

KECSKEMÉTI FŐISKOLA
GÉPIPARI ÉS AUTOMATIZÁLÁSI MŰSZAKI FŐISKOLAI KAR
INFORMATIKA TANSZÉK

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

Akusztikus lokalizáció

Készítette:

Dömötör László

II. évf. mérnök informatika szak

Konzulens:

Dömötör Zénó

tanszéki mérnök

KECSKEMÉT

2012

Tartalom

1. Bevezetés	3
1.1. Motiváció.....	3
1.2. Elért eredmények.....	3
1.3. A LabVIEW-ről általában.....	3
1.4. LabVIEW fejlesztőkörnyezet.....	4
1.4.1. Front panel, blokk diagram	4
1.4.2. Modularitás (subVI).....	5
1.4.3. Struktúrák.....	7
1.4.5. Egyebek	9
2. Hardvereszközök	10
2.1. Mérőkártya.....	10
2.2. Mikrofonerősítő	10
2.3. Fizikai kialakítás, elrendezés bemutatása	11
3. Szoftverfejlesztés	11
3.1. Számítás levezetése.....	13
4. Próbamérések	14
4.1. 1. próbamérés	14
4.2. 2. próbamérés	17
4.3. 3. próbamérés	18
4.4. 4. próbamérés	20
4.5. Mérési pontatlanság hatása és a multiplexelt bemenetből fakadó pontatlanság.....	22
4.6. Valós mérési eredmény.....	23
5. A LabVIEW program bemutatása	25
6. Összefoglalás	26
6.1. Eredmények értékelése.....	26
6.2. Tervek a jövőre nézve	26
7. Felhasznált szakirodalom	27

1. Bevezetés

1.1. Motiváció

Napjainkban egyre nagyobb a robotok térhódítása. Az autonóm módon működő robotok számos szenzort tartalmazhatnak, úgy mint fény-, hang-, távolság- stb. érzékelő. A dolgozatban bemutatott rendszer egy tetszőleges hangforrás helyét határozza meg, így egyike lehet az adott robotra felkerülő szenzoroknak. A hangforrás helyének detektálásával a robot akár emberi hanggal irányítható, oly módon, hogy utasításokat ad az irányító a robotnak arra nézve, hogy hozzá képest milyen irányba és mekkora utat kell megtenni.

Az általam készített eszköz a hangforrás helyét határozza meg. Ennek értelmében számos más felhasználási területe is lehet. Születtek már hasonló feladatot megvalósító eszközök, viszont a megvalósítás elvében vannak eltérések. E dolgozat célja bemutatni az eszközt és működését.

1.2. Elért eredmények

Az eddig elkészült eszköz három mikrofonból, három erősítőből és egy National Instruments által gyártott mérésadatgyűjtő kártyából áll. Ehhez LabView-ban készült egy program, mely a beérkező jelek alapján kiszámolja az x, y koordinátákat. Két szenzor esetén a mikrofonokkal egy vonalban lévő hangforrás helyét lokalizálja az eszköz, 1 cm nagyságrendű pontossággal, valamint meghatározza, hogy melyik oldalról jött a hang. Három mikrofon esetén síkban határozza meg a hangforrás helyét.

1.3. A LabVIEW-ről általában

A mérés és automatizálás témakörben élen járó National Instruments Corporation által kifejlesztett LabVIEW elődje virtuális műszert kezelni és létrehozni képes szoftver volt. Folyamatos fejlesztés során a LabVIEW számos eszközzel bővült, mára a mérnökök számára mérési, automatizálási, folyamatirányítási, szimulációs és tesztelési célokra egyaránt használható. A programozás során virtuális műszereket (Virtual Instruments – VI) hozunk létre. A létrehozott virtuális műszerek mind az iparban, mind pedig az oktatásban eredményesen használhatóak.

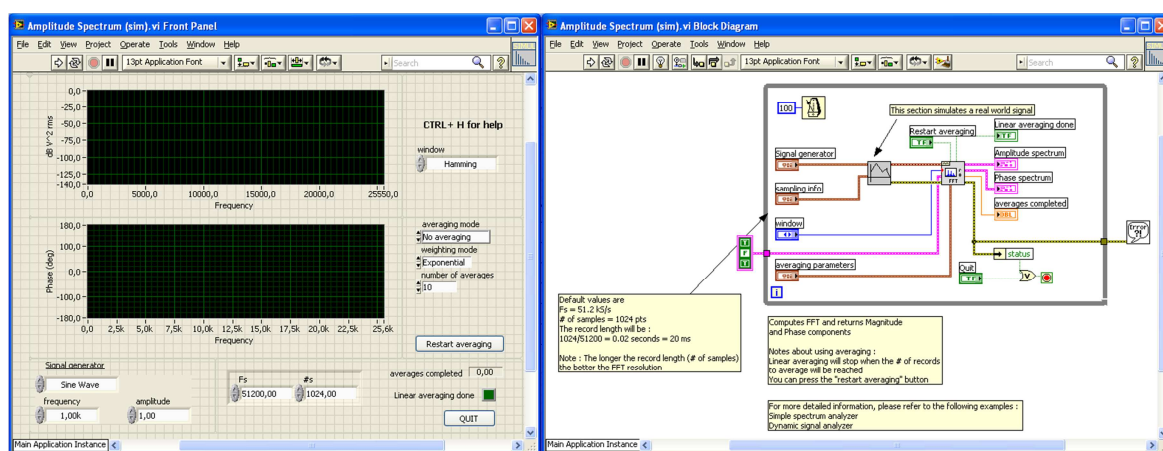
A virtuális műszerek kialakulását és elterjedését a személyi számítógépek rohamos fejlődése tette lehetővé. Használatuk során erős és viszonylag olcsó hardvert kapunk, (az

egyre alacsonyabb ár mellett, egyre nagyobb teljesítményre képes PC-k segítették elő az ilyen rendszerek elterjedését és fejlődését). A PC – megfelelő mérőkártya és szoftver társítása után – a mérnöki világban szükséges műszerként használható. A virtuális műszer használata közben jelenlévő PC könnyíti a kommunikációt, naplózást, elemzést, stb. A technológia sajátossága a testre szabható mérőműszer. A széles körben történő elterjedésének kritériuma a mérőkártyák szabványos illesztő felülete.

1.4. LabVIEW fejlesztőkörnyezet

A LabVIEW grafikus programozási környezet, amely adatfolyam elven, G-nyelven programozható. A programfejlesztés front panel (előlap) és blokk diagram (grafikus programot tartalmazó rész – kvázi forráskód) segítségével történik [1], [2].

A LabVIEW-val létrehozott program a VI (virtual instruments – virtuális műszer). A VI három komponensből áll, melyek a következők: az előlap (front panel), blokk diagram (block diagram), ikon és konnektor (icon and connector pane). A 1. ábra bal oldalán látható a front panel, melyen megfigyelhetők a kontrollok és indikátorok, az 1. ábra jobb oldalán pedig a blokk diagram, amely a „forráskódot” tartalmazza.



1. ábra. LabVIEW front panelja és blokk diagramja (Forrás: LabVIEW/NI Example Finder)

1.4.1. Front panel, blokk diagram

Az előlap nem más, mint a mérést végző virtuális műszer kezelőfelülete, ezen helyezhetjük el a felhasználó által testre szabható virtuális be- és kimeneti perifériákat, más szóval a front panelbeli elemeket. A elemek nevei a szoftverben kontrol (control) és indikátor (indicator). A kontrol lehet nyomógomb, kapcsoló, csúszka, forgatógomb, szám

vagy szöveg bevitelére alkalmas mező és egyéb bemeneti eszköz. Az indikátor lehet LED, grafikon, szám és szöveg kijelzésére alkalmas mező és egyéb kimeneti eszköz.

Az előlaphoz tartozik egy blokk diagram, amely az előlapon szereplő elemek működését szabja meg. Ez tartalmazza a forrásnyelvi leírást, amely G nyelven készíthető el. A blokk diagramon megjelennek az előlaphoz helyezett elemek szimbólumai, melyek segítségével programozhatjuk az ezek közötti kapcsolatot. Függvényeket, struktúrákat, aritmetikai és logikai műveleteket és egyéb eszközöket használhatunk programunk elkészítéséhez. A 1. ábrán látható front panelen (bal oldali panel) megfigyelhetők a kontrollok és indikátorok, míg a blokk diagramon (jobb oldali panel) a forrásnyelvi leírás.

1.4.2. Modularitás (subVI)



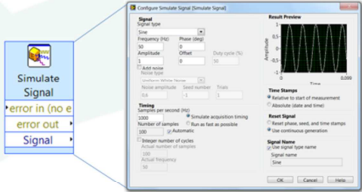
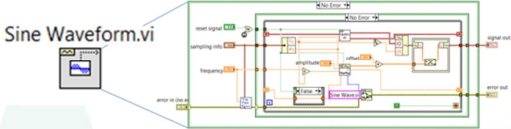
Express VI-ok, VI-ok és függvények



Express VI-ok: dialógusablakban konfigurálható interaktív VI-ok

SubVI-ok: huzalozással testreszabható, modularizált VI-ok

Függvények: a LabVIEW alapvető elemei, nincs előlapjuk és blokkdiagramjuk

Add


Subtract


Multiply


Divide

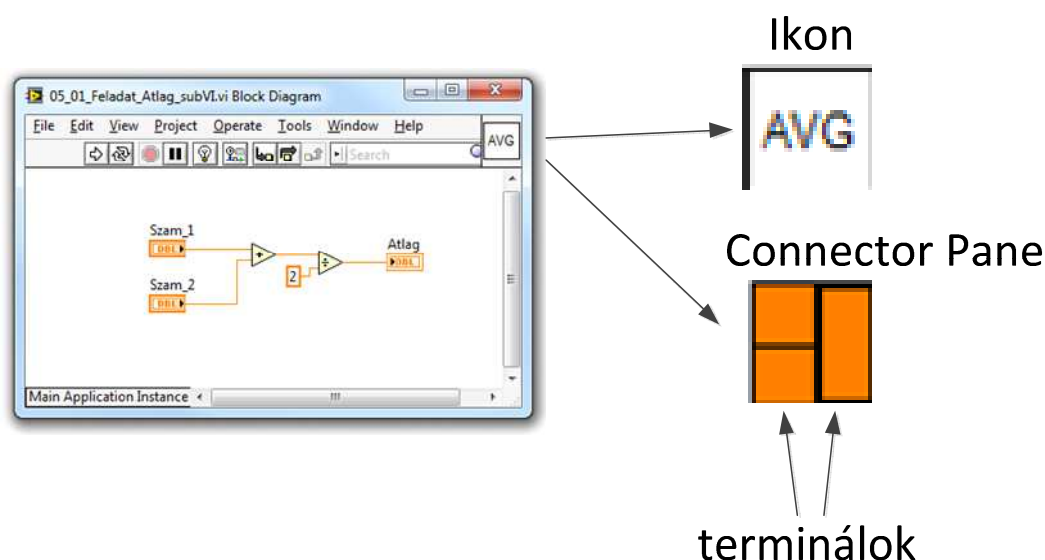

4/28
4. File I/O

A fenti műveletek hierarchia szintjüket tekintve lehetnek magas és alacsony szintűek. A magas szintű műveletek az expressz VI-ok. Komplet feladat ellátására készültek, ezért könnyíti a programozó feladatát, viszont csak bizonyos célfeladatok ellátására alkalmasak. Vegyük a fájlkezelő műveleteket példaként. A magas szintű fájlművelet teljesíti a fájlkezelésnél előforduló mindhárom fájlműveletet, jelesül a fájlmegnyitást, fájlolvasást-írást és fájllezárást¹.

¹ Negyedik műveletként a magas szintű fájlművelet elvégzi a hibakezelést.

Az alacsonyszintű műveletek a függvények és a subVI-ok². Ezeket a műveleteket használva a programozó alapelemekből építkezhet. Meghatározhatók az egyéni feladatrészek működése, rugalmasságot biztosít a programozásban, viszont nagyobb programozói tudást igényel. Fontos szem előtt tartani, hogy az alacsonyszintű műveletek hatékonyabbak és eredményesebbek, mint a magas szintűek. A fájlkezelő eszköztárat példaként tekintve a fájlmegnyitás vagy létrehozás, fájl bezárás, stb. művelet tartozik ebbe a kategóriába.

A subVI helyes használatával a LabVIEW-ban megírt program modulárisrá tehető. A subVI-ok úgy működnek, mint a konvencionális programnyelvek szubrutinjai vagy függvényei. Használatához szükség van ikonra és Connector Pane-re (kapcsolófelületre) (2. ábra.).



2. ábra. Ikon és Connector Pane

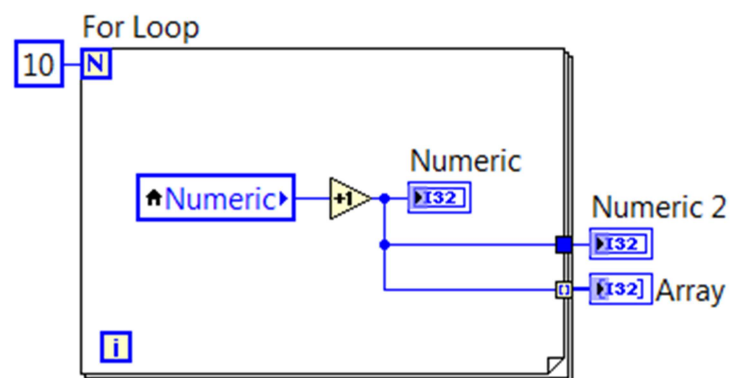
Az ikon a VI grafikus reprezentációja, ez jelképezi a VI-t más blokkdiagramokban. A subVI rendelkezik alapértelmezett ikonnal, viszont ennek használata kerülendő, mivel nem utal a subVI működésére (átláthatatlan programot eredményez). A Connector Pane a VI kapcsolófelülete, be- és kimeneti terminálokat tartalmaz. A VI a terminálokon keresztül adja át és kapja meg az adatokat más VI-októl.

² A subVI a magas- és az alacsonyszintű műveletek között helyezkedik el, mivel jelen írásban a műveleteket két szintre bontjuk, így a subVI az alacsonyszintű műveletek közé soroljuk.

1.4.3. Struktúrák

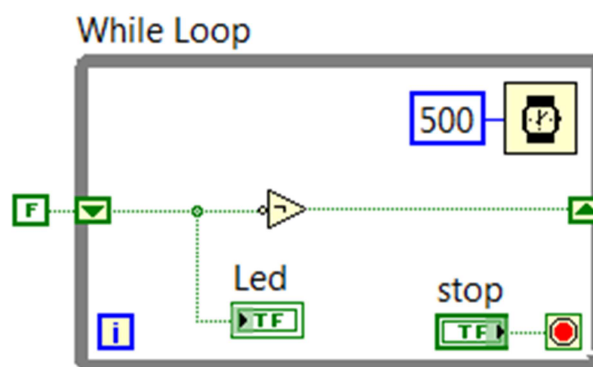
A LabVIEW-ban használhatunk struktúrákat, mint a legtöbb programozási nyelvben. A blokkdiagram funkció palettáján belül a struktúrák alpalettáján találhatók az elérhető struktúrák.

For ciklus (3. ábra): Elöltesztelős ciklus, az „N” bemeneti terminálba (Count Terminal) kötött numerikus adat határozza meg a lefutásra kerülő ciklusok számát. Rendelkezik ciklusszámlálóval (i, Iteration Terminal), amely 0 indexről indul. Az „N” bemeneti terminálba nullát kötve a ciklus egyszer sem fut le.



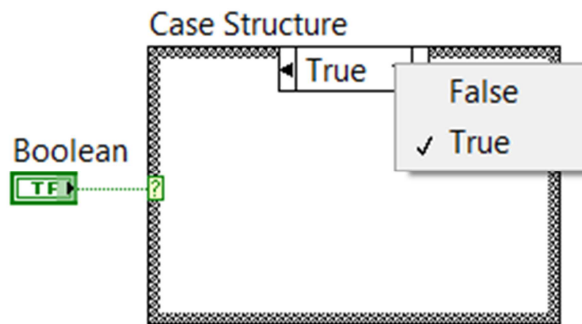
3. ábra. For Loop ciklus

While ciklus (4. ábra): Hátraltesztelős ciklus, így legalább egyszer lefut. A feltétel terminálba (Conditional Terminal) érkező logikai értéktől függ a futása (a 4. ábrán látható While ciklus a stop gomb megnyomásáig fut).



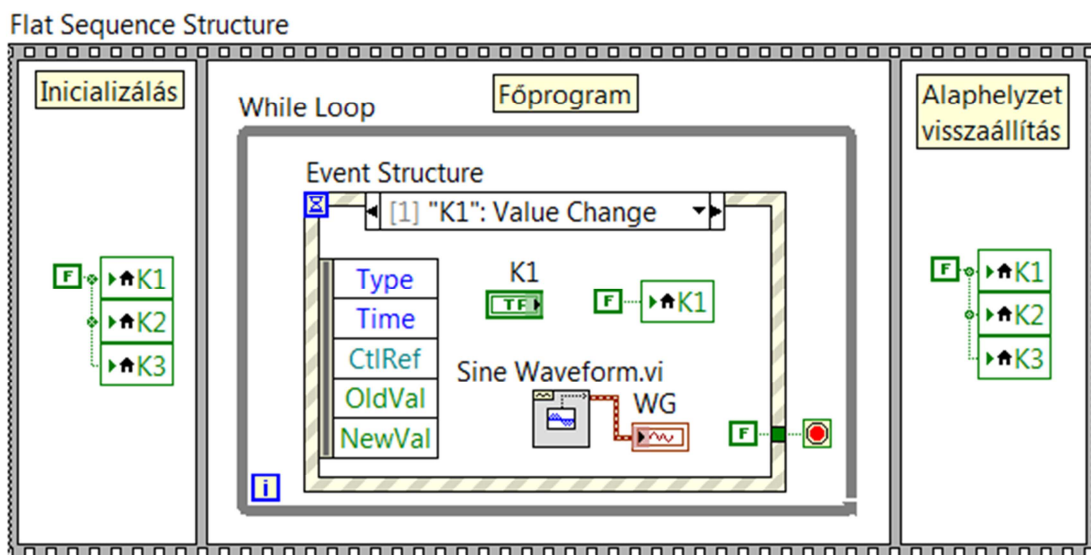
4. ábra. While Loop struktúra

Case struktúra (5. ábra): Elágazás – a Case Selectorba (kérdőjellel jelölt terminál) beérkező adat értékétől függ, hogy melyik Case ág fut le. A Selector³ felirata szerinti Case ág jeleníődik meg, a többi ág e mögött, rejtett formában van jelen. A Case struktúra egyszer fut le, (ha nincs egyéb struktúrába foglalva), oly módon, hogy mindenképpen lefut az egyik Case ága, mivel ha egyik feltétel sem teljesül, akkor a kötelezően kiválasztandó default ág kerül lefutásra.



5. ábra. Case struktúra

Flat Szekvencia Struktúra (6. ábra): A Flat szekvencia kereteket (Frame) tartalmaz, amely kereteken belül elhelyezett kódrészlet mindenképpen lefut és mindenképpen csak egyszer fut. A struktúrához új Frame-k adhatók a struktúra szegélyén történő jobb egérgomb kattintás segítségével.



6. ábra. Flat Szekvencia Struktúra

³ A case struktúra felső részének közepén helyezkedik el, segítségével kiválaszthatjuk az egyes Case ágakat.

A fenti struktúrák esetében létrehozhatók be- és kimeneti Tunnel-ek adatok adott struktúrába történő be- és kivezetése érdekében. A Tunnel a struktúra szegélyén négyzetként jelenik meg, használatára láthatunk egy példát a 3. ábrán, ahol a „Numeric 2” és „Array” nevű indikátorok Tunnel-eken keresztül kapják meg értékeiket. A ciklikus működésre képes struktúrák Tunnel-ei beállíthatók, hogy indexeljék⁴ az adatokat vagy sem⁵. A kimeneti Tunnel az őt tartalmazó struktúra lefutása után továbbítja az adatot.

A For és a While ciklus –, tehát a ciklikus működésre képes struktúrák – szegélyén Shift regiszter hozható létre. A Shift regiszter jobb és a bal oldali részből áll. Aktuális ciklusban a Shift regiszter jobb oldali termináljába adatot töltve, megjelenik az adat a következő ciklusban a Shift regiszter bal oldali termináljában.

1.4.5. Egyebek

A megírt VI-ok, subVI-ok, globális változók, stb. projektbe rendezhetők. A Getting Started ablakban az Empty Project menüpontot választva létrejön egy új és üres projekt. A felugró Project Explorer ablakban jelennek meg a projekthez hozzáadott alkotóelemek, itt hozzáadhatunk, eltávolíthatunk részeket.

A LabVIEW-ban használhatunk többféle változótípust, úgymint lokális változó, változóként használt, rejtett indikátor, globális változó, funkcionális globális változó, stb. A felsoroltak közül a globális változó használatakor feltétlen szükséges projektet létrehozni, mivel ez a változó egy speciális, LabVIEW által létrehozott könyvtárba kerül (a létrehozott könyvtár a PC fájlrendszerében egy fájlként látszik).

A tabulátor kontrol lapokból áll, amely lapok a front panelen jelennek meg. A lapokon elhelyezhetőek front panelbeli elemek. A tabulátor kontrol felső felén található Page Selector segítségével választhatjuk ki a megjeleníteni kívánt lapot. A nem látható lapon lévő frontpanelbeli elem futásidőben aktívan működik, általában a helytakarékoság, tagoltság biztosítására használjuk (elrejtünk bizonyos elemeket).

⁴ Ebben az esetben azt jelenti az indexelés, hogy a Tunnel-be beérkező adatokat ciklusonként tömbbe fűzi.

⁵ Ha a Tunnel indexelése nem engedélyezett, akkor a ciklusok lefutása után csak a legutolsó adat áll rendelkezésre, a többi pedig eldobásra kerül.

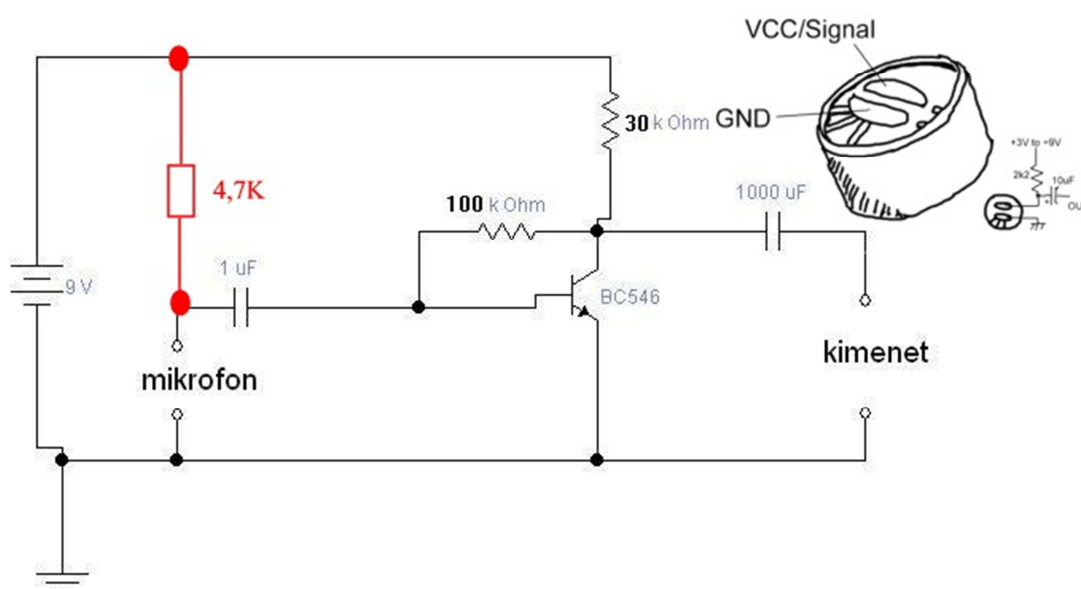
2. Hardvereszközök

2.1. Mérőkártya

A mérésadatgyűjtő kártya National Instruments cég által gyártott USB-6211 típusú eszközzel történik. Ez a multifunkcionális mérésadatgyűjtő kártya 16 analóg bemenettel rendelkezik (16-bit, 250kS/s), két analóg kimenettel (16-bit, 2500kS/s), valamint 4db digitális bemenet és 4db digitális kimenet található rajta. A mérőkártya USB-n keresztül kommunikál a PC-vel.

2.2. Mikrofonerősítő

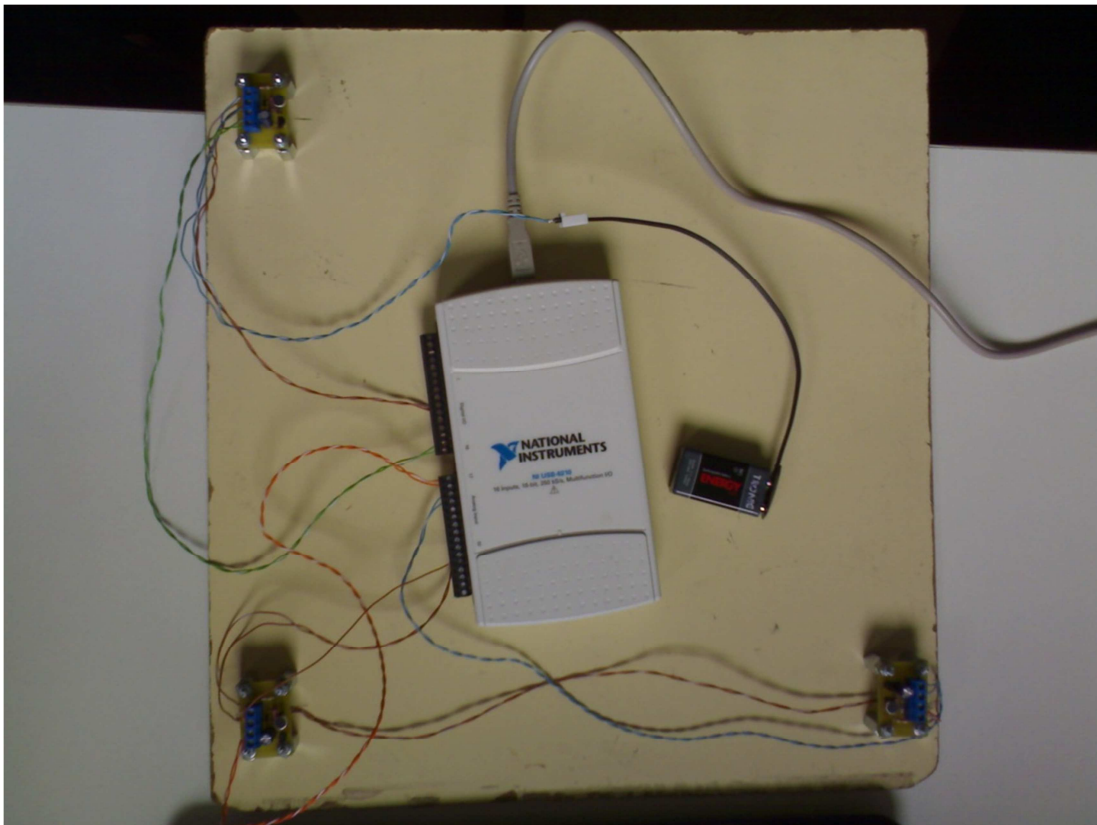
Mikrofonerősítőként egy egyszerű tranzisztoros erősítőkapcsolást használtam. Az alkalmazott erősítőkapcsolás impulzusszerű hangoknál (pl.: taps, csettintés) megfelelően használható. Beszédhang esetén - amely a TDK dolgozat továbbfejlesztési lehetőségei között szerepel - az egyes csatornákon érkező jelalakok között mintaillesztésre van szükség a lényeges információk (úthossz-különbségek)kinyerésének érdekében. Amennyiben a feldolgozandó hang beszédhang, úgy más megfontolások alapján kell építeni egy összetettebb erősítőkapcsolást, melynek paraméterei megfelelnek az akusztikában használatos paraméterekkel.



7. ábra. Mikrofonerősítő kapcsolási rajza

2.3. Fizikai kialakítás, elrendezés bemutatása

A mikrofonok tulajdonképpen egy koordináta rendszerre vannak ráillesztve oly módon, hogy az egyik az origóban a másik kettő pedig az x és y tengelyen helyezkednek el (pozitív irányban, bár ennek nincs jelentősége). Az x és y tengelyen elhelyezkedőknek az origótól mért távolságuk megegyezik.



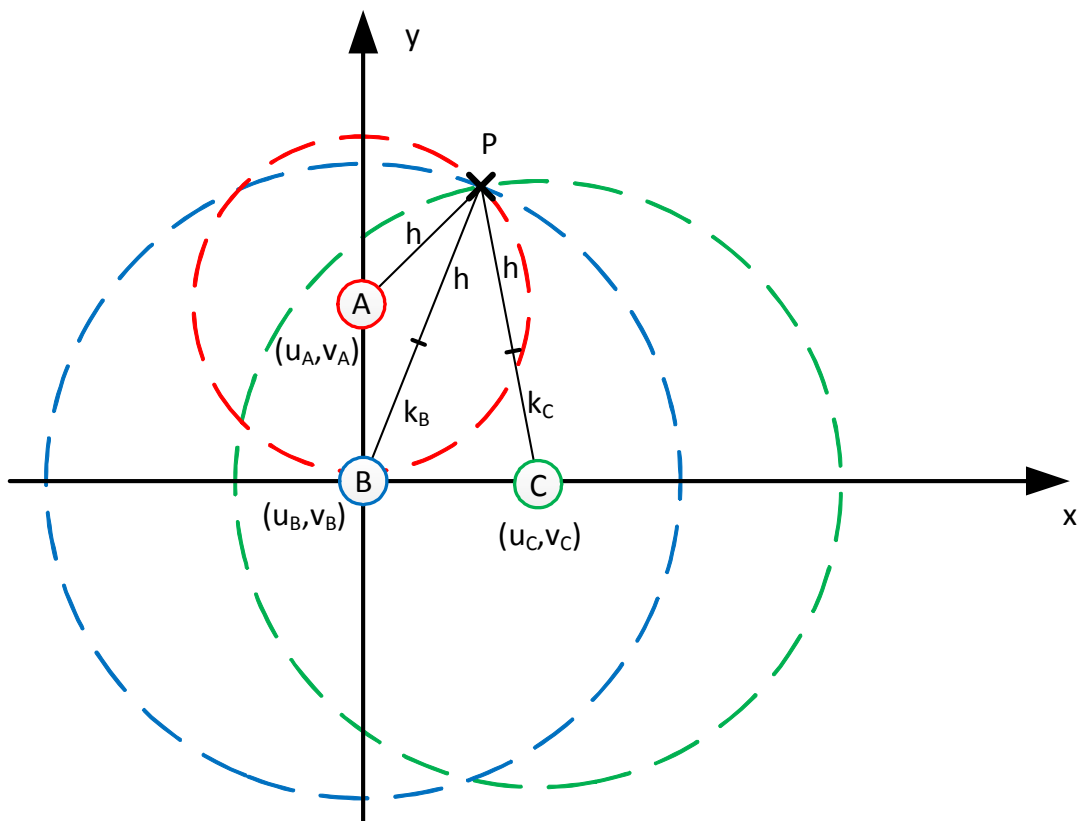
8. ábra. Az eszköz fizikai megvalósítása

3. Szoftverfejlesztés

A hangforrás helyének meghatározásához a következő elvet használtam. A beérkező hang hatására a hangforráshoz legközelebb eső szenzor triggerel, tehát ez számít 0 időpillanatnak. Ezáltal azt tudjuk meg, hogy mennyi idő alatt ér a hang az első szenzorhoz képest a másik kettőhöz. Így tulajdonképpen egy időkülönbséget kapunk, amit a hangsebesség birtokában át tudunk számolni útkülönbséggé. Minden szenzorból egy kört kezdünk növelni ugyanolyan mértékben, oly módon, hogy az egyes mikrofonokhoz

tartozó útkülönbséget vesszük a kör sugarának. Természetesen a triggerelő mikrofonnak 0 a sugara. A körök növelésének hatására lesz egy hármas metszéspont, ami a hangforrás helyének felel meg. Matematikai oldalról úgy történik a megvalósítása, hogy felírjuk a 3 kör köregyenletét és a hozzájuk tartozó sugarakat. Így kapunk hat egyenletet, amelyet levezetve megkapjuk az x, y koordináták képletét. Ezekbe a képletekbe behelyettesítve kapjuk meg a hangforrás helyét. Ezzel a módszerrel 3 mikrofon segítségével 2 dimenzióban lehet meghatározni a hangforrás helyét.

Felmerült a mérés pontatlanságából adódó esetleges exponenciális mértékben növekvő hiba. Sikerült mérésekkel igazolni, hogy az előbb említett hiba a hangforrás helyétől függően változhat. Ha a hangforrás egyvonalon van két szenzorral, akkor nem kapunk pontos eredményt. Ennek kiküszöbölése jövőbeni tervekben szerepel. A mérések a Matlab nevű program segítségével lettek elvégezve.



9. ábra. A számítás elvének meghatározása

3.1. Számítás levezetése

$$(x-0)^2 + (y-V_A)^2 = r_A^2$$

$$(x-0)^2 + (y-0)^2 = r_B^2$$

$$(x-U_C)^2 + (y-0)^2 = r_C^2$$

$$r_A = h$$

$$r_B = h + k_B$$

$$r_C = h + k_C$$

Ezekből az egyenletekből megkapjuk x-et és y-t:

$$y = \frac{2 \cdot h \cdot k_B + k_B^2 + V_A^2}{2 \cdot V_A} = h \cdot \frac{k_B}{V_A} + \frac{k_B^2 + V_A^2}{2 \cdot V_A} = h \cdot \left(\frac{k_B}{V_A} \right) + \left(\frac{k_B^2 + V_A^2}{2 \cdot V_A} \right)$$

$$x = h \cdot \left(\frac{k_B - k_C}{U_C} \right) + \left(\frac{k_B^2 - k_C^2 + U_C^2}{2 \cdot U_C} \right)$$

Részletes levezetés:

$$I. (x-0)^2 + (y-V_A)^2 = h^2$$

$$II. (x-0)^2 + (y-0)^2 = (h+k_B)^2$$

$$III. (x-U_C)^2 + (y-0)^2 = (h+k_C)^2$$

$$I. \Rightarrow 1. \Rightarrow x^2 + y^2 - 2 \cdot y \cdot V_A + V_A^2 = h^2$$

$$II. \Rightarrow x^2 + y^2 = h^2 + 2 \cdot h \cdot k_B + k_B^2$$

$$III. \Rightarrow x^2 - 2 \cdot x \cdot U_C + U_C^2 + y^2 = h^2 + 2 \cdot h \cdot k_C + k_C^2$$

$$II. - I. \Rightarrow 2. \Rightarrow 2 \cdot y \cdot V_A - V_A^2 = 2 \cdot h \cdot k_B + k_B^2$$

Ebből kifejezzük y-t:

$$y = \frac{2 \cdot h \cdot k_B + k_B^2 + V_A^2}{2 \cdot V_A} = h \cdot \frac{k_B}{V_A} + \frac{k_B^2 + V_A^2}{2 \cdot V_A}$$

$$III. - I. \Rightarrow 3. \Rightarrow 2 \cdot y \cdot V_A - V_A^2 - 2 \cdot x \cdot U_C + U_C^2 = 2 \cdot h \cdot k_C + k_C^2$$

$$2 \cdot y \cdot V_A - V_A^2 + U_C^2 - 2 \cdot h \cdot k_C - k_C^2 = 2 \cdot x \cdot U_C$$

2. egyenletből következik:

$$2 \cdot y \cdot V_A = 2 \cdot h \cdot k_B + k_B^2 + V_A^2$$

Ezt beírva a 3-as egyenletbe:

$$2 \cdot h \cdot k_B + k_B^2 + V_A^2 - V_A^2 + U_C^2 - 2 \cdot h \cdot k_C - k_C^2 = 2 \cdot x \cdot U_C$$

$$x = (2 \cdot h \cdot k_B + k_B^2 + U_C^2 - 2 \cdot h \cdot k_C - k_C^2) / 2 \cdot U_C$$

$$x = h \cdot \left(\frac{k_B - k_C}{U_C} \right) + \left(\frac{k_B^2 - k_C^2 + U_C^2}{2 \cdot U_C} \right)$$

Tehát x-re és y-ra a következő képleteket kapjuk:

$$x = h \cdot \left(\frac{k_B - k_C}{U_C} \right) + \left(\frac{k_B^2 - k_C^2 + U_C^2}{2 \cdot U_C} \right)$$

$$y = h \cdot \left(\frac{k_B}{V_A} \right) + \left(\frac{k_B^2 + V_A^2}{2 \cdot V_A} \right)$$

4. Próbamérések

Az x-re és y-ra kiszámolt képletekkel a Matlab nevű programban a következő próbamérések lettek elvégezve.

4.1. 1. próbamérés

A hangforrás az első síknegyedben van és a hang a „B” szenzorhoz ér először, aztán a „C”-hez és úgy az „A”-hoz.

A próbamérések során az a, b, c távolságokat Pitagorasz-tétel tétel segítségével számoltam.

$$a = \sqrt{80^2 + 35^2} = 87,3212$$

$$b = \sqrt{20^2 + 35^2} = 40,3113$$

$$c = \sqrt{20^2 + 65^2} = 68,0074$$

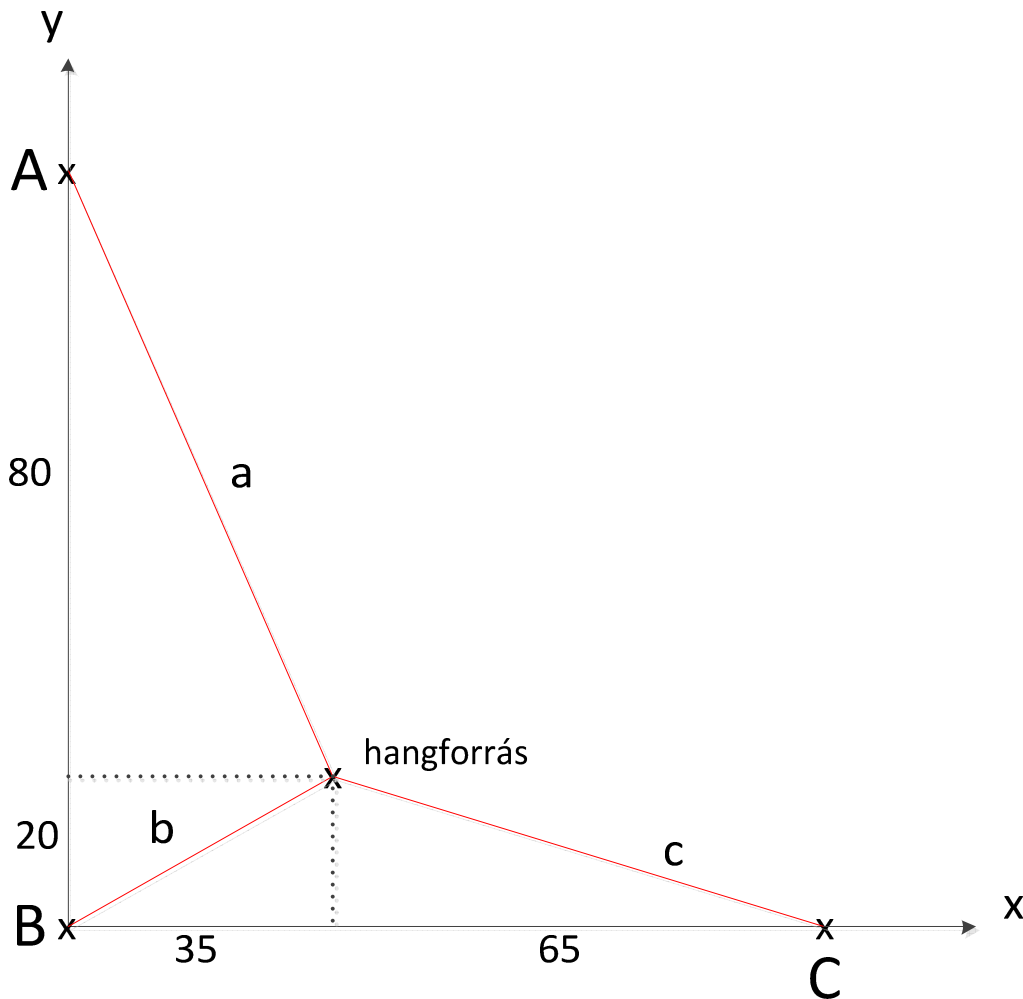
Ebből a k_B és k_C a következőképpen jön ki:

$$k_B = b - a = 40,3113 - 87,3212 = -47,0099$$

$$k_C = c - a = 68,0074 - 87,3212 = -19,3138$$

Erre a rendes méréseknél nincs szükség, mert ott egyből k_B -t és k_C -t kapjuk meg, tehát a távolságkülönbségeket, mivel az első hangot érzékelő mikrofontól számítjuk a 0.

időpillanatot, így megkapjuk a két mikrofon közti időkülönbséget, amelyből már számolható az útkülönbség.



10. ábra. 1. próbamérés ábrázolása

A képlet Matlabban:

$$A1 = h^2 * \left(\left(\frac{k_B - k_C}{U_C} \right)^2 + \left(\frac{k_B}{V_A} \right)^2 - 1 \right) + h * \left(\frac{k_B - k_C}{U_C} * \frac{k_B^2 + U_C^2 - k_C^2}{U_C} + \frac{k_B}{V_A} * \frac{k_B^2 + V_A^2}{V_A} - 2 * k_B \right) +$$

$$+ \left(\frac{k_B^2 + U_C^2 - k_C^2}{2 * U_C} \right)^2 + \left(\frac{k_B^2 + V_A^2}{2 * V_A} \right)^2 - k_B^2$$

A másodfokú megoldóképletbe behelyettesítve:

$$a = \left(\frac{k_B - k_C}{U_C} \right)^2 + \left(\frac{k_B}{V_A} \right)^2 - 1$$

$$b = \left(\frac{k_B - k_C}{U_C} * \frac{k_B^2 + U_C^2 - k_C^2}{U_C} + \frac{k_B}{V_A} * \frac{k_B^2 + V_A^2}{V_A} - 2 * k_B \right)$$

$$c = \left(\frac{k_B^2 + U_C^2 - k_C^2}{2 * U_C} \right)^2 + \left(\frac{k_B^2 + V_A^2}{2 * V_A} \right)^2 - k_B^2$$

$$H1 = \frac{-b - \sqrt{b^2 - 4 * a * c}}{2 * a}$$

$$H2 = \frac{-b + \sqrt{b^2 - 4 * a * c}}{2 * a}$$

A két megoldást behelyettesítjük x és y képletébe:

$$X1 = H1 * \left(\frac{k_B - k_C}{U_C} \right) + \left(\frac{k_B^2 + U_C^2 - k_C^2}{2 * U_C} \right)$$

$$Y1 = H1 * \frac{k_B}{V_A} + \frac{k_B^2 + V_A^2}{2 * V_A}$$

$$X2 = H2 * \left(\frac{k_B - k_C}{U_C} \right) + \left(\frac{k_B^2 + U_C^2 - k_C^2}{2 * U_C} \right)$$

$$Y2 = H2 * \frac{k_B}{V_A} + \frac{k_B^2 + V_A^2}{2 * V_A}$$

Így két megoldást kapunk a másodfokú egyenlet miatt. Behelyettesítés után a végeredmény:

$$X1 = 35$$

$$Y1 = 20$$

$$X2 = 81,8557$$

$$Y_2 = 99,5307$$

Hogy melyik a helyes megoldás, azt abból lehet megállapítani, hogy melyik síknegyedben van a hangforrás.

Jelen esetben az első megoldás a jó, mert az első síknegyedből érkezik a hang.

4.2. 2. próbamérés

A hangforrás az negyedik síknegyedben van és a hang a „B” szenzorhoz ér először, azután a „C”-hez és úgy az „A”-hoz.

A levezetés az előző méréshez hasonlóan történik.

Az a, b, c távolságok kiszámítása:

$$a = \sqrt{135^2 + 25^2} = 137,2953$$

$$b = \sqrt{25^2 + 35^2} = 43,0116$$

$$c = \sqrt{35^2 + 75^2} = 82,7647$$

k_B és k_C kiszámítása:

$$k_B = b - a = 43,0116 - 137,2953 = -94,2837$$

$$k_C = c - a = 82,7647 - 137,2953 = -54,5306$$

Végeredmény:

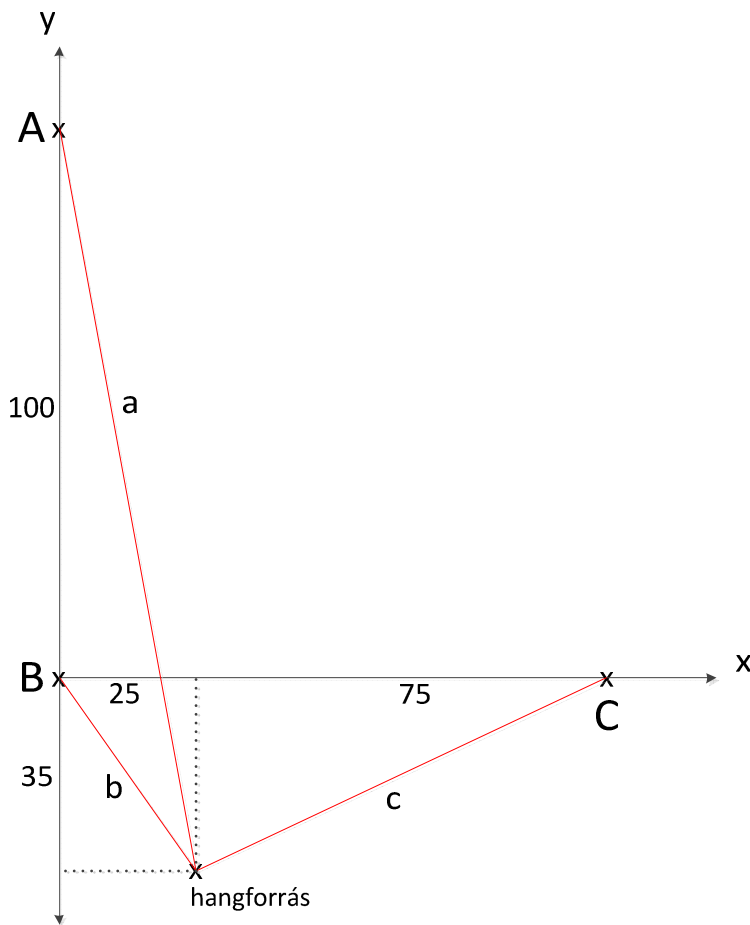
$$X_1 = 25$$

$$Y_1 = -35$$

$$X_2 = -312,6896$$

$$Y_2 = -835,9091$$

A negyedik síknegyedben is az első a helyes megoldás.

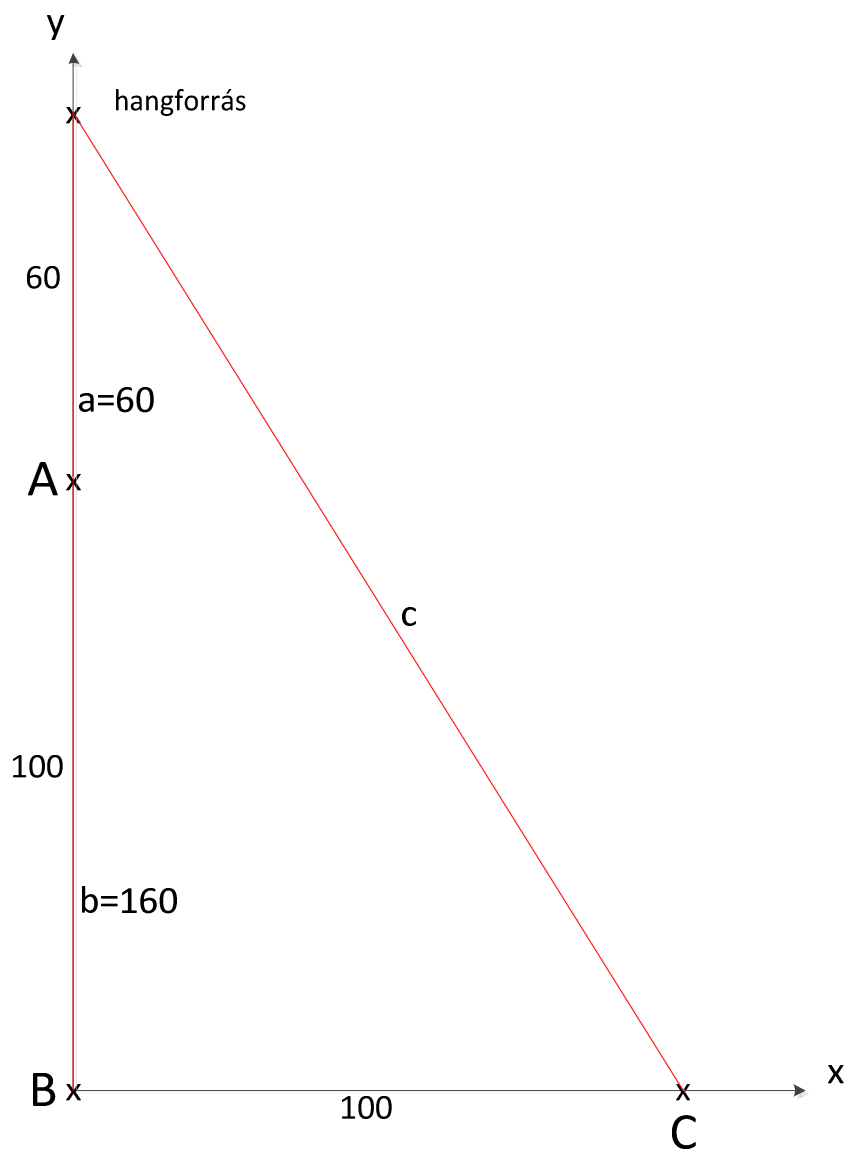


11. ábra. 2. próbamérés ábrázolása

Meg kellett vizsgálni a kritikus pontokból történő hang számításának helyességét. Ilyen pontok lehetnek például, ha a hangforrás egyvonalban van két szenzorral vagy ha mindhárom szenzorhoz egyenlő távolságra van. Szemléltet néhány erre vonatkozó mérést.

4.3. 3. próbamérés

A hangforrás az első és a második síknegyed határán van és a hang az „A” szenzorhoz ér először, azután a „B”-hez és úgy a „C”-hez.



12. ábra. 3. próbamérés ábrázolása

Az a, b, c távolságok kiszámítása:

$$a = 60$$

$$b = 160$$

$$c = \sqrt{100^2 + 160^2} = 188,6796$$

k_B és k_C kiszámítása:

$$k_B = b - a = 100$$

$$k_C = c - a = 188,6796 - 60 = 128,6796$$

Ha a hangforrás két síknegyed határán helyezkedik el, akkor komplex számot kapunk. Amennyiben erről a komplex számról leválasztjuk az imaginárius részt, tehát csak a valós részt vesszük figyelembe, akkor helyes eredményt kapunk. Ilyenkor mindkét megoldás jó.

$$X1 = 1,77 \cdot 10^{-14} + 9,52 \cdot 10^{-7}i$$

$$\text{Imaginárius rész leválasztása után: } X1 = 1,77 \cdot 10^{-14} \approx 0$$

$$Y1 = 1,6 \cdot 10^2 - 3,32 \cdot 10^{-6}i$$

$$\text{Imaginárius rész leválasztása után: } Y1 = 1,6 \cdot 10^2 = 160$$

$$X2 = 1,77 \cdot 10^{-14} + 9,52 \cdot 10^{-7}i$$

$$\text{Imaginárius rész leválasztása után: } X2 = 1,77 \cdot 10^{-14} \approx 0$$

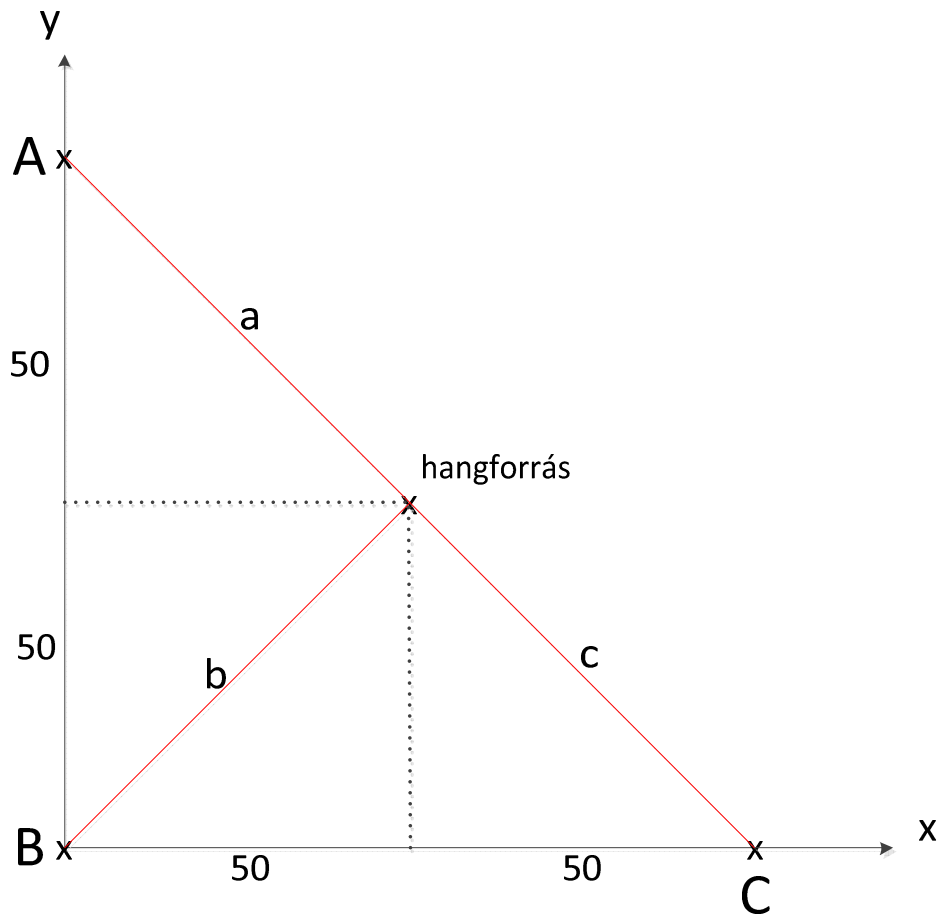
$$Y2 = 1,6 \cdot 10^2 - 3,32 \cdot 10^{-6}i$$

$$\text{Imaginárius rész leválasztása után: } Y2 = 1,6 \cdot 10^2 = 160$$

Az imaginárius rész leválasztása program segítségével lehetséges.

4.4. 4. próbamérés

A hangforrás mind a három mikrofontól egyforma távolságra van, tehát az első síknegyedben.



13. ábra. 4. próbamérés ábrázolása

Az a, b, c távolságok kiszámítása:

$$a = \sqrt{50^2 + 50^2} = 70,71$$

$$b = \sqrt{50^2 + 50^2} = 70,71$$

$$c = \sqrt{50^2 + 50^2} = 70,71$$

k_B és k_C kiszámítása:

$$k_B = b - a = 0$$

$$k_C = c - a = 0$$

A képletbe behelyettesítve a következő eredményt kapjuk:

$$X_1 = 50$$

$$Y_1 = 50$$

$$X^2 = 50$$

$$Y^2 = 50$$

4.5. Mérési pontatlanság hatása és a multiplexelt bemenetből fakadó pontatlanság

Meg kellett vizsgálni, hogy ez a módszer mennyire érzékeny a hibára. Az erre vonatkozó mérések oly módon lettek kivitelezve, hogy hibás adatokat generáltam és azokat helyettesítettem be a képletbe, eközben figyeltem az eltérés mértékét. Ez alapján sikerült belátni, hogy a hiba exponenciális növekedése a hangforrás helyétől függ. Nevezetesen, ha a hangforrás egyvonalban van két mikrofonnal, akkor a hiba exponenciálisan nő. Tehát ebben az esetben az eszköz nagyon érzékeny a hibára.

Szemléltetésül vesszük a 3-as próbamérést és hibát generálunk oly módon, hogy változtatjuk az egyik távolságot, például az „b”-t:

Az eredeti távolságok:

$$a = 60$$

$$b = 160$$

$$c = \sqrt{100^2 + 160^2} = 188,6796$$

Így az x-re és y-ra a következő eredményt kaptuk:

$$x = 0$$

$$y = 160$$

Növeljük az „b” távolságot a többi változatlanul hagyjuk:

$$a = 60$$

$$b = 161$$

$$c = \sqrt{100^2 + 160^2} = 188,6796$$

Az így kapott eredmény:

$$x = 6,69$$

$$y = 143$$

Ez azt jelenti, hogy 1,6%-os eltérésnél, durván 10%-os eltérés mutatkozik a kimeneten.

A távolság nagyobb mértékű változtatása esetén:

$$a = 60$$

$$b = 165$$

$$c = \sqrt{100^2 + 160^2} = 188,6796$$

Az így kapott eredmény:

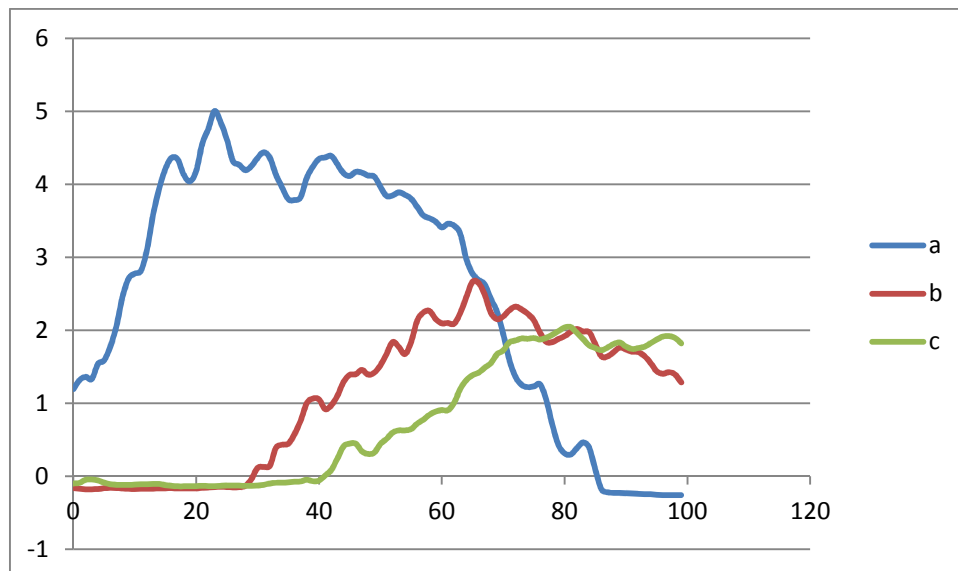
$$x = 22,47$$

$$y = 104,5$$

8,6%-os eltérés esetén 34,7%-os hiba mutatkozik a kimeneten.

Egy másik hiba lehetőség a multiplexelt AD-ből fakadó pontatlanság. Az eddig végzett mérések során ennek a tényezőnek nem volt szerepe. Ellenben, ha továbbfejlesztésre kerül sor, például folyamatos beszédet szeretnénk lokalizálni, akkor nem szabad figyelmen kívül hagyni ezt a tényezőt sem.

4.6. Valós mérési eredmény



14. ábra. Mérés eredményének diagramja

A 14-es ábrán az látszik, ahogy az első mikrofon, tehát az „a” érzékeli a hangot és amikor eléri a küszöb értéket, ami jelen esetben az 1V, akkor triggerel. Ez a 0. időpillanat. Ehhez képest lehet lemérni, hogy mennyi idő múlva éri el a második mikrofon a küszöb szintet. Itt a „b” éri el következőre, aztán pedig a „c”.

Az x tengelyen nem az idő van, hanem a mintavételezések száma. Ha 50kHz-en mintavételezünk, akkor egy minta $20\mu s$ ($1/50000$). Tehát a diagramon az első és a második mikrofon között 39 minta van. Ebből megkapjuk az időkülönbséget: $20\mu s * 39 = 0,78ms$. Tehát az első és a második szenzor között 0,78ms telik el. A hang

terjedési sebességének tudatában ebből már kiszámíthatjuk a távolságot is:

$$0,78\text{ms} * 345 \frac{\text{m}}{\text{s}} = 0,2691\text{m} . \text{ 20-25 fok körüli hőmérsékletet feltételezünk, ezért a}$$

hangsebességet $345 \frac{\text{m}}{\text{s}}$ -nak vesszük.

Az első és a harmadik szenzor közötti idő és út különbséget hasonlóképpen számolhatjuk ki: a két mikrofon között 62 minta van, így az időkülönbség: $20\mu\text{s} * 62 = 1,24\text{ms}$. Ebből a távolságot kiszámolva a következő eredmény kapjuk:

$$1,24\text{ms} * 345 \frac{\text{m}}{\text{s}} = 0,4278\text{m} .$$

Ez a két eredmény a k_B és k_C -nak felel meg, tehát

$$k_B = 0,2691\text{m}$$

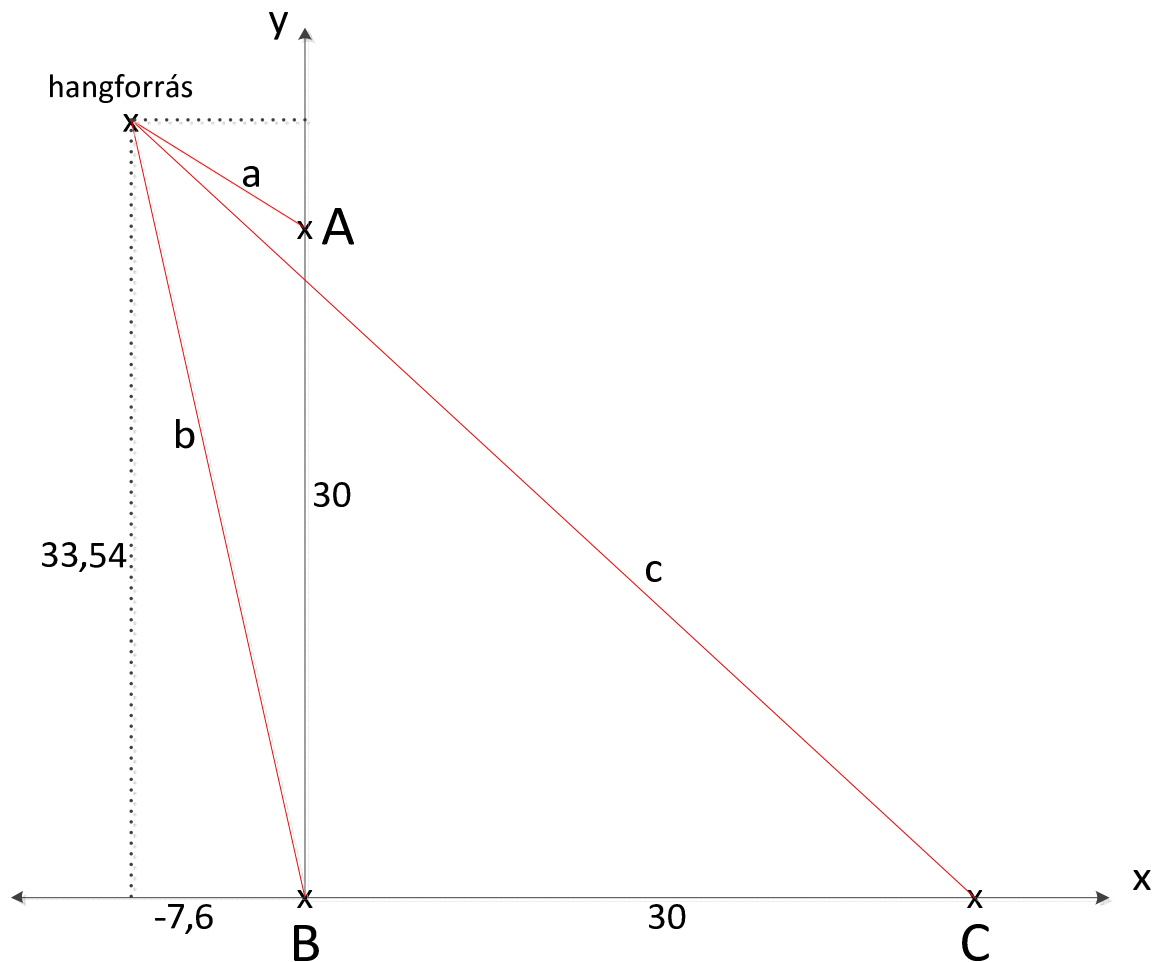
$$k_C = 0,4278\text{m}$$

Ha ezeket behelyettesítjük a már ismertetett képletekbe, akkor megkapjuk az x és y koordinátákat.

$$X1 = -7,6095\text{cm}$$

$$Y1 = 33,5404\text{cm}$$

A próbamérések során egységnyi hosszokkal számoltunk, most viszont oda kell figyelni a mértékegységekre. A tényleges megvalósításban a mikrofonok 30cm-re helyezkednek el egymástól.



15. ábra. Valós mérési eredmény ábrázolása

5. A LabVIEW program bemutatása

A LabVIEW-ben fejlesztett VI a termelő-fogyasztó tervezési mintát követi.

A termelő ciklusban történik a mikrofonos egységekből érkező jel mintavételezése 50 kHz-es frekvencián. A mintavételezett jel 10 000 mintát tartalmazó csomagok formájában kerül átadásra a fogyasztó ciklusnak. A két ciklus (termelő, fogyasztó) közötti kommunikáció Queue-n keresztül történik, amely kialakításánál fogva adatvesztés és adattöbbszörözés nélkül képes adatokat továbbítani. A Queue gyakorlatilag egy FIFO, melynek mérete meghatározható az inicializáció során.

A fogyasztó ciklus felel a feldolgozásért, másodpercenként 5-ször fut le, mivel 50 kHz-es mintavételezés mellett 10 000-es csomagok formájában történik az adattovábbítás. A feldolgozó ciklusban egyenként a három csatorna jelsorozatában keressük, hogy a

mintavételezett jelben szerepel-e egy impulzus, amely impulzus lehet pl. egy csettintés eredménye. Amennyiben gerjesztjük a rendszerünket azzal a bizonyos hangimpulzussal, úgy ez a hangimpulzust mindhárom csatorna esetén detektálható, ezért a LabVIEW-s VI meghatározza, hogy az egyes csatornák esetén az impulzushoz milyen index tartozik. A fogyasztó ciklus az indexkülönbségeket úthosszkülönbségekre átszámolja. Az úthosszkülönbségekből a dolgozatban megadott képletek segítségével (melyek a LabVIEW-s kódba implementálásra kerültek) meghatározza az x és y koordinátákat.

6. Összefoglalás

Dolgozatomban egy olyan szenzorhálózatot mutattam be, amely hang útján határozza meg a hangforrás helyét. Ezen eszköz nagy előnye, hogy elegendő önmagában a hang kibocsátása, melyből több szenzort alkalmazva megállapítható a hangforrás hely, nem igényel egyéb kiegészítő komponenseket.

6.1. Eredmények értékelése

Munkám során megismertem a LabView-t, - amely egy szoftveresen definiált mérőműszer - és az ehhez tartozó mérésadatgyűjtő kártyát. Kialakítottam a mérési összeállítást, illesztettem a mikrofonerősítőként használt tranzisztoros erősítőkapcsolást a mérőkártyához. Felállítottam a mérés matematikai modelljét.

6.2. Tervek a jövőre nézve

Az eszköz továbbfejlesztésére számos lehetőség kínálkozik. Ezek közül az egyik, ha nem csak egy impulzus hatására működne az eszköz, hanem folyamatos jel, például folyamatos beszéd lokalizációja esetén is. Ehhez szükség van mintaillesztésre és frekvenciamérésre.

A jövőben szeretném tökéletesíteni az eszközt és beépíteni valamilyen robotba. További alkalmazási terület, ha két vagy több robot kommunikációját oldjuk meg ily módon. A robotok valamilyen hangot idéznének elő és ez alapján lokalizálnák egymás helyzetét. Természetesen az alkalmazott hang lehetne ultrahang is, tehát a hallható frekvencia feletti tartomány.

Egy következő megoldás lehet, ha a robotok lokalizálandó hangként egy dedikált frekvenciát bocsátanak ki, melyből az adott robot azonosítható.

Az említett továbbfejlesztési lehetőségek egy szakdolgozat, vagy egyéb tudományos kutatómunka tárgyát képezhetik.

7. Felhasznált szakirodalom

[1] – <http://www.labview.com>

[2] – Scharnitzky Viktor: Vektorgeometria és lineáris algebra

[3] - <http://www.mathworks.com/products/matlab/>

[4] - <http://www.hobbielektronika.hu>