunk szempontjából is kedvező.

A mikrofontömb funkcionalitásához szükséges 15 jel, 1 táp és 1 referencia vezetékhez így 2 darab Cat 5e FTP kábelt használtam fel. A vezetékek mikrofon felőli végét fixen hozzáferrasztottam a csatlakozási pontokhoz, a jelkondicionáló felőli végükön pedig egy közös, 18 pólusú Mini-fit csatlakozót helyeztem el.

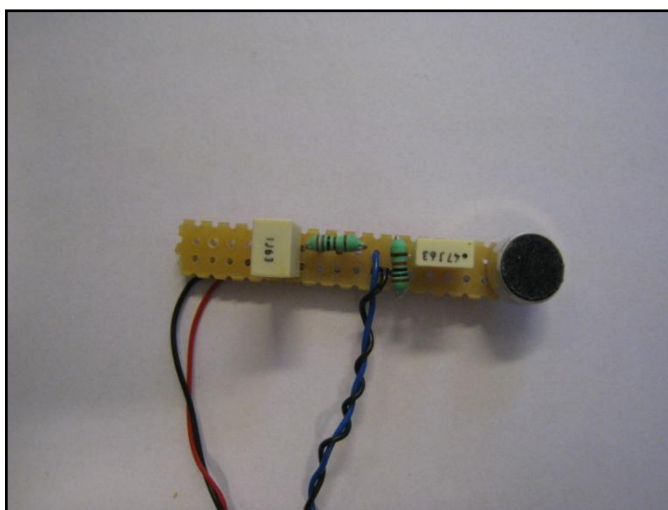
#### 4.3.2 Lokális előfeszítő áramkör

A korábbi, 8 mikrofonos rendszer közös, nagy előfeszítő áramkörével szemben az új tömbben a mikrofonok előfeszítő áramkörét közvetlen az electret kapszulák mellett alakítottam ki. Ezzel csökkentettem a hely- és vezetékezési igényt és a tápláláson keresztül csatolt zajokat is. Az előfeszítő áramkör kapcsolási rajza az alábbi ábrán látható:



**4.3 ábra:** A lokális előfeszítő áramkör kapcsolási rajza (forrás [7])

A kapszulák megfelelő érzékenységének eléréséhez a [7] irodalom 3 V és 12 V közötti előfeszítő feszültséget javasol. Mivel mind a DAQ-6068, mint az USB-6363 mintavevő kártya adatlap szerint rendelkezik 5 V-os, stabilizált, analóg kimenettel, ezért kézenfekvő volt ezt használni a mikrofonok táplálásához. A későbbiekben a mérések során kiderült, hogy a mérőkártyák által kiadott 5 V rendkívül zajos, így a végső összeállításban a mikrofonok tápfeszültségét a jelkondicionáló áramkör stabil, 5 V-os, pozitív tápvonaláról nyertem.

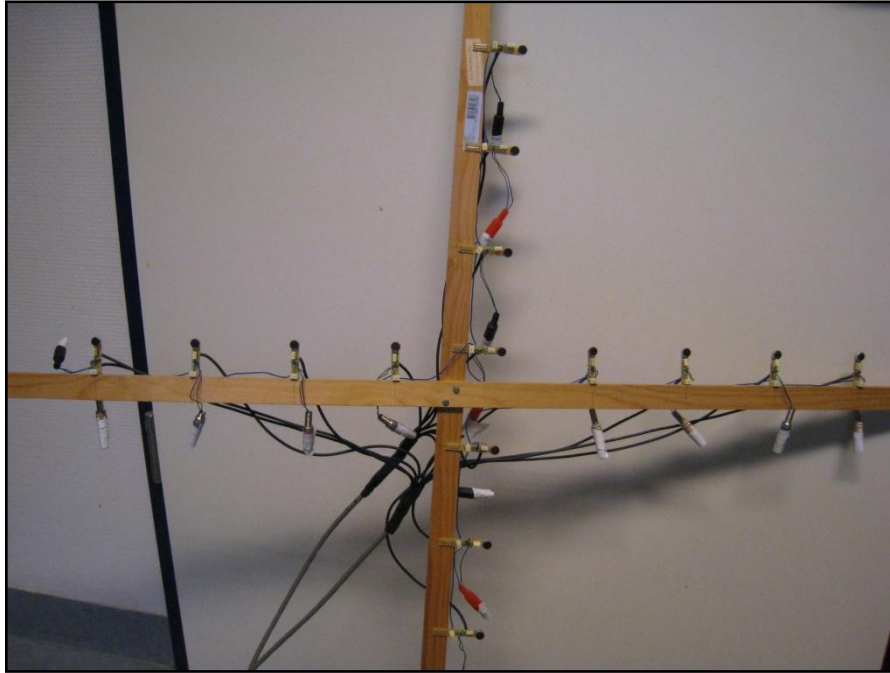


**4.4 ábra:** A mikrofon és az előfeszítő áramkör

### 4.3.3 Mikrofonelrendezés

A 2.2 fejezetben tárgyalt, gyakorlati életben is alkalmazott mikrofonelrendezések közül első megközelítésben a kereszt alakzatot választottam, mivel ez egyszerűen összeállítható, a mikrofonok helye könnyen meghatározható, és a későbbiekben a régebbi, lineáris elhelyezéssel könnyen összehasonlítható az ezzel kapott eredmény.

A kereszt elrendezés megvalósításához vízszintesen 9, függőlegesen pedig 7 mikrofont rögzítettem két, középen összeerősített fenyőfa léchez. A középső (horizontálisan az ötödik, vertikálisan a negyedik) mikrofon a kialakításban közös, így valóban 15 mikrofonból áll a létrehozott hangrögzítő rendszer, melyet az alábbi ábra szemléltet:



**4.5 ábra:** A 15 elemű, kereszt elrendezésű tömb

## **4.4 Jelkondicionáló fokozat**

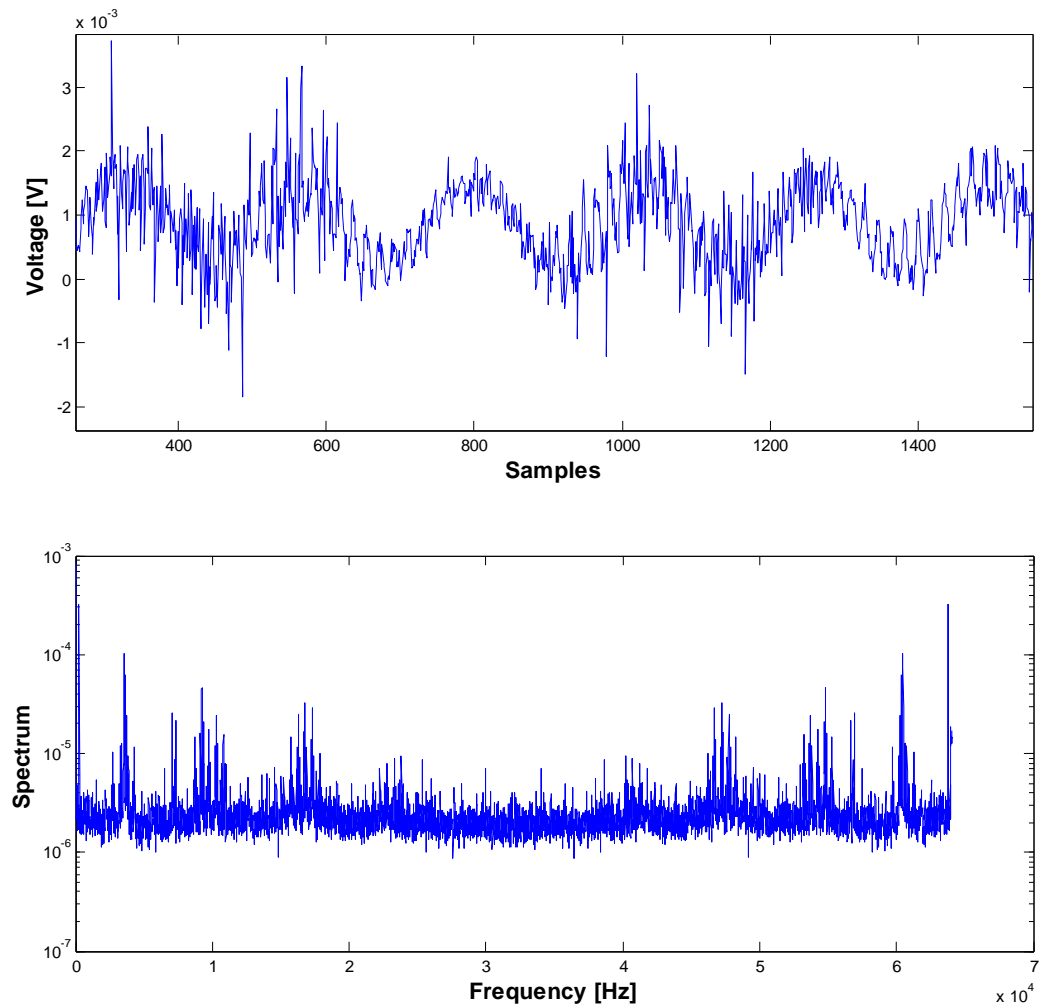
A mérési adatgyűjtő kártya bemérése és a megtervezett mikrofontömb összeállítása után megpróbáltam célirányosan a mikrofonok jelét rögzíteni a mintavevő eszközzel. Több, előre nem várt akadályba ütköztem. Egyrészt az először használt DAQ-6062 mintavevő kártya beépített erősítője és 12 bites felbontása nem bizonyult elegendőnek a 4-5 mV amplitúdó tartományba eső mikrofonjelek megfelelő felerősítéséhez. Másrészt a mikrofonokat harmonikus hangokkal gerjesztve az A/D átalakítókhoz érkező jeleken a harmonikus gerjesztéssel egy nagyságrendbe eső zaj volt megfigyelhető. E két jelenség kiküszöbölése érdekében először méréseket végeztem a problémák pontos megismeréséhez, majd terveztem és implementáltam egy jelkondicionáló fokozatot.

### **4.4.1 Szűrők és erősítés kiválasztása**

A mikrofonjelek alacsony jelszintjéből fakadó probléma megoldása valamennyire kézenfekvő, be kell építeni egy előerősítő fokot a mintavevő kártya elé. A mérési tapasztalatok alapján csatornánként százszoros előerősítésre van szükség, így a felerősített jelek kényelmesen feldolgozhatók az A/D átalakító -1/+1V mérési tartományában.



A zajok eredetének, frekvenciáinak és teljesítményének becslésére méréseket végeztem: a mikrofonokat különböző harmonikus frekvenciájú hangokkal gerjesztettem, a jeleket pedig közvetlenül kötöttem az A/D átalakító bemenetére. A 275 Hz-es gerjesztéssel kapott mérési eredményeket az alábbi ábrák szemléltetik.

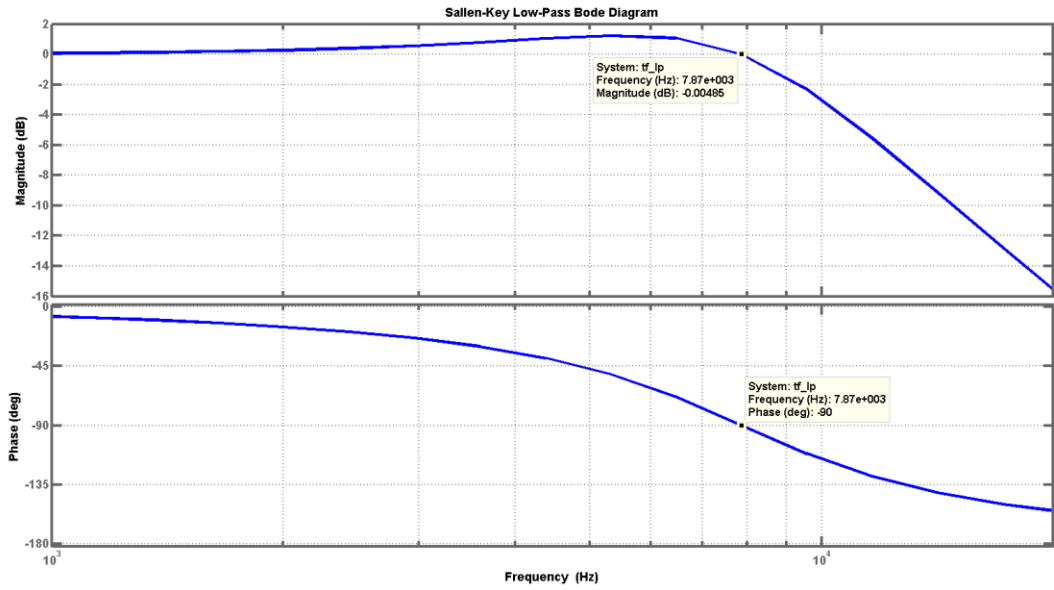


**4.6 ábra:** 275 Hz-cel gerjesztett mikrofon digitalizált jele az idő és a frekvencia tartományban

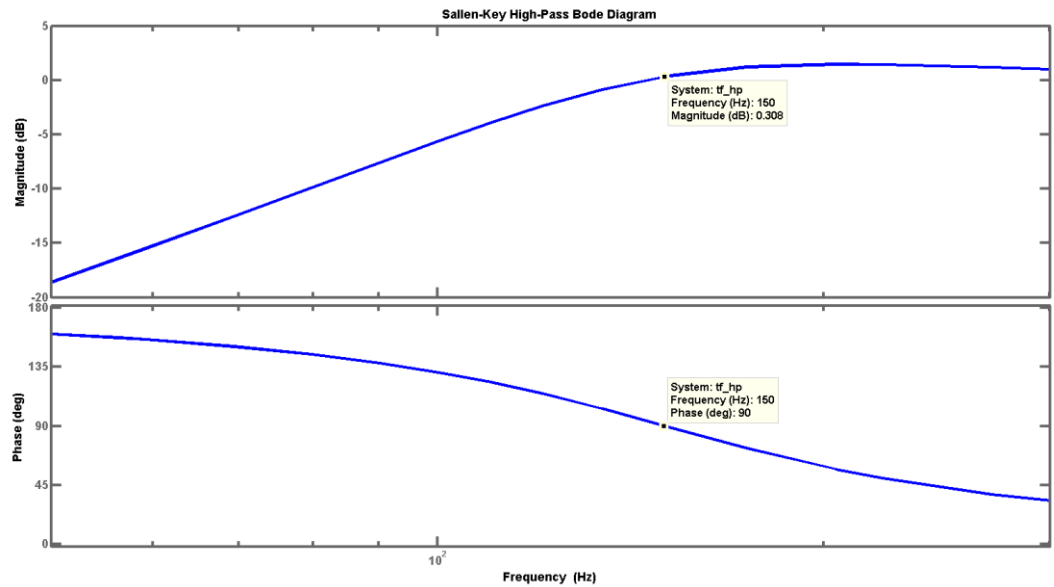


**4.7 ábra:** A mikrofonok bemérése

A mérési tapasztalatok alapján megállapítottam, hogy egy 150 Hz törésponti frekvenciájú felüláteresztő és egy 8 kHz törésponti frekvenciájú aluláteresztő, másodfokú szűrővel elegendő zajelnyomást tudok elérni a specifikáción kívül eső frekvenciatartományokban. A másodfokú szűrőket két Sallen-Key alappal valósítottam meg. A felüláteresztő tag csillapítására a 2-es, míg az aluláteresztő tag csillapítására 0.5-ös értéket választottam. A szűrők átviteli függvényeit az alábbi két Bode diagram szemlélteti:

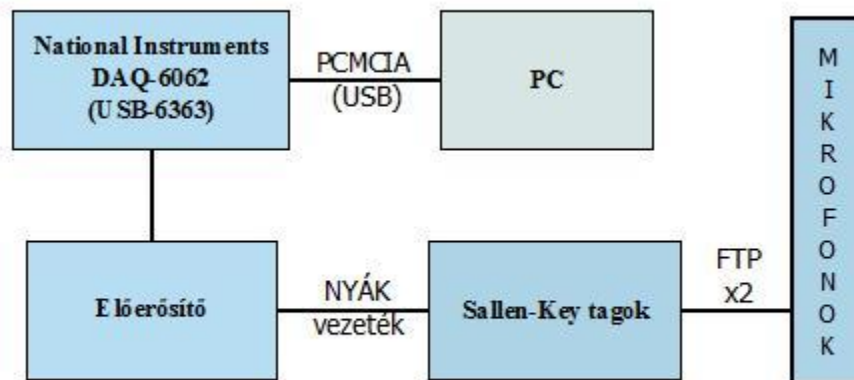


**4.8 ábra:** Aluláteresztő Sallen-Key tag átvitele



**4.9 ábra:** Felüláteresztő Sallen-Key tag átvitele

A fentieknek megfelelően a mintagyűjtő kártya elé, a bemeneti csatornákra terveztem egy előerősítő és egy szűrő fokozatot. A hardverrendszer blokkvázlatát és a részegységek közötti kapcsolatot a következő ábra szemlélteti



**4.10 ábra:** Módosított hardverrendszer

Az előerősítő és jelkondicionáló fokozat egy csatornát támogató prototípusát megépítettem, és az előzetesen elvárt viselkedést tapasztaltam. A következő lépés egy 16 csatornát támogató változat nyomtatott áramkörének megtervezése volt.

#### 4.4.2 Áramkörtervező-program kiválasztása

A jelkondicionáló fokozat nyomtatott áramkörét mindenképp egy CAD környezetben szerettem volna megtervezni, mivel az egyetemi tanulmányaim során nem oldottam meg még ilyen igénylő önálló feladatot, és véleményem szerint egy beágyazott fejlesztő számára alapvető elvárás a nyomtatott áramkörök tervezésében szerzett alapszintű tapasztalat.

A nyomtatott áramkör kapcsolási rajzának és az áramkör rajzolatának tervezésére alkalmas programok közül az *Altium Designer 2009 Winter*-re esett a választásom. Az Altium Designer segítségével lehet kapcsolási rajzot tervezni, a szükséges könyvtárak beszerzése után az összeállítás viselkedését szimulálni, nyomtatott áramköri rajzolatot kialakítani és footprint-et szerkeszteni. A program ellenőrizni tudja a nyomtatott áramkör és a kapcsolási rajz közötti konzisztenciát és az előre definiált gyártástechnológiai paramétereknek való megfelelést, továbbá képes a gyártáshoz szükséges leíró fájlok (ú.n. gerber fájlok) előállítására is. A program használatát a gyártója által nyilvánosságra hozott oktatási anyagokból ismertem meg ([9] irodalom).

A tervezés során nagy segítségemre volt a *multi-channel design* funkció, mely több, azonos kapcsolási rajzú komponens uniform kezelését teszi lehetővé. A kapcsolási rajzon és az áramköri rajzolatban történő módosítások egyszerre érvényesülnek az összes komponensen. A *multi-channel design* használatával nagyban csökkenteni lehet a redundanciát, és minimalizálni lehet az egyes változtatásokhoz szükséges időt. A multi-channel design használatával nem kellett minden bemeneti csatorna jelkondicionáló fokozatát külön-külön, egyesével implementálnom. Egyetlen hátrányként azt hoznám fel, hogy sajnos a fejlesztőkörnyezet általam használt verziója nem támogatta rendesen a többszintű implementációját.

#### 4.4.3 Áramkörtervezés menete

A nyomtatott áramkör tervezését az áramkör kapcsolási rajzának összeállításával kezdtem. A jelkondicionáló megvalósításához szükséges funkciókat – előerősítés, szűrés és tápellátás - elkülönítve, modulárisan valósítottam meg, kihasználva a fejlesztőkörnyezet által nyújtott támogatást. Az egyes modulok megfelelő ki- és bemeneti vezetékeit egy föléjük rendelt tervrajz lapon kötöttem össze egymással és a külvilággal kapcsolatot biztosító csatlakozókkal.

Az kapcsolási rajz után a beépítendő alkatrészek kiválasztása következett. A szűrő és előerősítő moduloknál a lehető legkisebb eltéréssel próbáltam lemásolni SMD alkatrészekkel az előzetes tesztelésre használt prototípus áramkör furatszerelt alkotóelemeit, hogy a próba panelen létrehozott prototípus működését a lehető legpontosabban megközelítsem. Így ebben nem sok tere volt már a kreativitásnak. Azokból az elemekből gazdálkodtam, melyek furatszerelt megfelelője korábban megtalálható volt a Méréstechnika Tanszék önálló laboratóriumában. A tápellátás terén a kezem nem volt ennyire megkötve, pusztán az erősítők előzetesen alkalmazott szimmetrikus feszültségéhez, a mikrofon előfeszítések feszültség szintjéhez és mindezek áramigényéhez kellett igazodnom.

Miután eldöntöttem, milyen alkatrészeket kívánok beépíteni a jelkondicionáló fokozatba, egyenként meg kellett rajzolnom a footprint-üket az áramkörrajzolat kialakítása előtt. Ebben segítségemre volt az Altium Designer footprint és komponens varázslója, valamint footprint könyvtárkezelője.

Ezután következett a nyomtatott áramkörön az alkatrészek elhelyezése a footprint-ek segítségével, majd a megfelelő összeköttetések kialakítása vezetékvezéssel. Itt már figyelembe kellett vennem a gyártó cég által közölt technológiai korlátokat is, például a minimális furatvastagságot, minimális vezeték szélességet és az elemek közötti minimális távolságot. A választott CAD program mindehhez komoly támogatást nyújtott.

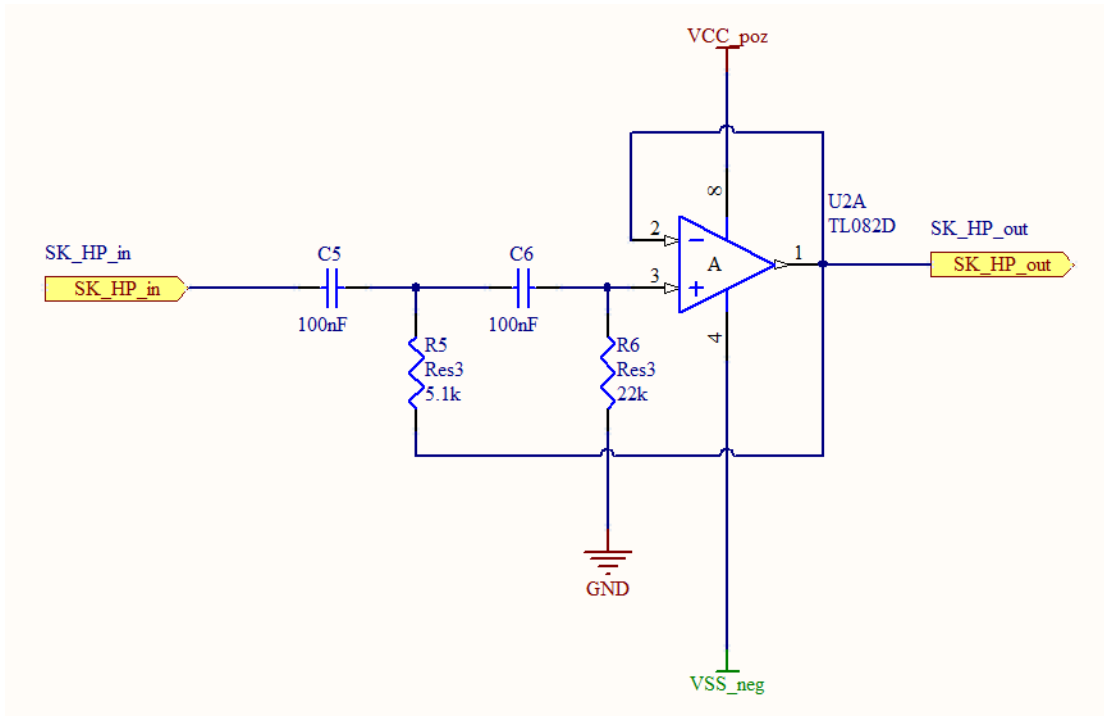
A nyomtatott áramkör megtervezése után leadtam a gerberfájlokat gyártásra, majd beszereztem a szükséges alkatrészeket. Az elkészült nyomtatott áramkörre többszöri makacs próbálkozással ráégettem három ujjbegyemet, közben beültettem az alkatrészeket, majd bemértem a jelkondicionáló áramkört, és kijavítottam a gyártási és tervezési hibákat.

#### 4.4.4 Sallen-Key tagok, előerősítő fok

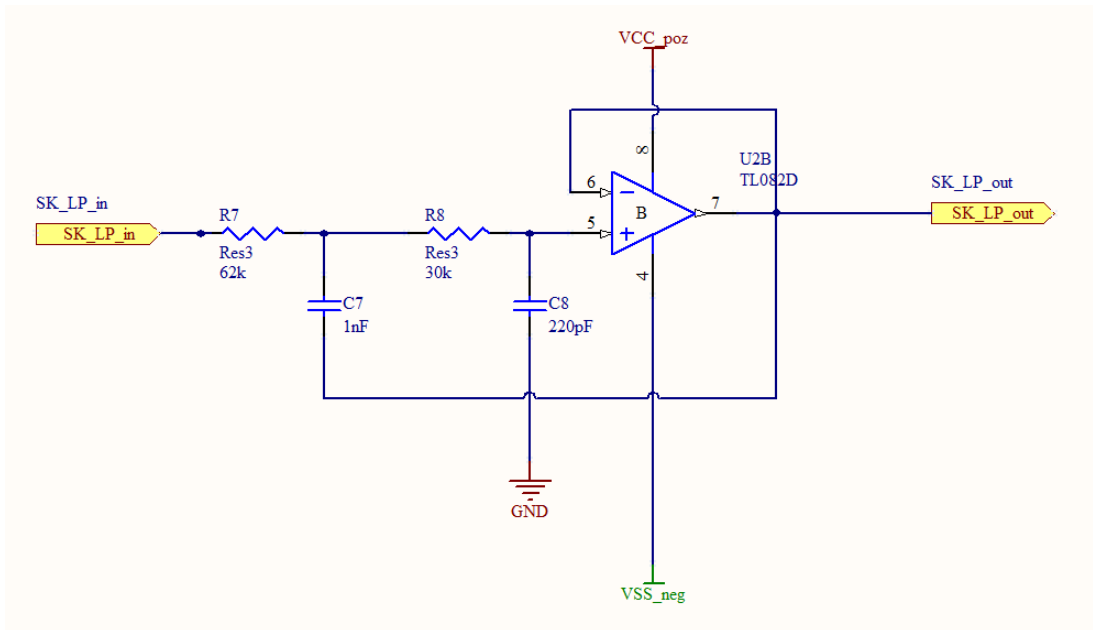
Mivel mind a 16 bemeneti csatornához külön-külön jelkondicionáló fokozatokra van szükség, a tervezés időszükségletének csökkentése érdekében a Sallen-Key tagok és az előerősítő fok tervezéséhez az Altium Designer előző fejezetben bemutatott, *multi-channel design* funkcióját használtam. Bár ez a megközelítés önmagában lehetővé tenné, hogy egyetlen csatorna tervéből levezessem mind a 16 csatornát, ez a gyakorlatban harmadával megnöveli a szükséges műveleti erősítő integrált áramkörök számát. Ennek oka az, hogy a prototípusban a TL082 típusú műveleti erősítő IC-t használtam, amely egy tokban két műveleti erősítőt tartalmaz. Egy csatornához három műveleti erősítőre van szükség, így érdemesnek találtam két csatornát összevonni a tervben, így négy helyett három darab integrált áramkör elegendő ugyanahhoz a funkcióhoz.

Az egyes csatornákhöz tartozó két Sallen-Key tagot azonos tokban lévő erősítőkhöz rendeltem. A két különböző csatorna erősítő fokának megvalósításához közös tokban lévő műveleti erősítőket használtam. Az integrált áramkör csatornái közötti áthallás elhanyagolható mivoltát mérésekkel bizonyítottam

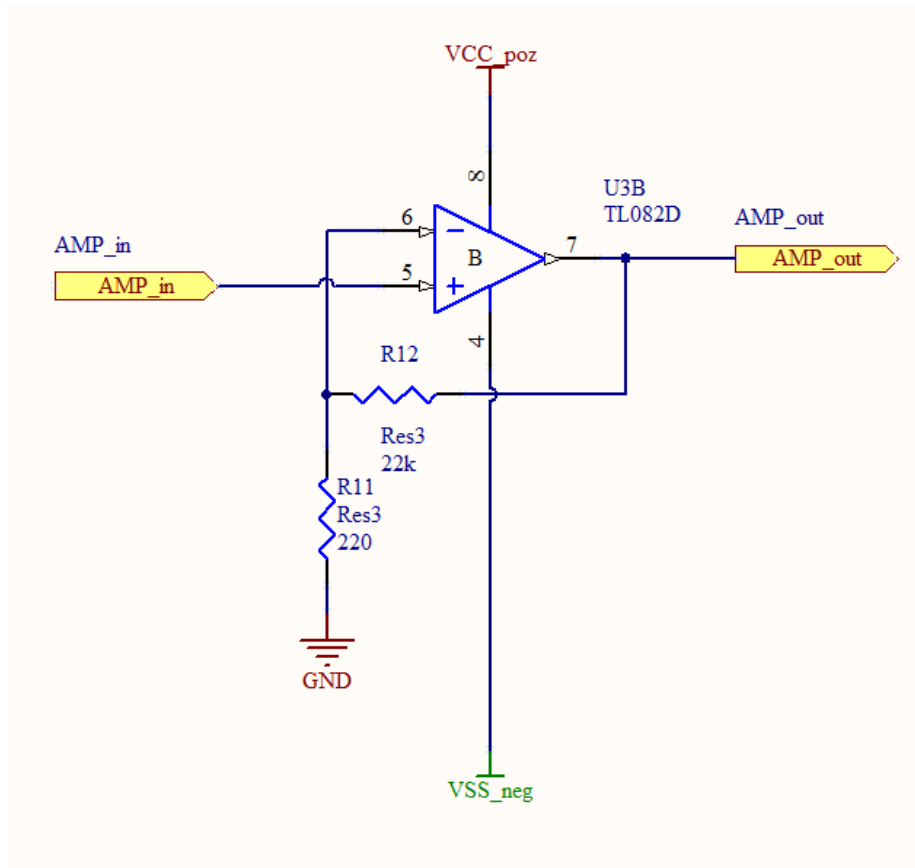
A 4.4.1 fejezetben részletezett karakterisztikákat megvalósító Sallen-Key tagok kapcsolási rajzát az alábbi két ábra szemlélteti. Utánuk a százszoros erősítést megvalósító nem-invertáló fokozat rajza található.



**4.11 ábra:** Felüláteresztő Sallen-Key tag ( $f_c \sim 150$  Hz)



**4.12 ábra:** Aluláteresztő Sallen-Key tag ( $f_c \sim 8$  kHz)



**4.13 ábra:** Erősítő fokozat kapcsolása

A kapcsolás realizálásához felhasznált minden ellenállás 1%-os tűréssel rendelkezik. A kondenzátorok típusválasztásakor a frekvenciamenet szempontjából legkedvezőbb kerámia kondenzátorokra esett a választásom. Az egységárat is figyelembe véve, 5%-os tűrésű kondenzátorokat építettem a kapcsolásba. A műveleti erősítőket a Texas Instruments TL082CDG4 ([15]) integrált áramkörét felhasználva valósítottam meg. Az erősítő IC-k pozitív és negatív tápfeszültségét egy-egy 100 nF-os, 10%-os tűrésű kondenzátorral hidegítettem.

#### 4.4.5 Tápellátás, vezetékezés

Az erősítőket +5 és -5 V tápfeszültséggel láttam el. A pozitív és a negatív ág maximális áramfelvétele az előzetes mérések alapján egyenként 180 mA-re becsülhető.

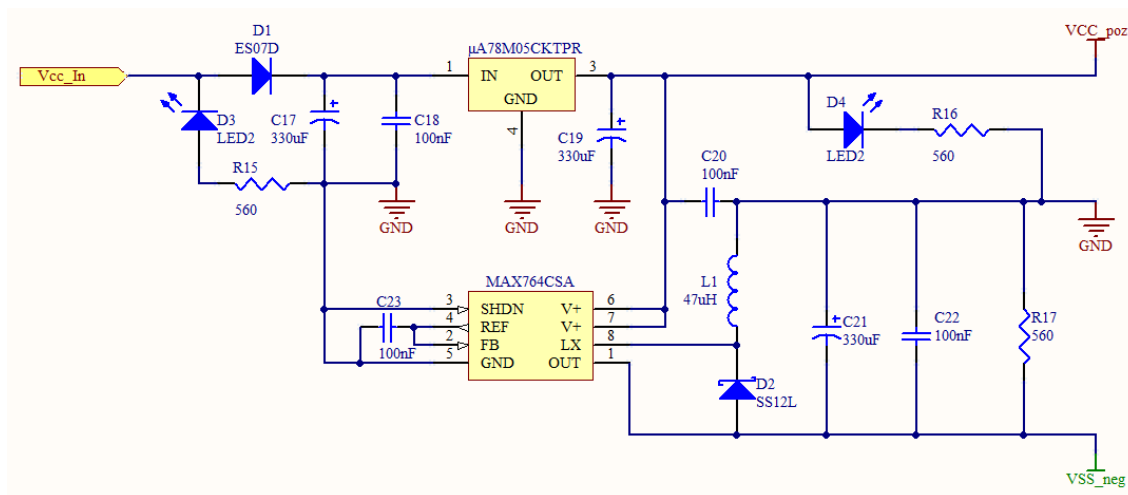


Mivel a negatív tápfeszültség közvetlenül a pozitívból van előállítva, így a negatív táp biztosítására elegendő egy 250 mA terhelhetőségű DC/DC konverter, míg a pozitív tápfeszültség stabilizálására egy 500 mA maximális áramú, 7805-ös típusszámú stabilizáló IC-t kellett választanom.

A hálózati táplálásra használt kapcsolóüzemű AC/DC konverter 7.5 V-os kimeneti feszültségéből, a Texas Instruments MA78M05CKC típusú ([16]) integrált áramkörének segítségével állítottam elő a stabilizált pozitív tápfeszültséget (+5 V).

A pozitív tápfeszültségből egy Maxim MAX764CSA típusjelű ([18]), kapcsolóüzemű DC/DC átalakítót felhasználva hoztam létre a negatív tápfeszültséget (-5 V). A DC/DC konverter kiválasztásánál ügyeltem arra, hogy az a hangfrekvenciás tartományban ne keltsen jelentős zajt.

A két stabil tápfeszültséget szolgáltató IC megfelelő lábain elhelyeztem egyegy elektrolit puffer és egyegy kerámia hidegítő kondenzátort is. A tápcsatlakozó után beépítettem egy kis nyitófeszültségű védődiodát az ellentétes polaritás elleni védelem érdekében. A tápellátásért felelős áramkör kapcsolási rajza a következő ábrán látható:



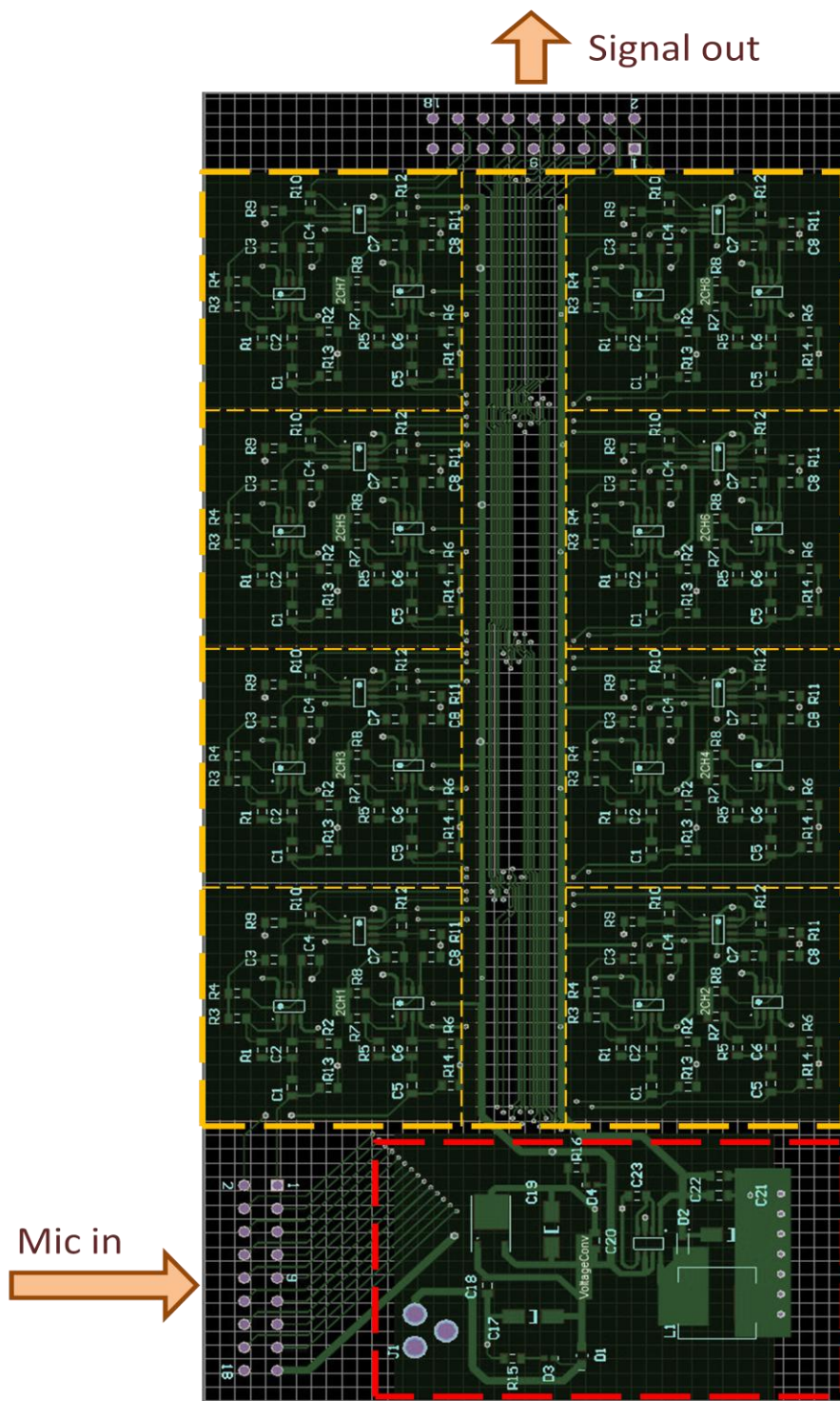
**4.14 ábra:** Tápellátás kapcsolási rajza

#### 4.4.6 Áramkör rajzolat

A jelkondicionáló nyomtatott áramkörének megtervezésekor törekedtem a különböző csatornák egymást zavaró hatásának kiküszöbölésére, csökkentésére. Mivel az áramkör csak kétoldalasra készült, nem volt lehetőségem külön táprétegek kialakítására, azonban fontosnak tartottam a közös impedancián keresztül csatolt zajok minimalizálását. Ezért az egyik oldalt földrétegnek dedikáltam, ezen csak a feltétlenül szükséges átvezetéseket engedtem meg. Ez a törekvésem mutatkozik a tápvezetékek kialakításában is. Mind a pozitív, mind a negatív tápot egy viszonylag vastag buszon vezetem végig. A kezdőpontjuktól a végpontig 5 század voltnál kisebb feszültségesés tapasztalható. A fő tápbuszhoz csatlakoznak kvázi csillagpontoszerűen az egyes részegységek tápjai.

Egy fontos zavarforrásként a fentiekén kívül még figyelembe vettem a kapcsolóüzemű DC/DC átalakítóhoz tartozó dióda-induktivitás-kondenzátor hármason lejátszódó jelenséget. Ezeket az áramköri elemeket megpróbáltam a nyomtatott áramkör szélén elhelyezni a zaj csökkentése érdekében. Továbbá konzulensem tanácsára törekedtem a DC/DC átalakító földje és a nyomtatott áramkör föld rétege közötti lehető legkisebb ellenállás elérésére.

A következő ábra a nyomtatott áramkör felső rétegének rajzolatát szemlélteti. Az ábrán a **Mic in** jelöli a mikrofonokról érkező jelek bemenetét; a **piros kereten** belül található a tápellátás vezetékvezetése; a **sárga szaggatott területen** belül találhatóak a jelkondicionáló fokozatok; a **Signal out** jelzi a jelkondicionáló kártya kimeneteit, melyek utána a NI USB-6363 kártya bemenetéül szolgálnak.



4.15 ábra: A felső nyomtatott áramköri réteg rajzolata

#### 4.4.7 Jelkondicionáló kártya bemérése, hibák javítása

##### 4.4.7.1 Gyártási hibák felderítése és orvoslása

A jelkondicionáló kártya működésének ellenőrzését az alkatrészek beültetése után kezdtem el vizsgálni. Első probléma, mellyel szembesültem, egy rövidzár volt a pozitív tápfeszültség és a föld vezetékek között. A pozitív tápfeszültség stabilizálására hivatott IC ezt különböző füstjelekkel és egy hangjelenséggel adta tudtomra. A hiba helyének meghatározásához egy nagy pontosságú ellenállás mérőt használtam, melyet a tanszék bocsátott rendelkezésemre. Miután a hiba helyét sikerült körülbelül 20 cm<sup>2</sup> pontossággal meghatározni, a kimért területen található összes alkatrészt leforrasztottam, így láthatóvá váltak a pad-ek is. Mivel nagyítóval sem láttam forrasztási hibából adódó zárlatot, elkezdtem gyanakodni, hogy a nyomtatott áramkör gyártása során a nyomtatott vezetékvezés okozza a zárlatot. Mivel a gyártó nem vállalt garanciát, a nyomtatott áramkört újragyártatni költséges lett volna. Nem maradt más választásom, időt nem kímélve, egy hegyes tűvel milliméter mély árkokat vájtam az érintett tápvezeték szakasz mindkét oldalán. Ez végül orvosolta a problémát.

##### 4.4.7.2 Tervezési hibák javítása

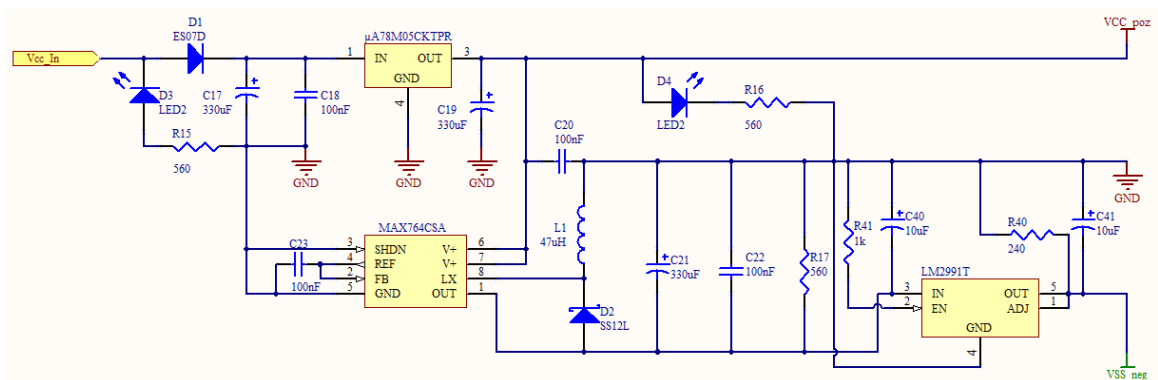
A tápzárlat elhárítása után elkezdtem a jelkondicionáló kártya bemérését. Az első mérési sorozatban egyszerre egy bemeneti csatornát gerjesztettem tisztán harmonikus jelekkel, és leellenőriztem, hogy az össze kimeneti csatornán a várt jelalak figyelhető-e meg. A várt jelalakon a bemenethez tartozó kimeneten az átviteli függvénynek megfelelő jelet, a többi kimeneten pedig nagyon alacsony szintű fehér zajt értek. Meglepő módon, az összes kimeneten azt tapasztaltam, hogy a kimeneti jel spektrumában túszerű zajok jelennek meg, de a meghajtott bemenethez tartozó kimeneten megjelenik maga a gerjesztésre várt válasz is. A nem várt viselkedés diagnosztizálását az erősítők bemeneteinek földre kötésével kezdtem. A kimeneten ismét megjelent a korábban tapasztalt zaj. A viselkedés, miszerint földelt bemenet esetén a műveleti erősítő kimenetén zaj tapasztalható, tápellátásbeli hibákra enged következtetni.

Következő lépés a hiba okának feltárására a tápvezetéseken megjelenő jelek analízisa volt. A várakozásoknak megfelelően mindkét tápvezeték jelének spektrumában ki lehetett mérni a jelvezetéseken is megjelenő tuskéket.

A negatív feszültséget előállító DC/DC konverter tápbemenetét megszakítva a hatás eltűnt a pozitív tápvezetéken is. Ebből arra következtettem, hogy a Maxim DC/DC konverter az adatlapján szereplő 300 kHz-es kapcsolási tranzienseken felül egyéb zavarokkal is szennyezi a kimeneti tápfeszültséget. Ráadásul ezek a tranziensek rendszerint nagymértékű, periodikus terhelésingadozással is párosulnak, így terjedhet tova a pozitív tápfeszültségre is a zavar. Két megoldandó probléma tehát a pozitív táp szennyezése és a negatív tápfeszültség zajszűrése.

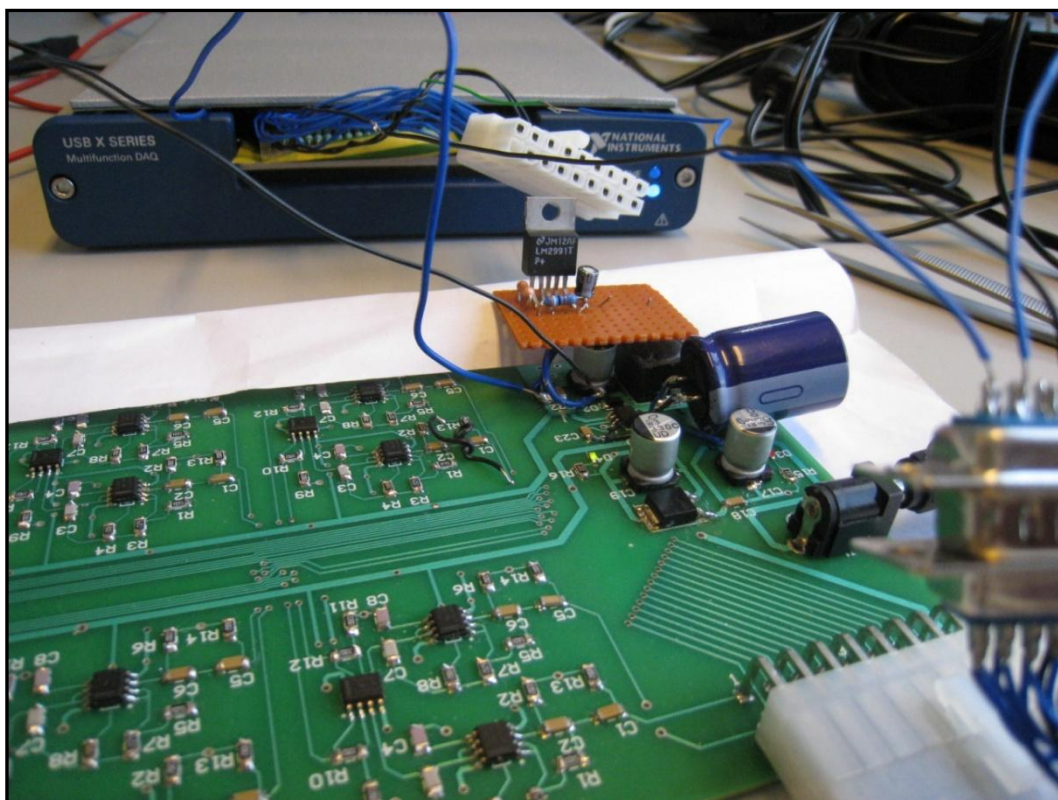
A pozitív tápfeszültségre gyakorolt hatást egy nagy puffer kondenzátor segítségével csökkenteni tudtam, bár sajnos így sem lett teljesen elhanyagolható a maradandó zaj értéke.

A negatív tápfeszültségen megjelenő zaj csökkentésére egy stabilizáló IC-t alkalmaztam, mely bemenetére a DC/DC átalakító kimenetét kötöttem, kimenete pedig az erősítők negatív tápját szolgáltatta. A stabilizáló integrált áramkörrel szemben három fő követelményem volt: alacsony kimeneti feszültségesés; aránylag nagy magas frekvenciás elnyomás, legalább 250 mA terhelhetőség. A választásom a Texas Instruments LM2991T típusjelű áramkörre esett, melynek 0.3 V-os abszolút feszültségesése -5 V-os bemenet mellett, és az adatlap szerint 40 dB-es elnyomást biztosít 10 kHz frekvencián ([17]). A stabilizáló áramkör csatlakoztatásához szükséges az eredeti kapcsolás módosítása, ez a következő rajzon látható.



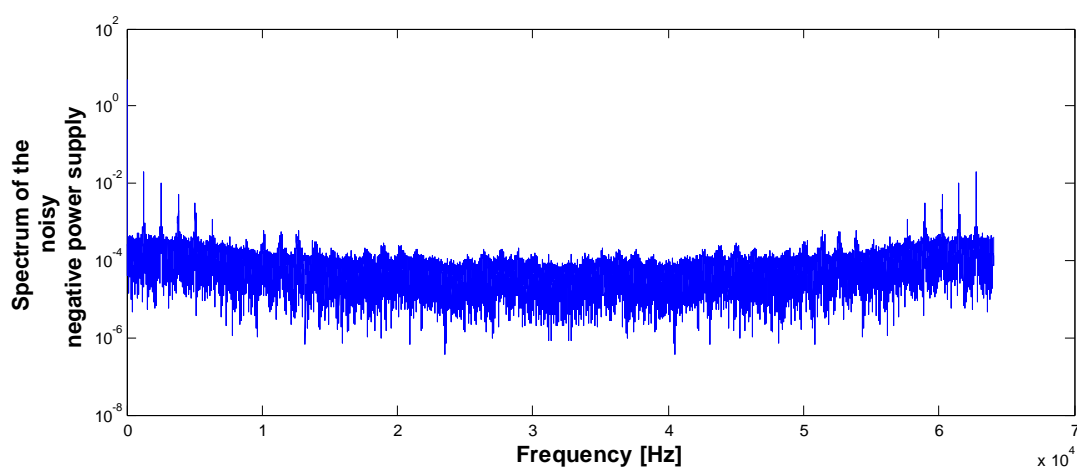
**4.16 ábra:** A tápellátás javított kapcsolási rajza

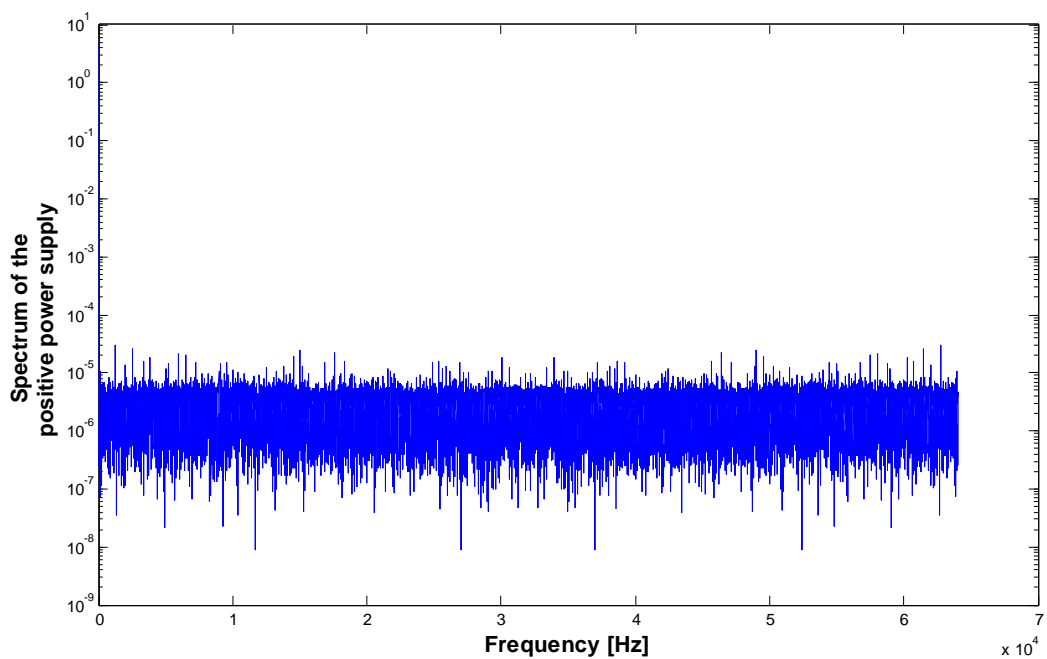
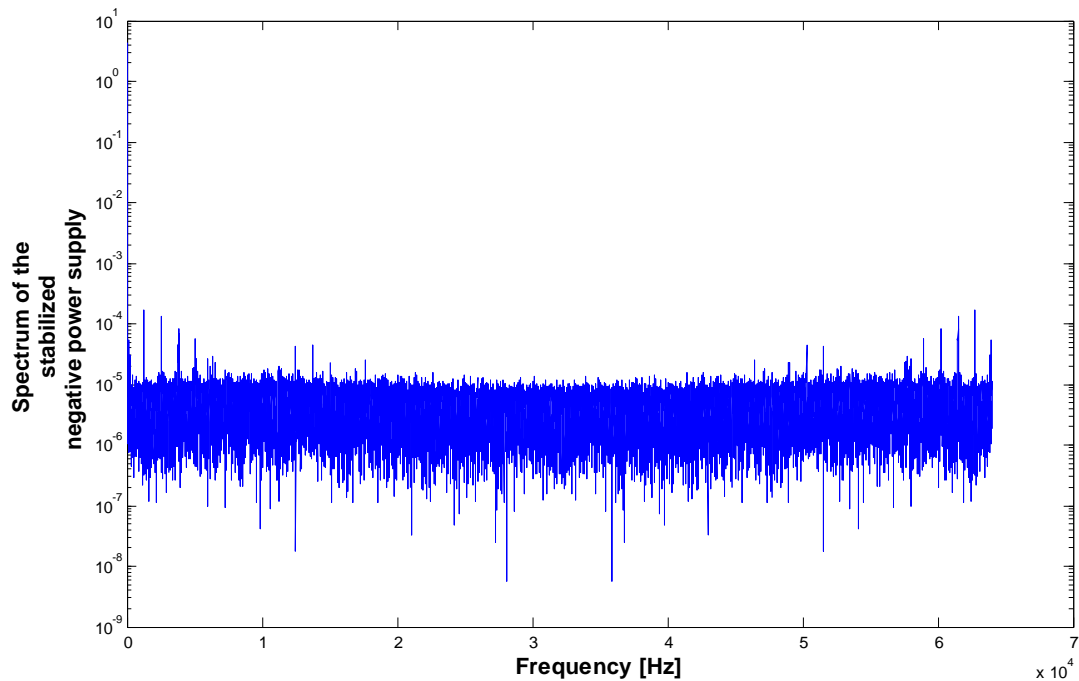
A következő ábra a tárgyalt két javítást (puffer kondenzátor és stabilizáló IC) szemlélteti:



**4.17 ábra:** A tápellátás hibájának javítása

A hardver módosítása után mérésekkel ellenőriztem, hogy a tápvezetéseken valóban csökkent-e a zajszint. Az alábbi ábrák szemléltetik az eredményeket. Az első ábra a zajos negatív táp, a második ábra a szűrt negatív táp, a harmadik ábra pedig a pozitív tápon maradt zaj spektrumát mutatja be.





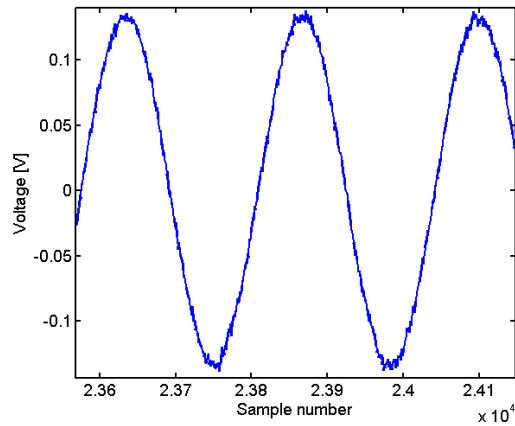
**4.17 ábra:** A tápfeszültségek spektruma

#### 4.4.7.3 A jelkondicionáló fokozat működésének validálása

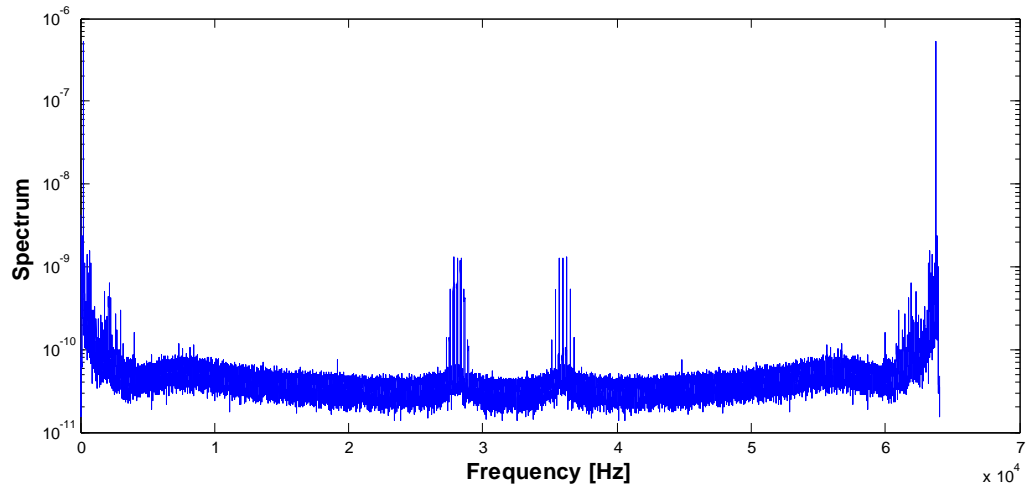
A tápellátásbeli problémák kijavítása után megismételtem az legelső méréssorozatot: a bemeneteket egyesével gerjesztve leellenőriztem a kimeneteken tapasztalható jelalakokat.



Az következő ábrák a 275 Hz-es gerjesztésre adott választ szemléltetik:



**4.18 ábra:** A 275 Hz-es válaszjel időtartománybeli alakja



**4.19 ábra:** A 275 Hz-es gerjesztésre adott válasz spektruma

A válaszjelben megjelenő, 20 kHz frekvencia feletti zavarokat szoftveresen egyszerű szűrni, mivel ez kívül esik a hasznos jel tartományán. A létrehozott jelkondicionáló fokot a hibák javítása után alkalmasnak találtam a kitűzött feladat ellátására.





## 5 Akusztikus nyálábformálás mérések

A mikrofonokról érkező, a jelkondicionáló kártya segítségével szűrt és előerősített jeleket egy National Instruments USB-6363 típusú adatgyűjtő kártyával digitalizáltam. A rögzített mintákat használtam a saját delay-and-sum nyálábformáló algoritmusom működésének vizsgálatára.

A PC-n implementált szoftver együttes három fő feladatot lát el:

- Az USB-6363 mintavevő kártya illesztése a személyi számítógéphez és az adatgyűjtés irányítása.
- Delay-and-sum nyálábformálás megvalósítása.
- Szimuláció végrehajtása a várt viselkedés megállapítására.

Mivel mindhárom felsorolt funkció megvalósítására alkalmas a Windows-os Matlab 2011b programcsomag, ezért a szoftverfejlesztéshez ezt a fejlesztő- és futtatókörnyezetet használtam.

Ebben a fejezetben részletesen tárgyalom a létrehozott szoftverelemek célját és működését, kitérve a tervezésnél figyelembe vett mérnöki megfontolásokra és ismertetve az egyes algoritmusok folyamatábráit. A Matlab kódot az érdeklődő olvasó a **1. Melléklet** megfelelő részében tanulmányozhatja, a kód tartalmaz pár megértést segítő megjegyzést is. A szoftverrendszer bemutatása után ismertetem a fő mérési eredményeimet, összehasonlítva azokat a szimulációk alapján elvárt viselkedéssel és a korábbi, 8 elemű mikrofontömbbel végzett kísérletekkel.

### 5.1 Szoftverrendszer

#### 5.1.1 Mintavevő kártya illesztése

##### 5.1.1.1 Kapcsolat a meghajtó programmal

Hogy a mikrofonok jelét digitalizálni tudjuk, a PC-ről kapcsolatot kell létesíteni a National Instruments (NI) USB-6363-as adatgyűjtő kártyával, majd lekérdezés-válasz (kliens-szerver) alapon vezérelni magát az adatgyűjtési folyamatot. Erre a NI két magas szintű interfészt ad a felhasználónak ([11]).

Egyik opció a Data-Aquisition – Measurement and Automation Explorer (MAX) használata, mely közvetlenül a mérőkártya meghajtó programjával teremt egy alacsony szintű kapcsolatot. A felhasználó a MAX kezelőfelületének segítségével úgynevezett adatgyűjtő taszkokat definiálhat az egyes mintavételi eszközökhöz. A taszk absztrakciós szintjén beállítható egy adatfolyam-szerű, idővezérelt működés, de ez alacsony szinten ugyanúgy eseményvezérelt lekérdezés-válasz módon valósul meg. Ez nem a legkedvezőbb a valós idejű működéshez, de a közvetlen driver kapcsolat miatt az átlagos késleltetés mérhető, és a jitter kézben tartható.

A másik magas szintű alternatíva a felhasználó számára a *Matlab Data Acquisition Toolbox* használata ([10]), mely a 2011b Matlab verziótól kezdve támogatja a NI USB-6363-at is. Ez a megoldás a Matlab magján keresztül létesít kapcsolatot a DAQ driverrel, majd a driveren keresztül kliens-szerver alapon kommunikál a mintavevő kártyával. A két köztes réteg miatt ez a megoldás nagyobb átlagos késleltetéssel jár, és a jitter számolása is nehezebb feladat. Mivel nálam nem követelmény a real-time működés, és a szoftverrendszer többi részét Matlabban implementáltam, ezért ezt az opciót választottam az adatgyűjtő kártyával való kommunikációra.

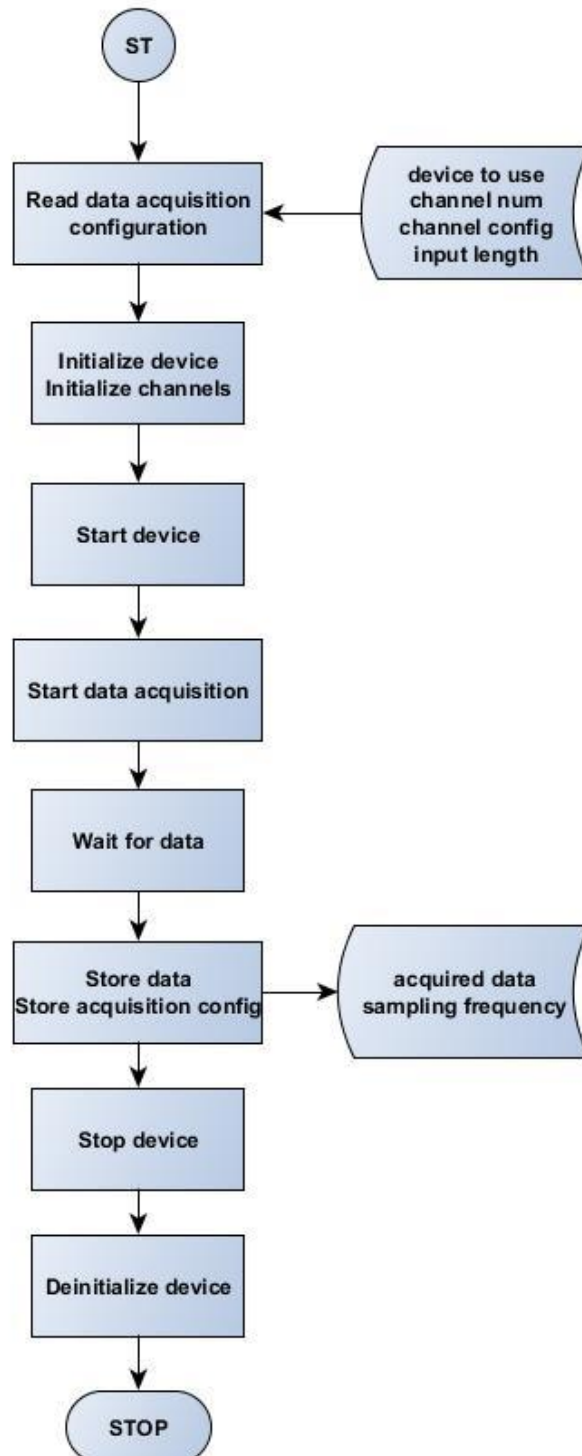
A teljesség igényével megemlítem, hogy a NI a felhasználók rendelkezésére bocsát egy alacsony szintű C könyvtárat is, mellyel a mérési adatfolyam közvetlenül irányítható. Ha esetleg a kedves olvasó valaha valós idejű mintavételt szeretne megvalósítani, mindenképp ennek a könyvtárnak a használatát ajánlom, valamiféle real-time operációs rendszeren. Kis szépséghiba, hogy jelenleg a NI-DAQmx Linuxra portolt változata mindent támogat, csak az USB-s mintavevő eszközöket nem.

#### 5.1.1.2 Adatgyűjtés Matlab alól

A Matlab Data Acquisition Toolbox két különböző absztrakciót támogat a mintavételi kártyával való kapcsolatteremtésre. Ezek közül az egyik interfészt nyújt a NIDAQmx driver összes alapvető funkciójához; a másik pedig egy session-alapú megközelítést támogat, mellyel akár több, külön szálon is indíthat a felhasználó lekérdezéseket, így alkalmas a program futási idejének optimalására is.

Én az egyszerűbb, nem session-alapú megoldást választottam, hisz az esetben az adatgyűjtés a feldolgozástól időben teljesen elkülönül, offline módon történik.

Az alábbi folyamatábra az adatgyűjtéshez használt algoritmust írja le.



5.1 ábra: Az adatgyűjtő algoritmus folyamatábrája

### 5.1.2 Delay-and-sum nyalábformálás algoritmus

A delay-and-sum nyalábformáló algoritmust a 2.1 fejezetben leírt elméletnek megfelelően implementáltam. Mivel a véges és a végtelen fókuszú nyalábformálás lényegében csak az alkalmazandó késleltetés számításában különbözik, a szoftverem felkészíttem mindkettő támogatására.

A program bemeneként a már rendelkezésre álló hangfelvételeket, a nyalábformálás módjától függően a fókuszpont koordinátáit, vagy a fókuszirányt, és a konfigurációs paramétereket várja. A referenciapont a mikrofontömb térbeli középpontja. Az algoritmus kimeneként az adott pontra, vagy irányba fókuszált, nyalábformált hangot szolgáltatja.

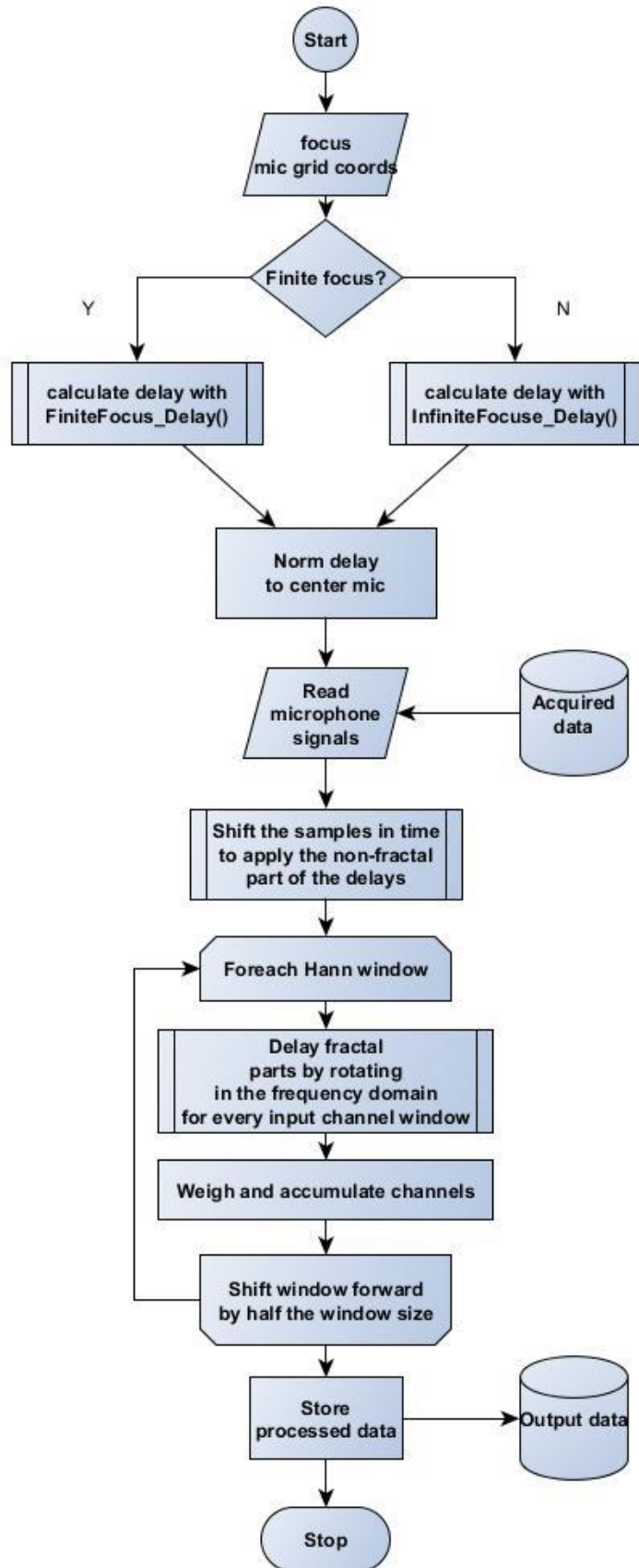
Az algoritmus úgynevezett mozgó-ablakos módszerrel (Hann ablakot használva) dolgozza fel a jeleket, így minimális módosítással felkészíthető akár valós idejű feldolgozásra is.

Az algoritmus folyamatábráját az **5.2 ábra** szemlélteti.

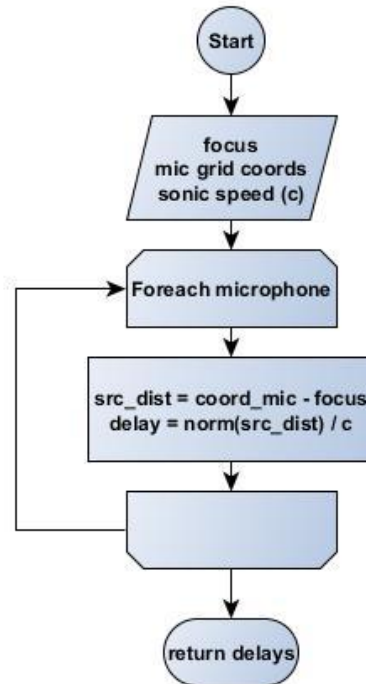
### 5.1.3 Szimulációs szoftver

A szimulációs program lényegében egy adott forrásból származó, harmonikus hang fázisát számolja ki a tér különböző pontjaiban. A térpontokba mikrofonokat képzelve a szimuláció a nyalábformáló algoritmus bemeneti hangfelvételeit állítja elő. Így a hardver összeállítást megkerülve lehet bemeneti adatokhoz jutni, és lehetőség nyílik kiértékelni a jelátvitelt, a jelkondicionáló fokozat és az A/D átalakítás minőségét. Segítségével könnyebben választ lehet kapni, hogy hiba esetén az algoritmus, vagy a hardver korlátaiba ütköztünk-e.

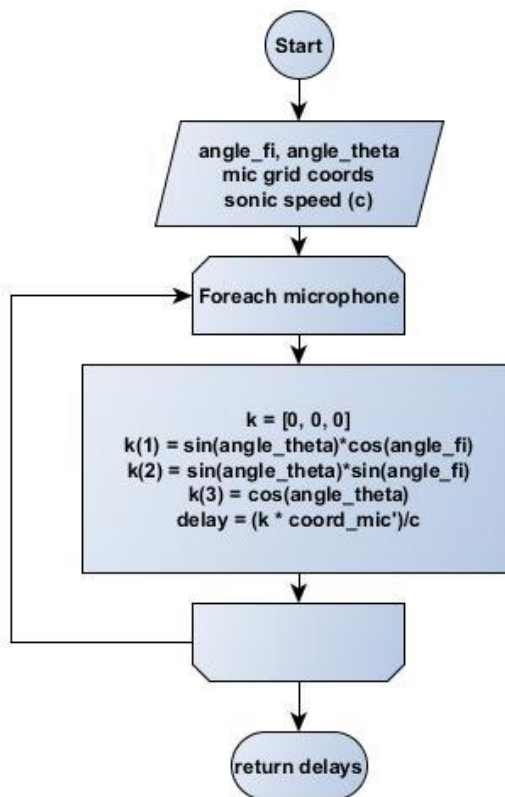
A szimuláció bemenete a harmonikus hang frekvenciája, a forrás koordinátái és a térpontok koordinátái, ahol a forrás jelét „rögzítjük”; kimenete pedig a szimulált hangsávok. A szimuláció kiterjeszhető több forrásra is úgy, hogy azt a forrásokra egyenként lefuttatjuk, majd az eredményt összegezzük. Természetesen ebben az esetben a virtuális mikrofontömb koordinátáit állandó értéken kell tartani a futtatások között.



5.2 ábra: Delay-and-sum nyalábformálás folyamatábrája



5.3 ábra: A véges fókusz távolságú nyílábformálás késleltetés számítása



5.4 ábra: A végtelen fókusz távolságú nyílábformálás késleltetés számítása

#### 5.1.4 Hangtérkép

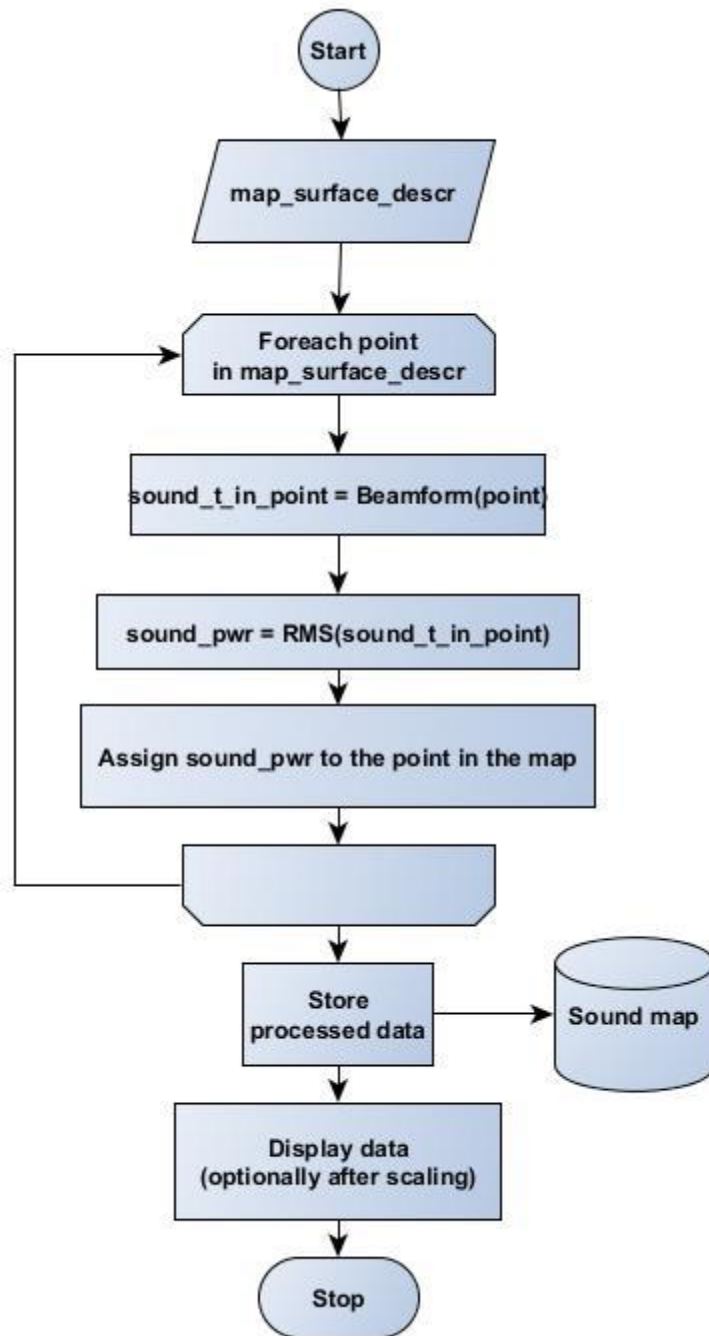
A hangtérkép arról hordoz információt, hogy a referenciapontban álló megfigyelő a tér különböző pontjaiból, illetve irányából milyen erősségű hangot érkezését érzékeli.

A hangtérkép előállításához véges fókusztávolságú nyalábformálásnál bemenetként megadott, korlátos sík (téglalap) pontjaira el kell végezni a 5.1.2 fejezetben ismertetett nyalábformáló algoritmust. Végtelen fókusztáv esetén a nyalábformáló algoritmust egy képzeletbeli, végtelen sugarú gömbszelet felületén kell lefuttatni. A mellékletben a hangtérkép létrehozására optimált kódot közöltem. Ha az olvasó a kimeneti jelet az időtartományban kívánja feldolgozni, kérem, a kevesebb zaj érdekében módosítsa a frekvenciatartományban végrehajtott négyzetablak szűrést a feldolgozási céljának megfelelő, laposabb frekvenciamenetűre. A futási idő csökkentése érdekében a felhasználónak lehetősége van a hangtérkép előállításakor csak egy előre kiválasztott frekvenciát vizsgálni.

A hangtérkép generáló algoritmus bemenetei a hangfelvételek, a határpontok, illetve határ-térszögek, a térkép felbontása és a nyalábformáló algoritmus konfigurációs paraméterei.

A hangtérképet előállító szoftver folyamatábráját az **5.5 ábra** mutatja be.





5.5 ábra: A hangtérkép készítésének folyamata

## 5.2 Süketszobás bemérés

Az elkészült hardver-szoftver összeállítás tervezettnek megfelelő működésének igazolására a BME Hálózati Rendszerek és Szolgáltatások Tanszék, I épületben található hangcsillapított szobájában végeztem méréseket. A lehetőség biztosításáért külön köszönetet mondok Nagy Attila Balázsnak.

### 5.2.1 Mérési elrendezés, mérési körülmények

Az akusztikus kamera bemérését igyekeztem úgy végezni, hogy az eredmények összehasonlíthatóak legyenek a korábbi, 8 elemű, lineáris mikrofontömbbel végzett mérésekkel. Ehhez helyre kellett állítanom egy, a korábbival minél jobban egyező elrendezést. (A korábbi elrendezést az 1. Melléklet tartalmazza.) Várhatóan az új hangkamera jobb érzékenységgel fog rendelkezni az elődjénél, és 200 Hz alatt is elfogadható szelektivitást lehet vele elérni.

A süketszobában a mérés ideje alatt az átlaghőmérséklet  $29^{\circ}\text{C}$ , az átlagos páratartalom 41% volt. Az egyik mérési összeállítást, melyben a jobb oldali hangfal az E1, a bal oldali pedig az E3 pozícióban van, az **5.1 ábra** szemlélteti:



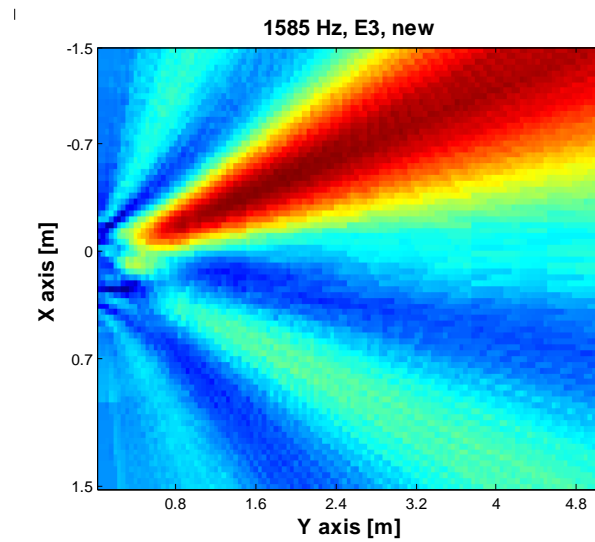
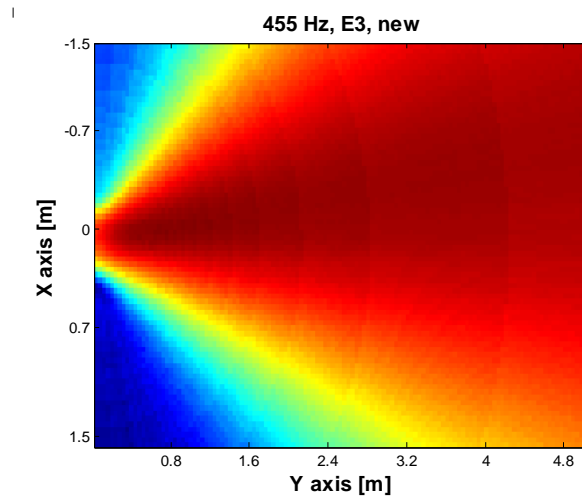
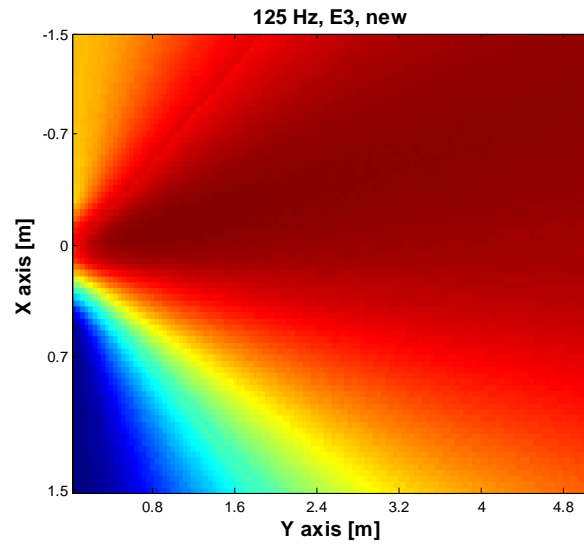
**5.1 ábra:** Süketszobás mérési elrendezés

### 5.2.2 Az új és a régi nyálábformáló rendszer összehasonlítása

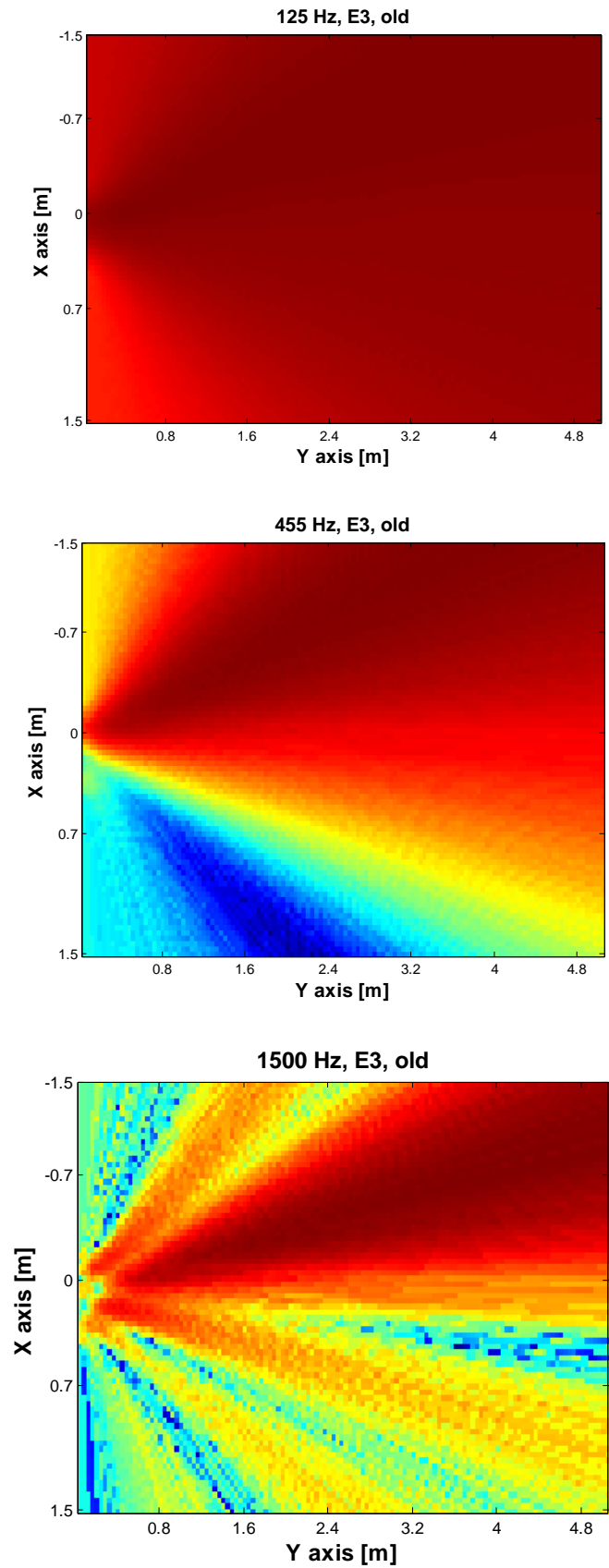
A régi és az újonnan létrehozott akusztikus kamera és a jeleket feldolgozó nyálábformáló algoritmus összehasonlítására legkézenfekvőbb mód az E3(-0.8, 3.6) elrendezésben felvett harmonikus hangok véges fókusz távolságú hangtérképének összehasonlítása 125 Hz, 455 Hz és 1500 Hz (1585 Hz) frekvenciákon.

Az első három ábra az új rendszerrel készített felvételek, míg a második három ábra a régi rendszer képeit szemléltetik közel azonos frekvenciájú gerjesztő hangok esetén. Az ábrák mind felülnézeti metszetet mutatnak.

Megfigyelhető, hogy az új akusztikus kamera alacsony frekvencián jobb szelektivitást mutat, míg a melléknyalábok körülbelül ugyanannál a frekvenciánál jelentkeznek.



**5.2 ábra:** Az új nyálábformáló rendszerrel készített akusztikus képek



**5.3 ábra:** A régi rendszerrel készített akusztikus képek rendre 125 Hz, 455 Hz és 1500 Hz frekvenciájú harmonikus források esetén

Az új mérőrendszer az előzetes várakozásoknak megfelelően alacsony frekvenciákon jobbnak bizonyult a réginél, míg magas frekvencián körülbelül ugyanolyan érzékenységet lehetett elérni vele, kedvezőbb az oldalnyalábok aránya a főnyalábhoz képest. Ezen felül az új rendszer ortogonális elrendezéséből adódóan több szabadsági fokon lehet értelmezni a mért adatokat.

## 6 Összegzés

A diplomamunkám során létrehoztam egy akusztikus kamerát, mely egy 15 elemű, kereszt elrendezésű mikrofontömbből, egy jelkondicionáló fokozatból és egy mintavevő kártyából állt. A mikrofonok által rögzített jelek a jelkondicionáló fokozathoz két darab Cat5e típusú FTP kábelen keresztül érkeznek, ezzel az előzetes várakozásoknak megfelelően a mikrofontömb a jelkondicionáló fokozat 10 méter sugarú környezetében elhelyezhető anélkül, hogy a jelekre továbbítás közben szignifikáns zaj rakódna. A jelkondicionáló fokozat a tápellátás hibáinak javítása után a mikrofonokról érkező jelek jel-zaj viszonyát a mérések alapján körülbelül 30 dB-lel javította a 150 Hz és 8 kHz közötti frekvenciatartományban. Az analóg jeleket egy National Instruments USB-6363 típusjelű mintavevő kártya segítségével digitalizáltam a későbbi feldolgozáshoz. Az A/D átalakítás mind a 15 bemeneti csatornán 64 kHz-es mintavételi frekvenciával történt a -1 V és +1 V közötti feszültségtartományban. A mintavevő kártya gyárilag egy kapcsolóüzemű hálózati adatterről van táplálva, de a tapasztalataim alapján érdemes e helyett egy labortápot használni, így az digitalizált jeleken alacsonyabb zaj terheli. Említésre méltó még, hogy a mintavevő kártya a mintavételi frekvencia és a mintavételezett csatornaszám függvényében különböző nemlineáris, additív zajokat produkál, melynek okát sajnos nem sikerült meghatároznom.

A digitalizált jeleket feldolgozó nyálábformáló szoftver off-line működésre lett felkészítve, tehát a nyálábformálás nem a mintavétellel párhuzamosan (mintaablaknyi késleltetéssel) történt. Ez a hangtérképek készítésénél elfogadható megkötés. A valós idejű nyálábformáláshoz további fejlesztésekre van szükség. További problémát jelent az algoritmus hosszú futási ideje. Egy 10 m<sup>2</sup> területű, 25 cm<sup>2</sup> felbontású sík térkép több perc alatt készül el. A futást gyorsítani lehet, ha csak bizonyos frekvenciákra korlátozzuk a feldolgozást.

A hangcsillapított szobában végzett mérések a korábbi tapasztalatokkal egyező eredményeket hoztak, és igazolták a nyálábformáló rendszer helyes működését.

# Irodalomjegyzék

- [1] Holics, László: Fizika 1. kötet (Klasszikus fizika), Második kiadás, ISBN 963 10 9452 9, Műszaki Könyvkiadó, Budapest, 1992
- [2] Balázs Péter – Hangforrás lokalizálása mikrofontömbbel (BME Villamosmérnöki és Informatikai Kar, Híradástechnikai Tanszék, 2008)
- [3] J. J. Christensen, J. Hald – Beamforming, Brüel & Kræf Technical Review (1/2004)
- [4] Ulf Michel: History of Acoustic Beamforming  
Berlin Beamforming Conference, Berlin, 11/2006  
([http://elib.dlr.de/47021/1/BeBeC\\_2006\\_Paper\\_Michel.pdf](http://elib.dlr.de/47021/1/BeBeC_2006_Paper_Michel.pdf))
- [5] Acoustic Localization of Sound Mirrors  
(<http://www.douglas-self.com/MUSEUM/COMMS/ear/ear.htm>)  
letöltés dátuma: 2013.12.18
- [6] *T. Betlehem, R. C. Williamson* Acoustic Beamforming Exploiting Directionality of Human Speech (10/2003)
- [7] *Tomi Engdahl* – Powering microphones  
([http://www.epanorama.net/circuits/microphone\\_powering.html](http://www.epanorama.net/circuits/microphone_powering.html))  
letöltés dátuma: 2014.05.20
- [8] Harm Jan Willem Belt, Cornelis Pieter Janse: Method and device for acoustic echo cancellation combined with adaptive beamforming  
United States Patent number 7035415 B2 (4/2006)
- [9] Altium Designer 2009 Winter Documentation  
(<http://wiki.altium.com/display/ADOH/Getting+Started+with+Altium+Designer>)
- [10] Matlab 2011b Data Acquisition Toolbox Documentation  
(<http://www.mathworks.de/de/help/daq/index.html>)
- [11] National Instruments: Traditional NI-DAQ User Manual  
(<http://digital.ni.com/manuals.nsf/websearch/74E3CF846234B47E86256CB7005DF156> és <http://www.ni.com/pdf/manuals/370696a.pdf>)
- [12] Roland Cakewalk UA-101 Owner's Manual  
(<http://www.roland.com/support/article/?q=manuals&p=UA-101>)
- [13] National Instruments DAQ-6062 User Manual  
(<http://www.ni.com/pdf/manuals/322641a.pdf>)



- [14] National Instruments USB-6363 User Manual  
(<http://www.ni.com/pdf/manuals/370083d.pdf> és  
<http://www.ni.com/pdf/manuals/370784d.pdf>)
- [15] Texas Instruments TL08xx JFET-Input Operational Amplifiers Datasheet  
(<http://www.ti.com/lit/ds/symlink/tl084m.pdf>)
- [16] Texas Instruments UA78M05CKTPR Datasheet  
(<http://www.ti.com/lit/gpn/ua78m05>)
- [17] Texas Instruments LM2991T Datasheet  
(<http://www.ti.com/lit/gpn/lm2991>)
- [18] Maxim MAX764CSA Datasheet  
(<http://datasheets.maximintegrated.com/en/ds/MAX764-MAX766.pdf>)

# Mellékletek

## 1. Melléklet Nyalábformáló szoftver forráskódjai

### Beamform\_Alg.m

```
function Out_Sound = Beamform_Alg( mic_ReceivedSignal,
sampling_freq, coord_MicGrid, focus, c, alg_Type )

[num_mic, signal_len] = size(mic_ReceivedSignal);

%calculate the sound propagation delay to each microphone
dt_MicGrid = zeros(num_mic,1);
if(strcmp(alg_Type,'infinite') == 1)
    if(len(focus) == 2)
        fi = focus(1);
        theta = focus(2);
    else
        fi = 0;
        theta = 0;
    end
    dt_MicGrid = InFiniteFocus_Delay( coord_MicGrid, fi,
theta, c );
else
    if(len(focus) == 3)
        coord_Focus = focus;
    else
        coord_Focus = [0, 0, 0];
    end
    dt_MicGrid = FiniteFocus_Delay( coord_MicGrid,
coord_Focus, c );
end;

%norm the delays to the center microphone
o_delay = dt_MicGrid(1);
dt_MicGrid = dt_MicGrid - o_delay;

%calculate the delay in bins
dk_MicGrid = dt_MicGrid * sampling_freq;
```

```

        %delay the non-fractal (whole) bins in the time domain
        max_delay_bin = max(dk_MicGrid);
        min_delay_bin = min(dk_MicGrid);      %NOTICE: min_delay_bin
is less than or equals 0.
        %signal_len = signal_len + ceil(min_delay_bin) +
floor(max_delay_bin) + 2;
        orig_signal_len = signal_len;
        signal_len = signal_len + abs(floor(min_delay_bin)) +
ceil(max_delay_bin); %igen, ez jó
        semiDelayed_MicSignals = zeros(num_mic, signal_len);
        for i = 1:num_mic;
            %delayed_signal = recd_signal[k + dk]
            delayed_start_i = floor(max_delay_bin) + 2;      %This is
the start index for the non-delayed signal.
            if(dk_MicGrid(i) > 0)
                delayed_start_i = floor(max_delay_bin) + 2 -
floor(dk_MicGrid(i));
                dk_MicGrid(i) = dk_MicGrid(i) -
floor(dk_MicGrid(i));      %The fractal part.
            else
                delayed_start_i = floor(max_delay_bin) + 2 +
abs(ceil(dk_MicGrid(i)));
                dk_MicGrid(i) = dk_MicGrid(i) - ceil(dk_MicGrid(i));
%The fractal part.
            end;

            semiDelayed_MicSignals(i,
delayed_start_i:(delayed_start_i + orig_signal_len - 1)) =
mic_ReceivedSignal(i, :);
            end;

        beamformed_signal = zeros(1, signal_len);

        %window the infinite signals (50% overlap; Hann window)
        Nfft = 512;
        N_window = Nfft + 2;
        beamformed_window_freqdom = zeros(1, N_window);
        hann_window = hann(N_window);
        %First window
        %i is the iterator for microphone indexes
        for i = 1:num_mic;
            %Transform the first windowed part into the frequency
domain
            %hann(N) = [0 hanning(N) 0]

```

```

        delayed_window_freqdom =
fft(semiDelayed_MicSignals(i,1:Nfft+2) .* hann_window',N_window);

        %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))

        %positive frequencies (W(th)*e^(-j*th*dt))
        for k = 2:ceil(N_window/2);
            delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(-j * (k-1) * (2*pi/(N_window-1))) *
dk_MicGrid(i));
        end

        %negative frequencies (W(th)*e^(j*th*dt))
        for k = ceil(N_window/2)+1:N_window;
            delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(j * (N_window-k+1) * (2*pi/(N_window-
1))) * dk_MicGrid(i));
        end

        beamformed_window_freqdom = beamformed_window_freqdom +
delayed_window_freqdom;
    end

    %Transform the first window back to the time domain.
    beamformed_signal(1:N_window) =
real(ifft(beamformed_window_freqdom/num_mic, N_window));

    %All the other windows but the last
    %s is the iterator of windowing
    last_s = 0;
    for s = floor(Nfft/2)+2:floor(Nfft/2):signal_len-Nfft
        beamformed_window_freqdom = zeros(1, N_window);
        %i is the iterator for microphone indexes
        for i = 1:num_mic;
            delayed_window_freqdom = fft([0,
semiDelayed_MicSignals(i,s:s+Nfft-1), 0].* hann_window', N_window);

            %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))

            %positive frequencies (W(th)*e^(-j*th*dt))
            for k = 2:ceil(N_window/2);
                delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(-j * (k-1) * (2*pi/(N_window-1))) *
dk_MicGrid(i));
            end

            %negative frequencies (W(th)*e^(j*th*dt))

```

```

        for k = ceil(N_window/2)+1:N_window;
            delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(j * (N_window-k+1) * (2*pi/(N_window-
1)) * dk_MicGrid(i));
            end

            beamformed_window_freqdom =
beamformed_window_freqdom + delayed_window_freqdom;
            end

            %Transform the first window back to the time domain.
            beamformed_signal(s-1:s+Nfft) = beamformed_signal(s-
1:s+Nfft) + real(iff(beamformed_window_freqdom/num_mic, N_window));

            last_s = s;
        end

        %The last window
        last_part_length = signal_len - (last_s+floor(Nfft/2)) + 2;
        beamformed_window_freqdom = zeros(1, N_window);
        %i is the iterator for microphone indexes
        for i = 1:num_mic;
            %For proper windowing the signal must be extended with 0
values
            %(until its length reaches the window length).
            delayed_window_freqdom =
fft([semiDelayed_MicSignals(i,last_s+floor(Nfft/2)-1:signal_len),
zeros(1,N_window-last_part_length)] .* hann(N_window)', N_window);

            %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))
            %positive frequencies (W(th)*e^(-j*th*dt)
            for k = 2:ceil(N_window/2);
                delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(-j * (k-1) * (2*pi/(N_window-1)) *
dk_MicGrid(i));
                end
            %negative frequencies (W(th)*e^(j*th*dt))
            for k = ceil(N_window/2)+1:N_window;
                delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(j * (N_window-k+1) * (2*pi/(N_window-
1)) * dk_MicGrid(i));
                end
        end
    end

```

```

        beamformed_window_freqdom = beamformed_window_freqdom +
delayed_window_freqdom;
    end

    %Transform the first window back to the time domain.
    %figure;
    %plot(beamformed_signal);
    %figure;
    %plot(real(ifft(beamformed_window_freqdom/num_mic,
mod_Nfft)));
    last_part = real(ifft(beamformed window freqdom/num mic,
N_window));
    beamformed_signal(last_s+floor(Nfft/2)-1:signal_len) =
beamformed_signal(last_s+floor(Nfft/2)-1:signal_len) +
last_part(1:last_part_length);

    Out_Sound = beamformed_signal;
end

```

### **daq\_6363.m**

```

close all;
clear all;

num_meres = 10;
len_meres = 12800;

ai = analoginput('nidaq','Daq6363');

chan = addchannel(ai, 20:20);
set(ai,'InputType','Differential');
%set(ai,'InputType','SingleEnded');
Fs = setverify(ai,'SampleRate',64000);
N = setverify(ai,'SamplesPerTrigger',len_meres);
data = zeros(num_meres, N);

set(chan,'SensorRange',[-0.1 0.1]);
set(chan,'InputRange',[-0.1 0.1]);
set(chan,'UnitsRange',[-0.1 0.1]);
set(chan,'Units','V');

for(i=1:num_meres)

```

```

        start(ai);
        wait(ai,2*N/Fs);
        data(i,:) = getdata(ai);
    end;

    data_f = zeros(1, N);
    for i=1:num_meres
        data_f = data_f + abs(fft(data(i,:)));
    end;
    data_f = data_f / num_meres;

    figure;
    semilogy(0:(Fs/N):Fs-(Fs/N),data_f/N);

    for i=1:1
        figure;
        plot(data(i,:));
    end;

    % figure;
    % plot(data(:,1) - data(:,2));
    %
    % figure;
    % plot(0:(Fs/N):Fs-(Fs/N),abs(fft(data(:,1) - data(:,2)))/N);

    stop(ai);
    delete(ai);
    clear ai;

```

### **Beamform\_SoundPropag.m**

```

%SoundPropag
%This function simulates the sound propagation from a point-like
source in a barrier-free environment.
%INPUT parameters:
%   coord_Source      -      Coordinates of the source
(dimension: 1x3)
%   coord_MicGrid     -      Coordinates of the microphones
(dimension: nx3)
%   c                 -      Sonic speed in m/s

```

```

function [Out] = Beamform_SoundPropag(coord_Source,
coord_MicGrid, test_wavfreq, c)

format long;

%create a sine wave (testing purposes only)
%test_wavfreq = 1005;
test_fs = 44.1*(10^3); %sampling freq
test_Ns = 2^11; %number of samples (sample
buffer length)

test_sampleTimes = [0:1/test_fs:(test_Ns-1)/test_fs];
test_wave = sin(2*pi*test_wavfreq * test_sampleTimes);
%test_wave = sin(2*pi*test_wavfreq * test_sampleTimes); *
test_sampleTimes).*sin(2*pi*2 * test_sampleTimes);

%get the number of microphones
num_mic = size(coord_MicGrid,1);

%calculate the sound propagation delay to each microphone
dt_MicGrid = zeros(num_mic,1);
for i = 1:num_mic
    dist = coord_MicGrid(i,:) - coord_Source;
    dt_MicGrid(i) = (norm(dist))/c;
end

%calculate the delay in bins
dk_MicGrid = dt_MicGrid * test_fs;

%delay the non-fractal (whole) bins in the time domain
%NOTICE: all elements in dt and dk_Mic_Grid are not-negative
max_delay_bin = max(dk_MicGrid);
%signal_len = signal_len + ceil(min_delay_bin) +
floor(max_delay_bin) + 2;
signal_len = test_Ns + ceil(max_delay_bin); %igen, ez jó
semiDelayed_MicSignals = zeros(num_mic, signal_len);
for i = 1:num_mic;
    %delayed_signal = recd_signal[k + dk]
    delayed_start_i = floor(dk_MicGrid(i)) + 1;
    dk_MicGrid(i) = dk_MicGrid(i) - floor(dk_MicGrid(i));
%The fractal part.

```



```

        semiDelayed_MicSignals(i,
delayed_start_i:(delayed_start_i + test_Ns - 1)) = test_wave;
        end;

        test_wave_i = test_wave;

        %Delay the signal in each mic
        mic_receivedSignal = zeros(num_mic,signal_len);
        Nfft = signal_len;

        %i is the iterator for microphone indexes
        for i = 1:num_mic;
            mic_recd_freqdom = zeros(1,Nfft);
            semiDelayed_MicSignals_f =
fft(semiDelayed_MicSignals(i,:),Nfft);

            %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))
            %positive frequencies (delayed = W(th)*e^(j*th*delay)
            for k = 2:ceil(Nfft/2);
                mic_recd_freqdom(k) = semiDelayed_MicSignals_f(k) *
exp(j * (k-1) * (2*pi/Nfft) * dk_MicGrid(i));
            end
            %negative frequencies (delayed = W(th)*e^(-j*th*delay))
            for k = ceil(Nfft/2)+1:Nfft;
                mic_recd_freqdom(k) = semiDelayed_MicSignals_f(k) *
exp(-j * (Nfft-k+1) * (2*pi/Nfft) * dk_MicGrid(i));
            end

            mic_receivedSignal(i,:) = real(iff(mic_recd_freqdom,
Nfft));
        end
        Out = mic_receivedSignal;

        %debug
        % figure;
        % hold on;
        % plot(test_wave,'green');
        % plot(mic_receivedSignal(1,:));
        % hold off;

        % phase = 180 * asin(real(mic_receivedSignal(:,1))) / pi;

```

```

% eye(1);

end

```

### FiniteFocus\_Delay.m

```

function dt_MicGrid = FiniteFocus_Delay( coord_MicGrid,
coord_Focus, c )
    dt_MicGrid = zeros(num_mic,1);
    for i = 1:num_mic
        dist = coord_MicGrid(i,:) - coord_Focus;
        dt_MicGrid(i) = (norm(dist))/c;
    end
end

```

### InFiniteFocus\_Delay.m

```

function dt_MicGrid = InFiniteFocus_Delay( coord_MicGrid,
angle_fi, angle_theta, c )
    %dt_MicGrid = zeros(num_mic,1);
    k = [sin(angle_theta)*cos(angle_fi),
sin(angle_theta)*sin(angle_fi), cos(angle_theta)];
    dt_MicGrid = (k * coord_MicGrid')/c;
end

```

### CrossGrid15.m

```

% This function generates a representation for the cross-shaped
% microphone grid used during the beamforming measurements.
% Microphones are situated in the X-Z plane.

function [MicGrid] = CrossGrid15(centre_coord)
    MicGrid = zeros(15, 3);

    %B1 -> B7 (AI1 -> AI7)
    for i = 1:1:7
        z_offset = 0.3 - ((i-1) * 0.1);
        MicGrid(i,:) = [0 0 z_offset] + centre_coord;
    end;

```

```

%B8 (AI8)
MicGrid(8,:) = [-0.1 0 0] + centre_coord;

%A2 -> A4 (AI9 -> AI11)
MicGrid(9,:) = [-0.4 0 0] + centre_coord;
MicGrid(10,:) = [-0.3 0 0] + centre_coord;
MicGrid(11,:) = [-0.2 0 0] + centre_coord;

%A5 -> A8 (AI12 -> AI15)
MicGrid(12,:) = [0.1 0 0] + centre_coord;
MicGrid(13,:) = [0.2 0 0] + centre_coord;
MicGrid(14,:) = [0.3 0 0] + centre_coord;
MicGrid(15,:) = [0.4 0 0] + centre_coord;

end

```

### SourceLocation.m

```

c = 344; %sonic speed
micGrid = CrossGrid15([0 0 0]);

n_meres = 1; %Meres sorszama

sample_range_low = 1.5*10^5;
sample_range_high = sample_range_low+4095;

% mics_Received = zeros(8,length(sample_range_low :
sample_range_high));

load('test.mat');

mics_Received = data(sample_range_low:sample_range_high,:);
sampling_freq = Fs;

% for i = 1:8
% [mics_Received(i,:), sampling_freq, nbits] =
wavread(sprintf('%s%d%s%d%s%d%s', 'Input\m', n_meres, '_', freq, '_emberiha
ng-', i, '.wav'), [sample_range_low sample_range_high]);
% [mics_Received(i,:), sampling_freq, nbits] =
wavread(sprintf('%s%d%s%d%s%d%s', 'Input\m', n_meres, '_', freq, '-
', i, '.wav'), [sample_range_low sample_range_high]);
% end;

```

```

grid_res = 0.025;           %test grid resolution in meter
width_border_half_dist = 1.5; %in meter
depth_border_dist = 5;      %in meter
colormap_len = 256;

num_mic = size(mics_Received,1);
centre = sum(micGrid) / num_mic; %test centre coordinates
Z_offset = -0.17;
centre(3) = centre(3) + Z_offset;

test_plane_amp = zeros((2*width_border_half_dist/grid_res)+1,
(depth_border_dist/grid_res));

d_i=0;

for d_coord = 0 : grid_res : centre(2) + depth_border_dist
    tic
    d_i=d_i+1;
    w_i=0;
    for w_coord = (centre(1) - width_border_half_dist) :
grid_res : (centre(1) + width_border_half_dist)
        w_i=w_i+1;
        focus = [w_coord d_coord centre(3)];

        %calculate the delay
        dk_MicGrid = zeros(num_mic,1);
        for i = 1:num_mic
            dk_MicGrid(i) = (norm(micGrid(i,:) - focus))/c;
        end
        dk_MicGrid = (dk_MicGrid - dk_MicGrid(1)) *
sampling_freq;

        %delay the non-fractal (whole) bins in the time domain
        max_delay_bin = max(dk_MicGrid);
        min_delay_bin = min(dk_MicGrid);

        orig_signal_len = sample_range_high-sample_range_low+1;
        signal_len = orig_signal_len + abs(floor(min_delay_bin))
+ ceil(max_delay_bin);
        semiDelayed_MicSignals = zeros(num_mic, signal_len);
        for i = 1:num_mic;

```

```

        %delayed_signal = recd_signal[k + dk]
        delayed_start_i = floor(max_delay_bin) + 2; %This
is the start index for the non-delayed signal.
        if(dk_MicGrid(i) > 0)
            delayed_start_i = floor(max_delay_bin) + 2 -
floor(dk_MicGrid(i));
            dk_MicGrid(i) = dk_MicGrid(i) -
floor(dk_MicGrid(i)); %The fractal part.
        else
            delayed_start_i = floor(max_delay_bin) + 2 +
abs(ceil(dk_MicGrid(i)));
            dk_MicGrid(i) = dk_MicGrid(i) -
ceil(dk_MicGrid(i)); %The fractal part.
        end;

        semiDelayed_MicSignals(i,
delayed_start_i:(delayed_start_i + orig_signal_len - 1)) =
mics_Received(i,:);
        end;

        Nfft = signal_len;
        beamformed_window_freqdom = zeros(1, Nfft);
        %i is the iterator for microphone indexes
        for i = 1:num_mic;

            delayed_window_freqdom =
fft(semiDelayed_MicSignals(i,:),Nfft);

            % %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))
            % %positive frequencies (W(th)*e^(-j*th*dt))
            % for k = 2:ceil(Nfft/2);
            %
            % delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(-j * (k-1) * (2*pi/(Nfft-1)) *
dk_MicGrid(i));
            % end
            % %negative frequencies (W(th)*e^(j*th*dt))
            % for k = ceil(Nfft/2)+1:Nfft;
            %
            % delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(j * (Nfft-k+1) * (2*pi/(Nfft-1)) *
dk_MicGrid(i));
            % end

            % beamformed_window_freqdom =
beamformed_window_freqdom + delayed_window_freqdom;

```

```

        freq_limit = 8000;
        iterator_limit = (freq_limit / (sampling_freq /
Nfft)) + 1;
        %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))
        %positive frequencies (W(th)*e^(-j*th*dt)
        for k = 2:ceil(iterator_limit);
            delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(-j * (k-1) * (2*pi/(Nfft-1)) *
dk_MicGrid(i));
        end
        %negative frequencies (W(th)*e^(j*th*dt))
        for k = Nfft-ceil(iterator_limit):Nfft;
            delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(j * (Nfft-k+1) * (2*pi/(Nfft-1)) *
dk_MicGrid(i));
        end
        beamformed_window_freqdom =
beamformed_window_freqdom + delayed_window_freqdom;

        end;
        beamformed_signal =
real(ifft(beamformed_window_freqdom/num_mic, Nfft));

        test_plane_amp(w_i,d_i) =
norm(beamformed_signal)/sqrt(length(beamformed_signal));
        test_plane_amp_db = 10*log10(test_plane_amp); %1 V
méréshatárra normált, ezért nem kell osztani 1-gyel :-))
        end
        toc
    end

    %min_val = min(min(test_plane_sense));
    min_val = 0;
    max_val = max(max(test_plane_amp));

    test_plane_amp_colour = floor((test_plane_amp - min_val) /
((max_val-min_val)/(colormap_len-1))) + 1;

    figureHandler = figure('Name',sprintf('meres%d',n_meres));
    colormap(jet(colormap_len));
    image(test_plane_amp_colour);

```

```

min_val = min(min(test_plane_amp_db));
max_val = 0;

test_plane_amp_colour = floor((test_plane_amp_db - min_val) /
((max_val-min_val)/(colormap_len-1))) + 1;

figureHandler = figure('Name',sprintf('meres%d',n_meres));
colormap(jet(colormap_len));
image(test_plane_amp_colour);

```

## FreqLocation.m

```

c = 344; %sonic speed
micGrid = CrossGrid15([0 0 2]);

n_meres = 1; %Meres sorszama
mon_freq = 1585; %Tesztelt frekvencia

sample_range_low = 1.5*10^5;
sample_range_high = sample_range_low+4095;
interpol_rate = 1;
leakage_toler = 0.1;

load('test.mat');

mics_Received = data(sample_range_low:sample_range_high,:);
sampling_freq = Fs;
% for i = 1:8
% [mics_Received(i,:), sampling_freq, nbits] =
wavread(sprintf('%s%d%s%d%s%d%s', 'Input\m', n_meres, '_', mon_freq, '-
', i, '.wav'), [sample_range_low sample_range_high]);
% end;

grid_res = 0.04; %test grid resolution in meter
width_border_half_dist = 1.5; %in meter
depth_border_dist = 5; %in meter
colormap_len = 256;

num_mic = size(mics_Received,1);
centre = sum(micGrid) / num_mic; %test centre coordinates

```

```

Z_offset = 0;
centre(3) = centre(3) + Z_offset;

test_plane_amp = zeros((2*width_border_half_dist/grid_res)+1,
(depth_border_dist/grid_res));

d_i=0;

for d_coord = 0 : grid_res : centre(2) + depth_border_dist
    tic
    d_i=d_i+1;
    w_i=0;
    for w_coord = (centre(1) - width_border_half_dist) :
grid_res : (centre(1) + width_border_half_dist)
        w_i=w_i+1;
        focus = [w_coord d_coord centre(3)];

        %calculate the delay
        dk_MicGrid = zeros(num_mic,1);
        for i = 1:num_mic
            dk_MicGrid(i) = (norm(micGrid(i,:) - focus))/c;
        end
        %dk_MicGrid = (dk_MicGrid - dk_MicGrid(1)) *
sampling_freq;
        dk_MicGrid = (dk_MicGrid - dk_MicGrid(4)) *
sampling_freq;

        %delay the non-fractal (whole) bins in the time domain
        max_delay_bin = max(dk_MicGrid);
        min_delay_bin = min(dk_MicGrid);
        orig_signal_len = sample_range_high-sample_range_low+1;
        signal_len = orig_signal_len + abs(floor(min_delay_bin))
+ ceil(max_delay_bin);
        semiDelayed_MicSignals = zeros(num_mic, signal_len);
        for i = 1:num_mic;
            %delayed_signal = recd_signal[k + dk]
            delayed_start_i = floor(max_delay_bin) + 2; %This
is the start index for the non-delayed signal.
            if(dk_MicGrid(i) > 0)
                delayed_start_i = floor(max_delay_bin) + 2 -
floor(dk_MicGrid(i));
                dk_MicGrid(i) = dk_MicGrid(i) -
floor(dk_MicGrid(i)); %The fractal part.

```



```

else
    delayed_start_i = floor(max_delay_bin) + 2 +
abs(ceil(dk_MicGrid(i)));
    dk_MicGrid(i) = dk_MicGrid(i) -
ceil(dk_MicGrid(i)); %The fractal part.
end;

    semiDelayed_MicSignals(i,
delayed_start_i:(delayed_start_i + orig_signal_len - 1)) =
mics_Received(i,:);
end;

    Nfft = signal_len*interp_rate;
    beamformed_window_freqdom = zeros(1, Nfft);
    %i is the iterator for microphone indexes
    for i = 1:num_mic;

        semiDelayed_signal =
fft(semiDelayed_MicSignals(i,:),Nfft);
        delayed_window_freqdom = zeros(1, Nfft);

        start_i = floor((mon_freq*(1-leakage_toler) * (Nfft-
1) / sampling_freq)+1);
        stop_i = ceil((mon_freq*(1+leakage_toler) * (Nfft-1)
/ sampling_freq)+1);

        %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))
        %positive frequencies (W(th)*e^(-j*th*dt))
        for k = start_i:stop_i;
            delayed_window_freqdom(k) =
semiDelayed_signal(k) * exp(-j * (k-1) * (2*pi/(Nfft-1)) *
dk_MicGrid(i));
        end
        %negative frequencies (W(th)*e^(j*th*dt))
        for k = Nfft-stop_i+1:Nfft-start_i+1;
            delayed_window_freqdom(k) =
semiDelayed_signal(k) * exp(j * (Nfft-k+1) * (2*pi/(Nfft-1)) *
dk_MicGrid(i));
        end
        beamformed_window_freqdom =
beamformed_window_freqdom + delayed_window_freqdom;
    end;

    beamformed_signal =
real(ifft(beamformed_window_freqdom/num_mic, Nfft));

```

```

        test_plane_amp(w_i,d_i) =
norm(beamformed_signal)/sqrt(length(beamformed_signal));

        test_plane_amp_db = 10*log10(test_plane_amp);      %1 V
méréshatárra normált, ezért nem kell osztani 1-gyel :-))

    end
    toc
end

%min_val = min(min(test_plane_sense));
min_val = 0;
max_val = max(max(test_plane_amp));

test_plane_amp_colour = floor((test_plane_amp - min_val) /
((max_val-min_val)/(colormap_len-1))) + 1;

figureHandler = figure('Name',sprintf('meres%d%s%dHz',n_meres,'
freq=',mon_freq));
colormap(jet(colormap_len));
image(test_plane_amp_colour);

min_val = min(min(test_plane_amp_db));
max_val = 0;

test_plane_amp_colour = floor((test_plane_amp_db - min_val) /
((max_val-min_val)/(colormap_len-1))) + 1;

figureHandler = figure('Name',sprintf('meres%d%s%dHz',n_meres,'
freq=',mon_freq,' dB'));
colormap(jet(colormap_len));
image(test_plane_amp_colour);

min_val = min(min(test_plane_amp_db));
max_val = max(max(test_plane_amp_db));

test_plane_amp_colour = floor((test_plane_amp_db - min_val) /
((max_val-min_val)/(colormap_len-1))) + 1;

figureHandler = figure('Name',sprintf('meres%d%s%dHz',n_meres,'
freq=',mon_freq,' dB'));
colormap(jet(colormap_len));
image(test_plane_amp_colour);

```

## SourceLocation\_VerticalPlane.m

```
c = 344; %sonic speed
micGrid = CrossGrid15([0 0 2]);

n_meres = 1; %Meres sorszama
%mon_freq = 520; %Tesztelt frekvencia

sample_range_low = 1.5*10^5;
sample_range_high = sample_range_low+4095;

% mics_Received = zeros(8,length(sample_range_low :
sample_range_high));

load('test.mat');

mics_Received = data(sample_range_low:sample_range_high,:);
sampling_freq = Fs;

% for i = 1:8
% [mics_Received(i,:), sampling_freq, nbits] =
wavread(sprintf('%s%d%s%d%s%d%s', 'Input\m', n_meres, '_', freq, '_emberiha
ng-', i, '.wav'), [sample_range_low sample_range_high]);
% [mics_Received(i,:), sampling_freq, nbits] =
wavread(sprintf('%s%d%s%d%s%d%s', 'Input\m', n_meres, '_', freq, '-
', i, '.wav'), [sample_range_low sample_range_high]);
% end;

grid_res = 0.05; %test grid resolution in meter
width_border_half_dist = 3.5; %in meter
hight_border_half_dist = 2; %in meter
colormap_len = 256;

num_mic = size(mics_Received,1);
centre = sum(micGrid) / num_mic; %test centre coordinates
depth_offset = 8;
centre(2) = centre(2) + depth_offset;

test_plane_amp = zeros((2*width_border_half_dist/grid_res)+1,
(2*hight_border_half_dist/grid_res)+1);
```

```

h_i=0;

for h_coord = centre(3) - hight_border_half_dist : grid_res :
centre(3) + hight_border_half_dist
    h_i=h_i+1;
    w_i=0;
    for w_coord = (centre(1) - width_border_half_dist) :
grid_res : (centre(1) + width_border_half_dist)
        w_i=w_i+1;
        focus = [w_coord centre(2) h_coord];

        %calculate the delay
        dk_MicGrid = zeros(num_mic,1);
        for i = 1:num_mic
            dk_MicGrid(i) = (norm(micGrid(i,:) - focus))/c;
        end
        dk_MicGrid = (dk_MicGrid - dk_MicGrid(1)) *
sampling_freq;

        %delay the non-fractal (whole) bins in the time domain
        max_delay_bin = max(dk_MicGrid);
        min_delay_bin = min(dk_MicGrid);

        orig_signal_len = sample_range_high-sample_range_low+1;
        signal_len = orig_signal_len + abs(floor(min_delay_bin))
+ ceil(max_delay_bin);
        semiDelayed_MicSignals = zeros(num_mic, signal_len);
        for i = 1:num_mic;
            %delayed_signal = recd_signal[k + dk]
            delayed_start_i = floor(max_delay_bin) + 2; %This
is the start index for the non-delayed signal.
            if(dk_MicGrid(i) > 0)
                delayed_start_i = floor(max_delay_bin) + 2 -
floor(dk_MicGrid(i));
                dk_MicGrid(i) = dk_MicGrid(i) -
floor(dk_MicGrid(i)); %The fractal part.
            else
                delayed_start_i = floor(max_delay_bin) + 2 +
abs(ceil(dk_MicGrid(i)));
                dk_MicGrid(i) = dk_MicGrid(i) -
ceil(dk_MicGrid(i)); %The fractal part.
            end;
        end;

```

```

        semiDelayed_MicSignals(i,
delayed_start_i:(delayed_start_i + orig_signal_len - 1)) =
mics_Received(i,:);
        end;

        Nfft = signal_len;
        beamformed_window_freqdom = zeros(1, Nfft);
        %i is the iterator for microphone indexes
        for i = 1:num_mic;

                delayed window freqdom =
fft(semiDelayed_MicSignals(i,:),Nfft);

                %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))
                %positive frequencies (W(th)*e^(-j*th*dt))
                for k = 2:ceil(Nfft/2);
                        delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(-j * (k-1) * (2*pi/(Nfft-1))) *
dk_MicGrid(i));
                end
                %negative frequencies (W(th)*e^(j*th*dt))
                for k = ceil(Nfft/2)+1:Nfft;
                        delayed_window_freqdom(k) =
delayed_window_freqdom(k) * exp(j * (Nfft-k+1) * (2*pi/(Nfft-1))) *
dk_MicGrid(i));
                end
                beamformed_window_freqdom =
beamformed_window_freqdom + delayed_window_freqdom;
                end;
                beamformed_signal =
real(ifft(beamformed_window_freqdom/num_mic, Nfft));

                test_plane_amp(h_i,w_i) =
norm(beamformed_signal)/sqrt(length(beamformed_signal));
                end
        end

        %min_val = min(min(test_plane_sense));
        % min_val = 0;
        % max_val = max(max(test_plane_amp));
        %
        % test_plane_amp_colour = floor((test_plane_amp - min_val) /
((max_val-min_val)/(colormap_len-1))) + 1;

```

```

%
% figureHandler = figure('Name',sprintf('meres%d',n_meres));
% colormap(jet(colormap_len));
% image(test_plane_amp_colour);

figure;
%test_plane_amp(1,1) = 0; % dummy value for scaling
handler = image(test_plane_amp);
set(handler,'CDataMapping','scaled');
cb = colorbar;
set(cb,'YTick',0:128:2048);
%colormap(jet(128*128));

```

### **FreqLocation\_VerticalPlane.m**

```

c = 344; %sonic speed
micGrid = CrossGrid15([0 0 2]);

n_meres = 1; %Meres sorszama
mon_freq = 455; %Tesztelt frekvencia

sample_range_low = 1.5*10^5;
sample_range_high = sample_range_low+4095;
interpol_rate = 1;
leakage_toler = 0.1;

load('test.mat');

mics_Received = data(sample_range_low:sample_range_high,:);
sampling_freq = Fs;
% for i = 1:8
% [mics_Received(i,:), sampling_freq, nbits] =
wavread(sprintf('%s%d%s%d%s%d%s','Input\m',n_meres,'_',mon_freq,'-
',i,'.wav'), [sample_range_low sample_range_high]);
% end;

grid_res = 0.05; %test grid resolution in meter
width_border_half_dist = 2; %in meter
hight_border_half_dist = 2; %in meter
colormap_len = 256;

```

```

num_mic = size(mics_Received,1);
centre = sum(micGrid) / num_mic; %test centre coordinates
depth_offset = 3.6;
centre(2) = centre(2) + depth_offset;

test_plane_amp = zeros((2*width_border_half_dist/grid_res)+1,
(2*hight_border_half_dist/grid_res)+1);

h_i=0;

for h_coord = centre(3) - hight_border_half_dist : grid_res :
centre(3) + hight_border_half_dist
tic
h_i=h_i+1;
w_i=0;
for w_coord = (centre(1) - width_border_half_dist) :
grid_res : (centre(1) + width_border_half_dist)
w_i=w_i+1;
focus = [w_coord centre(2) h_coord];

%calculate the delay
dk_MicGrid = zeros(num_mic,1);
for i = 1:num_mic
dk_MicGrid(i) = (norm(micGrid(i,:) - focus))/c;
end
%dk_MicGrid = (dk_MicGrid - dk_MicGrid(1)) *
sampling_freq;
dk_MicGrid = (dk_MicGrid - dk_MicGrid(4)) *
sampling_freq;

%delay the non-fractal (whole) bins in the time domain
max_delay_bin = max(dk_MicGrid);
min_delay_bin = min(dk_MicGrid);

orig_signal_len = sample_range_high-sample_range_low+1;
signal_len = orig_signal_len + abs(floor(min_delay_bin))
+ ceil(max_delay_bin);
semiDelayed_MicSignals = zeros(num_mic, signal_len);
for i = 1:num_mic;
%delayed_signal = recd_signal[k + dk]
delayed_start_i = floor(max_delay_bin) + 2; %This
is the start index for the non-delayed signal.

```

```

        if(dk_MicGrid(i) > 0)
            delayed_start_i = floor(max_delay_bin) + 2 -
floor(dk_MicGrid(i));
            dk_MicGrid(i) = dk_MicGrid(i) -
floor(dk_MicGrid(i)); %The fractal part.
        else
            delayed_start_i = floor(max_delay_bin) + 2 +
abs(ceil(dk_MicGrid(i)));
            dk_MicGrid(i) = dk_MicGrid(i) -
ceil(dk_MicGrid(i)); %The fractal part.
        end;

        semiDelayed_MicSignals(i,
delayed_start_i:(delayed_start_i + orig_signal_len - 1)) =
mics_Received(i,:);
        end;

        Nfft = signal_len*interpol_rate;
        beamformed_window_freqdom = zeros(1, Nfft);
        %i is the iterator for microphone indexes
        for i = 1:num_mic;

            semiDelayed_signal =
fft(semiDelayed_MicSignals(i,:),Nfft);
            delayed_window_freqdom = zeros(1, Nfft);

            start_i = floor((mon_freq*(1-leakage_toler) * (Nfft-
1) / sampling_freq)+1);
            stop_i = ceil((mon_freq*(1+leakage_toler) * (Nfft-1)
/ sampling_freq)+1);

            %k is the iterator for temporal frequencies (th =
2*pi/Nfft * (k-1))

            %positive frequencies (W(th)*e^(-j*th*dt))
            for k = start_i:stop_i;
                delayed_window_freqdom(k) =
semiDelayed_signal(k) * exp(-j * (k-1) * (2*pi/(Nfft-1))) *
dk_MicGrid(i);
            end
            %negative frequencies (W(th)*e^(j*th*dt))
            for k = Nfft-stop_i+1:Nfft-start_i+1;

```



```

                                delayed_window_freqdom(k) =
semiDelayed_signal(k) * exp(j * (Nfft-k+1) * (2*pi/(Nfft-1)) *
dk_MicGrid(i));

                                end

                                beamformed_window_freqdom =
beamformed_window_freqdom + delayed_window_freqdom;

                                end;

                                beamformed_signal =
real(iff(beamformed_window_freqdom/num_mic, Nfft));

                                test_plane_amp(h i,w i) =
norm(beamformed_signal)/sqrt(length(beamformed_signal));

                                end

                                toc

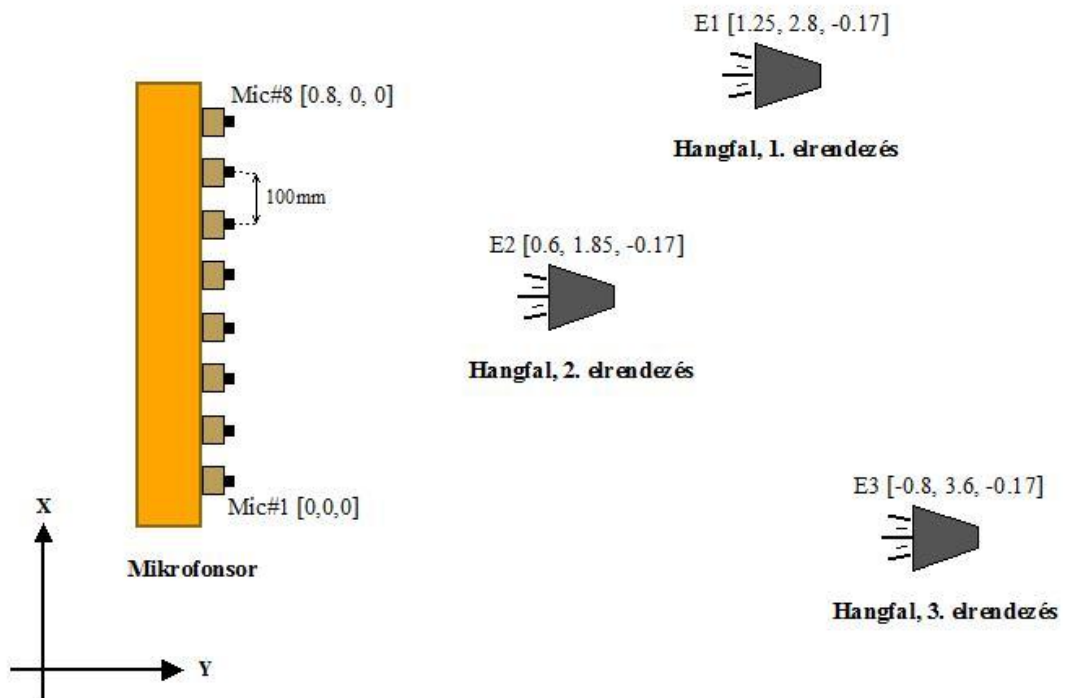
                                end

                                %min_val = min(min(test_plane_sense));
                                % min_val = 0;
                                % max_val = max(max(test_plane_amp));
                                %
                                % test_plane_amp_colour = floor((test_plane_amp - min_val) /
((max_val-min_val)/(colormap_len-1))) + 1;
                                %
                                % figureHandler = figure('Name',sprintf('meres%d',n_meres));
                                % colormap(jet(colormap_len));
                                % image(test_plane_amp_colour);

                                figure;
                                %test_plane_amp(1,1) = 0; % dummy value for scaling
                                handler = image(test_plane_amp);
                                set(handler,'CDataMapping','scaled');
                                cb = colorbar;
                                set(cb,'YTick',0:128:2048);
                                %colormap(jet(128*128));

```

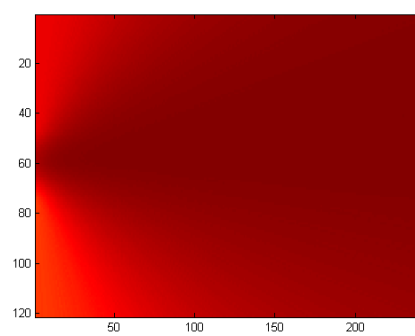
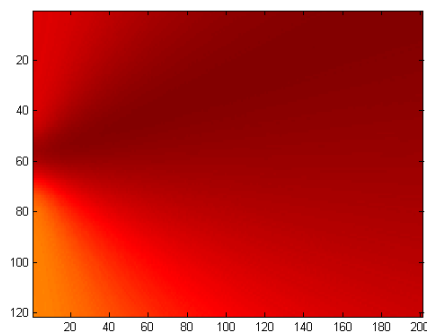
## 2. Melléklet Süketszobás mérések eredményei



4.8. ábra: Süketszobás mérés – elrendezés

Szimulált

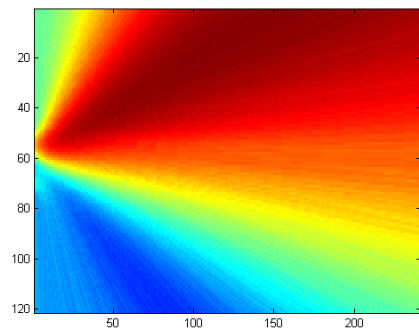
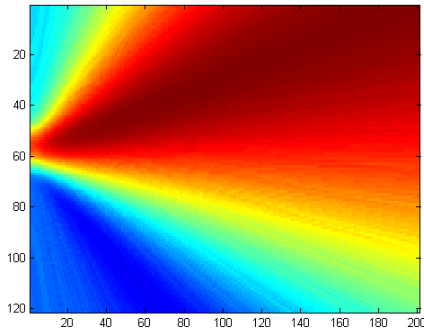
Mért, valós



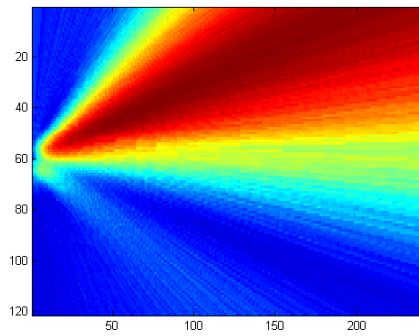
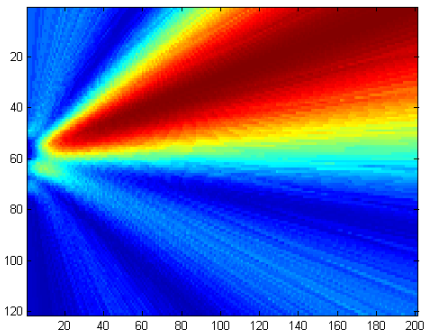
E3 elrendezés – 125 Hz-es eredmény

**Szimulált**

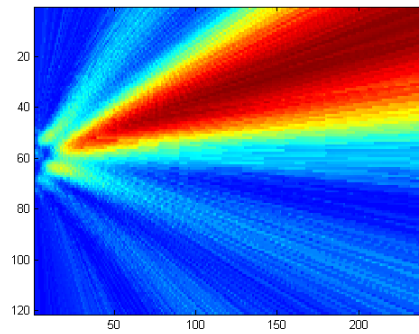
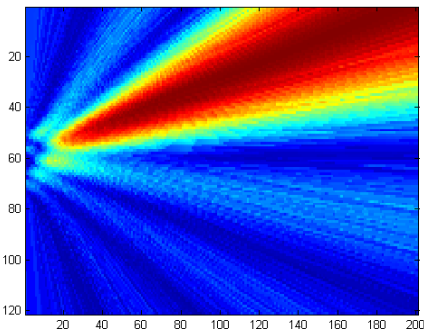
**Mért, valós**



**E3 elrendezés – 455 Hz-es eredmény**



**E3 elrendezés – 1000 Hz-es eredmény**



**E3 elrendezés – 1500 Hz-es eredmény**