Diplomamunka

László Vendel

Mikrofontömbön alapuló iránydetektáló rendszer fejlesztése

Témavezető: Németi Péter
FVT test engineer
National Instruments
Konzulens Dr. Légrády Dávid
egy. docens
BME NTI

2015.

Témakiírás

A nyalábformálást mint alapelvet széleskörűen használják rádiófrekvenciás rendszereknél, a hangtechnikában és az ultrahang diagnosztikában. Az alapelv lényege, hogy a sugározni kívánt nyaláb, illetve detektálás esetén a detektor iránykarakterisztikája nem mechanikus úton változtatható, mint pl. egy forgó radar esetében, hanem több, akár több ezer kisebb forrás vagy detektor jelének súlyozott, adott időkésésű összegzésével jön létre, valós idejű jelfeldolgozás segítségével.

A források vagy detektorok egymáshoz képesti helyzete, illetve a jelek késleltetésére és súlyozására használt algoritmus együttesen határozzák meg az iránykarakterisztikát. A megvalósítás különbözik, azonban az alapelvek megegyeznek hangfrekvenciás, ultrahang illetve rádiófrekvenciás hullámok esetén is.

A hallgató feladata egy olyan fejlesztői platform létrehozása, amellyel különböző iránydetektáló algoritmusok szimulációja és tesztelése valósítható meg. A diplomamunka során a hallgatónak lehetősége nyílik számos, az FPGA-k programozására, digitális szűrőkre, jelfeldolgozásra és szoftverfejlesztésre vonatkozó ismeret elsajátítására, mely felhasználható többek között ultrahang diagnosztikai eszközök fejlesztésére is.

Nyilatkozat

Alulírott *László Vendel*, a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatója kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, és a diplomatervben csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

_____

László Vendel
hallgató

Tartalomjegyzék

Kivonat

A nyalábformálás módszere széleskörűen használt rádiófrekvenciás rendszerekben és ultrahangos készülékekben, mellyel a detektorok nagy számát felhasználva jobb vételi karakterisztika és irányérzékenység érhető el. Egy ilyen rendszer a rádióantennák és piezo érzékelők felhasználása miatt viszonylag költséges és a tesztelésük körülményes. A diplomamunka fő motivációja az volt, hogy létrehozzak egy olcsó és egyszerű alternatívát, mely képes szimulálni és tesztelni ugyanazokat a módszereket, melyeket a rádiózásban vagy ultrahangos érzékelőkben alkalmaznak, és a jövőben felhasználható ilyen célú fejlesztésekre.

Ebből a célból készítettem egy mikrofontömbön alapuló iránydetektáló rendszert, melyhez MEMS mikrofonokat és egy Xilinx Spartan3-as FPGA-t tartalmazó NI sbRIO 9642-es board-ot, mellyel a mikrofonok vezérléséhez és az adatok feldolgozásához használtam. A diplomamunkámban ismertetem az elkészített rendszer felépítését, és a hozzá tartozó szoftveres környezetet, mely a beérkező adatok feldolgozását végzi. Az iránydetektáláshoz megvizsgáltam néhány módszert, többek között egy egyszerű jeleltoláson alapuló módszert, GCC-PHAT, és az MCCC algoritmusokat. Ezek a módszerek kisebb szögeknél jól teljesítettek, és felhasználhatóak a jövőbeli fejlesztés során.

Abstract

Beamforming is a widely used method among radio systems and ultrasonic devices to achieve better signal characteristics and direction sensitivity with multiple detectors. Building a system with radio antennas or piezo sensors can be expensive and hard to test. The motivation behind this thesis was to create a cheap and simple alternative, which is capable to simulate and test the same methods used in radio or ultrasonic systems, and can be used in the future for further development.

To achieve this goal, I developed a microphone array based direction finder system using MEMS microphones and an NI sbRIO 9642 board with a Xilinx Spartan3 FPGA, which I used for measure and process the data from the microphones. In this thesis, I describe the created system, and the software environment, which handles the measured data. For estimating the direction of the sound source, I examined different methods, including a simple delay and sum, GCC-PHAT and the MCCC algorithms. These methods performed well in lower angles, and can be utilised in further development.

Előszó

Napjainkban a nagy teljesítményű IC-k (Integrated Circuit) terjedésével megfigyelhető a valós időben (real-time) történő adatfeldolgozás térnyerése, ennek köszönhetően a mérőeszköz sokszor már nem csak a mért jel továbbítását végzi, hanem elvégzi annak alapszintű feldolgozását is, ezzel csökkentve a sávszélességet és tehermentesítve a mérőeszköz többi részét. A real- time feldolgozással lehetőség nyílik olyan adaptív mérőeszközök létrehozására, melyek a karakterisztikájukban folyamatosan alkalmazkodnak a környezeti változásokhoz (pl. smart antenna), így mindig a lehető legoptimálisabb teljesítményt nyújtva.

Ilyen adaptív eszközök csoportjába sorolhatóak a különféle szenzorrendszerek (Sensor Array), melyekkel a diplomamunkámban is foglalkoztam. A szenzorrendszer egy sokdetektoros mérőeszköz, melyben a detektorok szerepét rendszerint valamilyen akusztikus vagy elektromágneses szenzor tölti be. A mérés szinkronban történik az összes detektoron, és mivel minden egyes szenzor növeli a mérőrendszer szabadsági fokát, így a redundanciát kihasználva a mérés pontossága ezek számával arányosan növekszik. Ilyen szenzorrendszerekre példa a rádiózásban használt fázis-vezérelt antennarács, melyben a detektorok szerepét többnyire dipól antennák töltik be, vagy az ultrahangos transzdúcerek, melyekben szorosan egymás mellé helyezett piezokristályok detektálnak mechanikai rezgéseket.

Szenzorrendszerekkel történő mérés egyik kulcsfontosságú eleme a nyalábformálás, mellyel a rendszer irányérzékenysége tetszőlegesen megválasztható. Gyakori eset, hogy a jelforrás iránya (DOA - Diretion of Arrival) a mérés során nem ismert, így az elsődleges feladat ennek a minél pontosabb becslése. A beérkezés irányának ismeretében a szenzorok jelei összeadhatóak úgy, hogy a mikrofonokon vett jelek azonos fázisban legyenek, így a konstruktív interferencia miatt a jel minősége (SNR – Signal to Noise Ratio) jelentősen javítható.

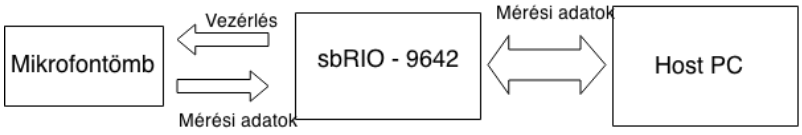
A diplomamunka fő motivációja az volt, hogy létrehozzak egy olyan platformot, amelynél a szenzorrendszereknél használt matematikai elvek és algoritmusok kipróbálhatóak és tesztelhetőek, viszont szemben az antennákkal vagy piezokristályokkal, olcsó és könnyen tesztelhető. Erre a célra építettem egy mikrofonokból álló szenzorrendszert, melyet a későbbi tesztek alapján igazoltam, hogy valóban alkalmas erre a célra. A munkát a National Instrument-nél (NI) végeztem, mely során NI-os hardvert illetve a LabVIEW szoftvert használtam.

A diplomamunkám első felében ismertetem a mérőeszköz fejlesztése során felhasznált eszközöket, illetve néhány hozzájuk kapcsolódó fogalmat. Ezután ismertetem a különböző problémákat, amik a fejlesztés során felmerültek, a gyakorlati megoldá-

sokat amiket ezekre alkalmaztam, illetve kitérek a szoftverre, mely a mérés vezérlését végzi. Ezután ismertetek néhány módszert, melyek használhatóak a DOA meghatározásához, majd bemutatom az ezekhez kapcsolódó mérési eredményeket és tapasztalatokat.

1 | A rendszer specifikációi

A mérőrendszert felépítésileg három részre osztható, az adatok kiértékelését végző számítógép (host-PC), a mérést végző hardverre (board), és a mikrofontömbre. A jelenlegi verzió 7 mikrofoncsatornát tud egyidejűleg kezelni, de ez komolyabb módosítás nélkül kiterjeszthető jóval több csatornára is. A mérést végző hardver egy NI sbRIO-9642-es board, mely tartalmaz egy Xilinx Spartan-3-as FPGA-t, ennek az I/O kivezetéseit használtam a mikrofonok vezérléséhez valamint a mért adatok begyűjtéséhez.



1.1. ábra. A készülék blokkvázlata

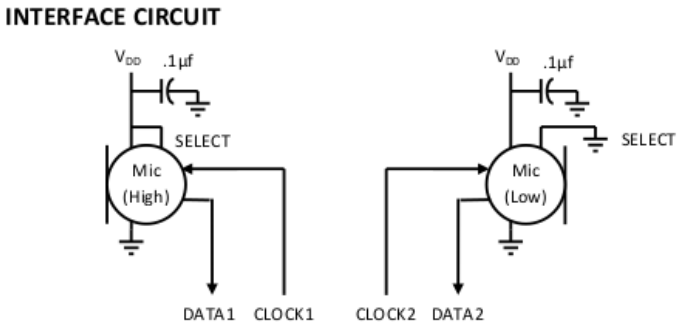
1.1. A mikrofonok

A fejlesztés során detektorként MEMS (Micro-electro Mechanical System) mikrofonokat használtam. Az ilyen típusú mikrofonokra jellemző, hogy a tokozáson belül, jellemzően a szilíciumlapkára integrálva tartalmaznak egy A/D (analog to digital) átalakítót, így a kimenetük is digitális jel. Az A/D konverziót egy Sigma-delta ADC végzi, melynek a formátuma PDM (Pulse Density Modulated) jel. A mikrofonok tápfeszültségét (Vdd) 3,3V-nak szabtam meg meg, hogy illeszkedjen az FPGA digitális kimeneteinek jelszintjéhez.

Pin#	Név	A jel típusa	Leírás
1	GROUND	Power	Föld
2	SELECT	Input (nem digitális)	Lo/Hi szenzor kiválasztása, alapból a földre van kötve
3	GROUND	Power	Föld
4	CLOCK	Digitális input	Az ADC mintavételezési frekvenciája
5	DATA	Digitális output	PDM formátumú jel
6	Vdd	Power	Táp

1.1. táblázat. A mikrofon kivezetései, és a betöltött funkciójuk

A mikrofonok tokozásán belül két szenzor foglal helyet, melyek támogatják a sztereó működési módot(1.2 ábra). Sztereó működés során a CLOCK jel fel és leszálló ágán felváltva történik mintavételezés a két szenzoron, melyek áramköre úgy lett kialakítva, hogy egyszerre mindig csak egy mikrofonon történjen mérés (1.5 ábra). A két szenzor adata a DATA kimeneten összefésülődik, így sztereó működés esetén az effektív mintavételezési frekvencia CLOCK órajel kétszerese.



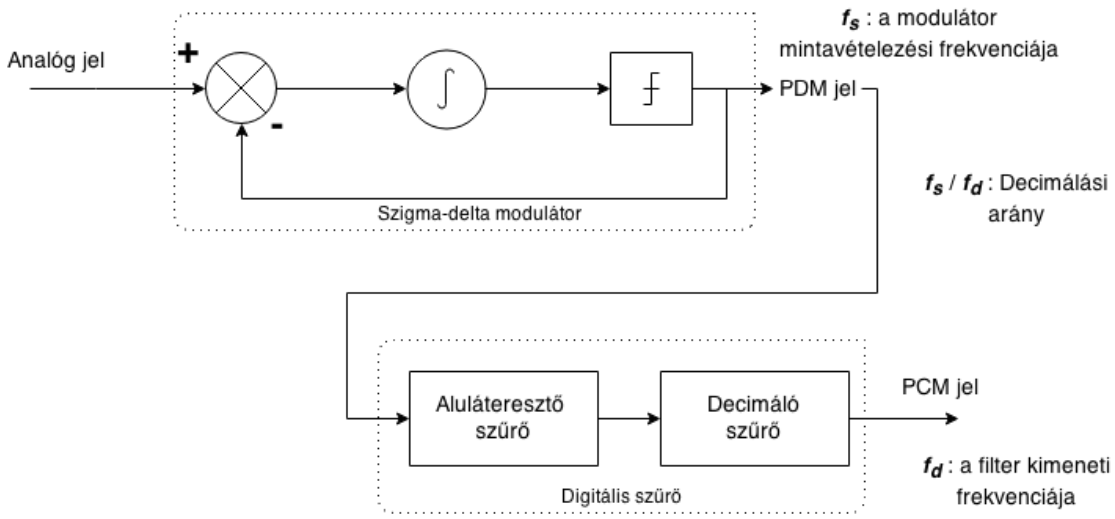
Microphone	SELECT	Asserts DATA On	Latch DATA On
Mic (High)	V_{DD}	Rising Clock Edge	Falling Clock Edge
Mic (Low)	GND	Falling Clock Edge	Rising Clock Edge

1.2. ábra. A mikrofon áramkörének sematikus rajza

1.1.1. Sigma-Delta A/D konverter

A mikrofonokban az analóg jel átalakítását egy beépített szigma-delta analóg-digitális konverter (ΣΔ ADC) végzi. A ΣΔ moduláció a delta modulációs A/D konverzió egyik formája, mely során nem a mintavételezett jel abszolút értéke, hanem annak a megváltozása kerül digitalizálásra. A ΣΔ ADC esetében a különbség A/D konverziója (legtöbb esetben) 1 biten történik, ami önmagában csupán csak annyit adna meg, hogy az előző mintavételhez képest a jel milyen irányban változott. Ahhoz, hogy ebből mégis kiértékelhető adat legyen, szükséges, hogy a mérni kívánt jel túlmintavételezett legyen, vagyis hogy a ΣΔ ADC mintavételezési frekvenciája jelentősen nagyobb legyen, mint a mérés szempontjából vizsgálandó legmagasabb frekvencia.

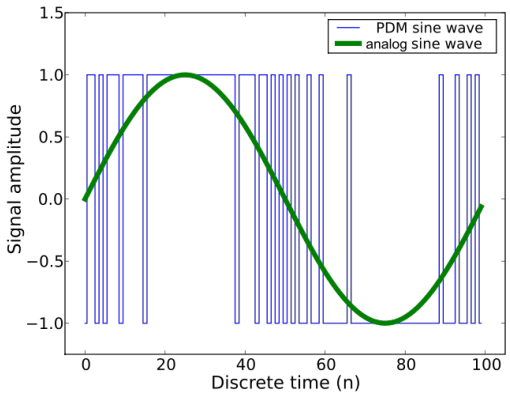
A ΣΔ ADC esetében a jel túlmintavételezése és a zajformálás együttes hatása, hogy a kvantálási zaj spektruma a mérendő tartományon jelentősen csökken. A jobb jel zaj viszony eléréséhez alkalmazható egy alul-áteresztő szűrő, mely a zaj spektrumából a nagyfrekvenciás komponenseket eltávolítja [1]. A digitális szűrést rendszerint FPGA-n vagy DSP-n történik, melyek gyorsaságuk és nagy számítási kapacitásuk miatt alkalmasak a túlmintavételezett jel szűrésére (a mikrofonok frekvenciája $\mathbf{f_s}$ = 5,714MHz). Az általam használt szűrő egy FPGA-n megvalósított 2.rendű CIC (Cascaded Integrator Comb) szűrő volt, melyet részletesen a 2.2.2 fejezetben ismertetek.



1.3. ábra. A szigma-delta ADC blokkdiagrammja [2]

1.1.2. A PDM jel

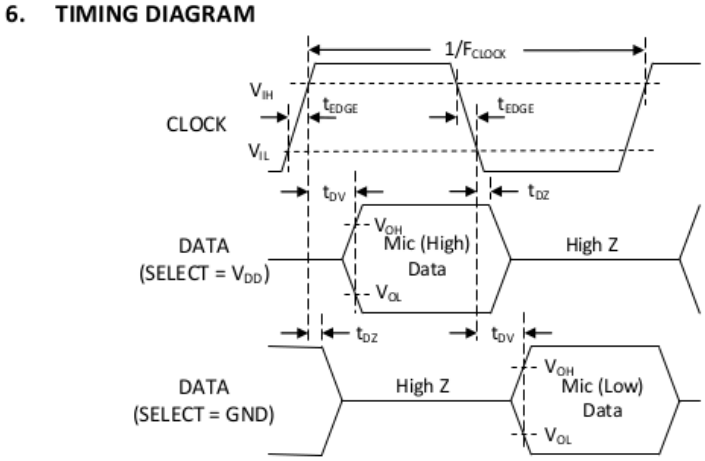
A PDM (Pulse Density Modulation) jelformátum a PCM (Pulse Code Modulation) kódolással szemben a digitalizált jelet nem x-bites szavakban, hanem egy 1-bites adatfolyamban reprezentálja, melyben az analóg jel arányos az adatfolyamban lévő 1-esek és 0-k arányával. Mivel a PDM jel az alacsony frekvenciás spektrumában tartalmazza az analóg jelet, így annak visszaállítása egy megfelelően tervezett aluláteresztő szűrővel könnyen kivitelezhető. PCM jellé való konverzió egy kicsit problematikusabb, mivel ebben az esetben a szűrést digitális szűrő végzi, így olyan decimáló szűrőt kell alkalmazni, amely biztosítja, hogy a nagyfrekvenciás jel ne lapolódjon át a mérendő (jelen esetben hallható) tartományba [**?**].



1.4. ábra. Analóg és PDM formátumú szinuszjel összehasonlítása [4]

A mikrofonokon kimenetén megjelenő PDM jel frekvenciája arányos a mikrofonokra leadott CLOCK jel frekvenciájával, mely jelen esetben 2,857MHz volt. A beérkező

PDM jel frekvenciája a sztereó elrendezés miatt valójában ennek a frekvenciánk a kétszerese, 5,714MHz. A CLOCK jel szolgál a mérés triggereléséhez, melyet egyben a mikrofonok jelének az összehangolására használtam. A kimeneten a mért adat csak egy adott időkésés múlva lesz mérhető (delay time for valid data 1.5 ábra), melyet 87,5ns-ra állítottam.



1.5. ábra. A mikrofonok időzítése

Clock Frequency	F_{CLOCK}		1.0	-	3.25	MHz
Clock Duty Cycle			40	-	60	%
Clock Rise/Fall Time	t_{EDGE}		-	-	10	ns
Fall-asleep Time[4,5]		F_{CLOCK} < 1 kHz	-	-	10	ms
Wake-up Time[4,6]		$F_{CLOCK} \geq$ 1 MHz	-	-	20	ms
Delay Time for Valid Data	t_{DV}	No load for min t_{DV}	18	-	125	ns
		Max C_{LOAD} for max t_{DV}				
Delay Time for High Z	t_{DZ}		0	-	16	ns

1.6. ábra. A mikrofonok időzítési paraméterei

1.2. Az NI sbRIO-9642

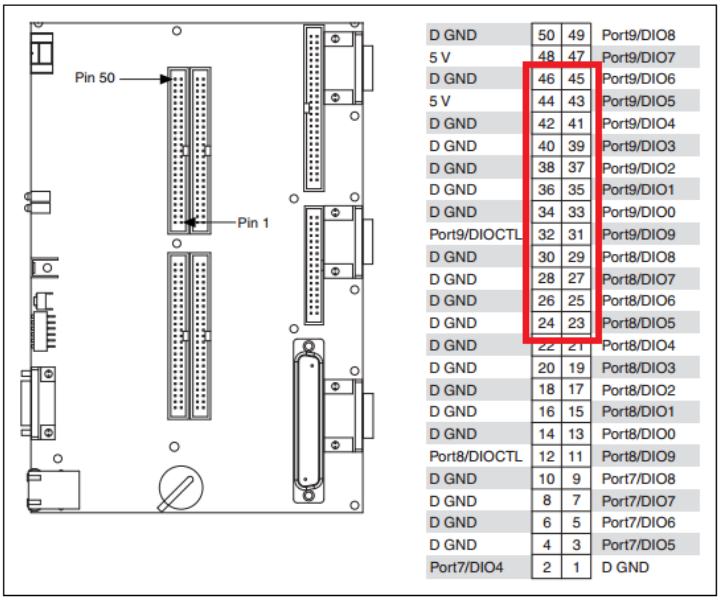
Az NI sbRIO-9642 (single-board Reconfigurable Input / Output) egy egyetlen nyomtatott áramkörre integrált beágyazott vezérlő és adatrögzítő kártya. A mérőrendszer szempontjából fontos volt egy olyan eszköz választása, ami rendelkezik fedélzeti FPGA-val, mivel ez végzi a mikrofonok vezérlését illetve a jelek digitális szűrését. Az sbRIO-9642 ennél jóval többre is képes lenne, habár a funkcióit tekintve közel sincs teljesen kihasználva, a fejlesztés során elsősorban az elérhetősége és könnyű kezelhetősége miatt került felhasználásra. Ennek az eszköznek a cseréje megoldható egy másik eszközre, amennyiben az megfelel az alábbi paramétereknek:

- beépített fedélzeti FPGA ()javasolt legalább 40MHz-es frekvenciájú

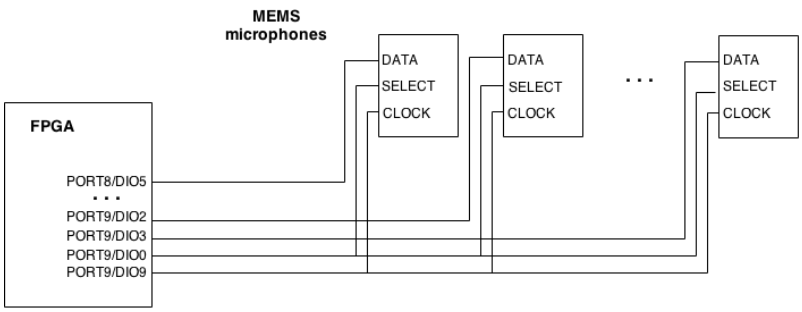
- DMA csatorna
- legalább hét DIO csatorna
- 3,3V-os DC táp a mikrofonokhoz

1.2.1. Kommunikáció

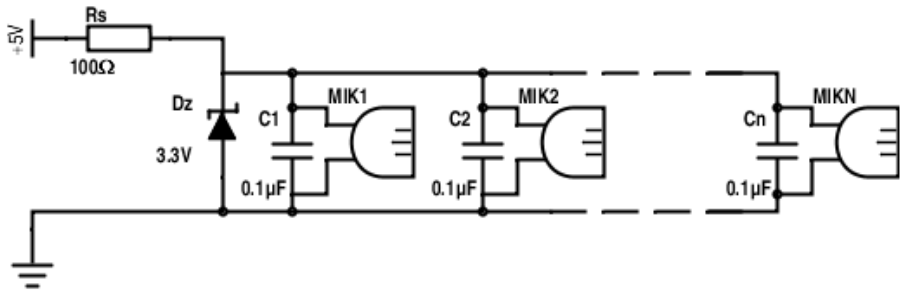
A mikrofonokkal való kommunikációra a board P3-as csatlakozóin lévő kimeneteket használtam (46-23, 1.7 ábra) melyek bekötési rajza a 1.8 ábrán látható.



1.7. ábra. Pinout of I/O Connector P3, 3.3 V Digital I/O



1.8. ábra. A mikrofonok és az FPGA bekötési rajza



1.9. ábra. A mikrofonok áramköri rajza

1.2.2. Tápellátás

A sbRIO-hoz egy standard 24V-os tápegységet használtam, és a mikrofonok tápellátásához a board digitális portjain lévő +5Vos kivezetéseket használtam. A +5V a csatlakozó digitális föld pontjaihoz képest van értelmezve és közvetlenül az sbRIO belső 5V-os tápcsatornájára van kötve. A +5V-os forrás túlfeszültség és túláram elleni védelmet is tartalmaz, azonban zajos terhelések és hirtelen ugrások az áramban nem kívánt felharmonikusok megjelenését eredményezheti a készülék tápcsatornái között. Ezek a tranziensek kiszámíthatatlan viselkedéshez vezethetnek digitális műszerek esetén, ezért egy olyan műszer csatlakoztatásánál, amely nem egy zajmentes, egyenletes külső terhelésként viselkedik, egy szűrő beiktatása szükséges, esetleg további áramkorlátozás. [5]

A tápfeszültség stabilizáláshoz és a jelszint illesztéséhez egy 3,3V-os teljesítmény szabályozó zener diódát használtam. A hirtelen feszültségugrások ellen a mikrofonokon egy-egy 0,1 μF-os bypass kapacitást használtam. A mikrofonok áramkörének a kapcsolási rajza az 1.9 ábrán látható.

2 | A szoftver

A szoftver készítésekor az elsődleges cél az volt, hogy a mikrofonokból érkező jel folyamatosan fel legyen dolgozva, és továbbítva legyen a PC felé. Az adatvesztést a lehetőségekhez mérten ki kellett küszöbölni, ezért törekedtem egy olyan megoldásra, ami ezen céloknak megfelel. Ehhez igyekeztem a PC-t minél inkább tehermentesíteni a számítások alól, így a szűrést és jelátalakítást teljes mértékben az FPGA végezte. További szempont volt, hogy a program minél flexibilisebb legyen, képes legyen kezelni tetszőleges számú mikrofont minimális módosításokkal, valamint hogy az elkészített kód könnyen módosítható és átlátható legyen. Ezen szempontokat szem előtt tartva készítettem az FPGA és a host-PC-n futó kódot, melyeket ebben a fejezetben ismertetek.

2.1. A LabVIEW fejlesztőkörnyezet

A diplomamunkám során a programozáshoz a LabVIEW platformot használtam. A LabVIEW (Laboratory Virtual Instrument Engineering Workbench) egy fejlesztői környezet, mely lehetővé teszi különféle hardverek programozását magas szintű, grafikus programozási nyelven. (A grafikus programozási nyelv neve egyébként G programnyelv, de gyakori, hogy ezt nevezik LabVIEW-nak). A magas szintű programozási nyelv előnye, hogy a fordító automatikusan elvégez néhány feladatot, például memória allokáció, hardverelemekkel való kommunikáció, így ezekkel a programozás során már nem kell külön foglalkozni. A LabVIEW használatával a fejlesztés egyszerűsíthető és felgyorsítható, egy LabVIEW fejlesztőt idézve: “We write low-level code so you don’t have to.”

A LabVIEW-ban használt G - programozási nyelv (G language) egy univerzális programozási nyelv, melyben a programozás – ellentétben a hagyományos programozási nyelvekkel– nem szövegesen, hanem grafikusan történik, a program működését leíró egyfajta folyamatábra megrajzolásával. Lévén univerzális programozási nyelv, megtalálható benne a legtöbb programozási nyelvben használt eszköz, mint például az adattípusok, ciklusok, eseménykezelés, változók, rekurzió, valamint lehetőséget nyújt olyan programozási paradigmák használatára is, mint például az objektum orientált programozás. Technikailag nézve a G nyelv egy "dataflow" alapú programozási nyelv, melyben az utasítások végrehajtása eltérően a script nyelvektől nem szekvenciálisan, hanem az adatok elérhetővé válásának a sorrendjében történik.

A LabVIEW-al készített programot VI (Virtual Instrument)-nak hívják, mely két fő elemből, a blokk diagramm (Block Diagram) és a front panelből (Front Panel) áll. A

programozás a blokk diagrammon történik, ezen történik az utasításokat tartalmazó node-ok felvitele és vezetékekkel (wire) való összeköttetése. A vezetékek a végrehajtás sorrendjét határozzák meg, a node-ok pedig a végrehajtandó feladatot, mint például valamilyen matematikai művelet, egy mérőeszköz számára kiadott parancs, vagy akár egy másik VI futtatása. A felhasználóval való interakciókért a front panel (Front Panel) felelős, mely egyben GUI-ként (Graphical User Interface) is funkcionál.

Mivel a blokk diagramm magas (algoritmikus) szinten ír le egy feladatot, így nagy előnye, hogy teljesen eltérő programozási paradigmával rendelkező hardverek is (mint például a CPU és az FPGA) ugyanabban az egységes környezetben programozhatóak, így nagyban leegyszerűsítve a fejlesztési és a tanulási időt. A LabVIEW erre a célra számos modullal és toolbox-al rendelkezik, így széles területen (Motion, Vision, DAQ, RF, DMM, FPGA, Real-Time stb.) biztosít egységes platformot a fejlesztéshez.

2.2. Az FPGA-n futó VI

Az FPGA kód tervezésekor a skálázható felépítést tartottam szem előtt. A jelenlegi verzió 7 mikrofont képes kezelni, de ez igény szerint, komolyabb módosítás véghezvitele nélkül bővíthető több mikrofonra is. A VI két fő részegységből áll, a mikrofonok mintavételezését és szűrést végző, illetve az adatoknak a DMA FIFOba történő kiírását végző egységből. Ez a két részegység a blokk diagrammon egy egy while-loop struktúrában jelenik meg, melyek logikailag az FPGA-n két, egymással párhuzamosan futó logikai modulnak felel meg.

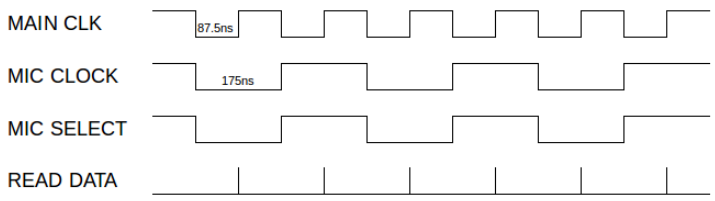
Megjegyzendő, hogy a host-PC-n futó mérőprogram csak az FPGA-VI-ból generált bitfájlt használja, így módosítás esetén ennek újragenerálása szükséges (ez meglehetősen sok idő, 40-50 perc is lehet). Az újragenerálást követően a régebbi bitfájlok ugyanúgy használhatóak maradnak, így például a szűrőalgoritmus megváltoztatása esetén lehetővé válik a régi és új szűrővel vett jel karakterisztikájának az összehasonlítása. A teljes VI az 2.11 ábrán látható

A VI tartalmaz néhány extra regisztert, melyek a VI főbb paramétereit tartalmazzák, mint például a FPGA VI által kezelt mikrofonok száma, vagy az alaórajel frekvenciája. Ezek a paraméterek a mérőprogram futtatásakor jutnak szerephez, mivel a mérőprogram a konfigurálást ezen értékek alapján végzi el. Az FPGA kód a 2.3.4 fejezetben feltüntetett fix paramétereken túl tetszőlegesen módosítható, például új filterek, több mikrofon, nagyobb alapórajel adható meg, így különböző FPGA kódok is tesztelhetőek ugyanabban a környezetben.

2.2.1. Adatgyűjtés

A mikrofonok vezérlése az FPGA digitális I/O kimenetein történik, a 2.1 ábrán látható időzítések alapján, ahol a MAIN CLK a VI-on belül használt alapórajel, a MIC CLOCK, és MIC SELECT a mikrofonok bemeneteire küldött jel, a READ DATA pedig a mikrofonok DATA kimenetének a mintavételezési ideje.

A mintavételezést végző modul egy speciális, Single Cycled Time Loop objektumban foglal helyet, mely biztosítja, hogy az FPGA kód szintézise során a mintavételezés

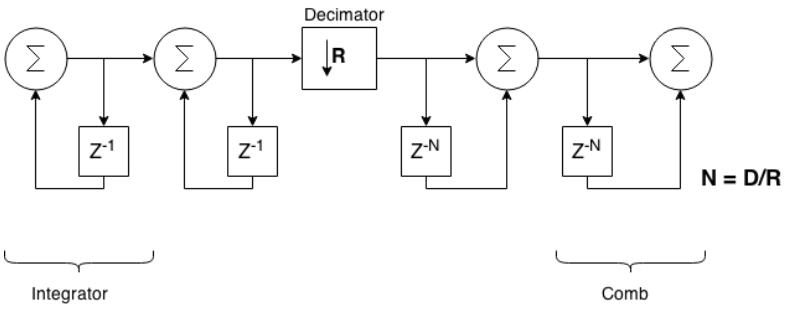


2.1. ábra. A mikrofonok időzítési diagramja

és szűrés beleférjen a megadott időzítési keretbe. A DMA FIFOba történő írást egy párhuzamosan futó ciklus végzi, ami mindig csak a decimálást követő ciklusban fut le. A két ciklus közötti kommunikációt egy belső FIFO memória végzi el, ami lehetővé teszi a két ciklus aszinkron működését. Az FPGA-n lévő szűrőt úgy terveztem, hogy az alkalmazott decimálási arány változtatható legyen, és így a kimeneti frekvencia hozzáigazítható a host-PC feldolgozási sebességéhez. A rendeltetés szerinti működéshez szükséges, hogy a DMA FIFO-ba történő írás gyorsabb legyen, mint a szűrő kimeneti frekvenciája, ezt a belső FIFO Timed Out változójának a monitorozásával ellenőriztem, ami jelzi, ha a lassú kiolvasás miatt a FIFO bufferje megtelt.

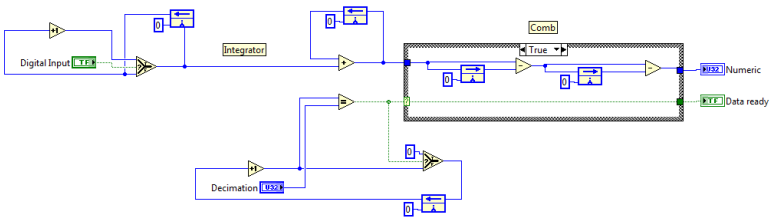
2.2.2. Digitális szűrő

A szűrő kiválasztásánál elsősorban egy olyan megoldást kerestem, ami alacsony számításigényének köszönhetően jól skálázható több mikrofonig, és aminek magas frekvenciákon jó a csillapítási tényezője. Az alkalmazott szűrő végül egy 2. rendű CIC filter lett, melynek jellegzetessége, hogy a hagyományos FIR (Finite Impulse Response) szűrőkkel ellentétben az algoritmus csak összeadást-kivonást tartalmaz, illetve felépítéséből adódóan tartalmaz egy decimáló fokozatot, melynek használatára a ΣΔ ADC mintavételezési frekvenciájának a redukálása miatt egyébként is szükséges lenne.



2.2. ábra. 2.rendű kaszkádosított CIC filter sematikus rajza

A CIC filter alapjaiban leginkább egy rekurzív mozgóátlagos szűrőre hasonlít (RRS - Recusive Running Sum), leszámítva, hogy az összeget a végén nem átlagolja. Felépítését tekintve két részegységre bontható, az integráló (Integrator) szakaszra, mely az adatok folyamatos összegzését végzi, illetve a fésű (Comb) szakaszra, ami egy adott



2.3. ábra. LabVIEW-ban megvalósított szűrő

késleltetéssel a korábban összegyűjtött adatok kivonását végzi. A késleltetés nagysága szabja meg az átlagolás szélességét és ezzel a felbontás nagyságát, így például ha a késleltetés 10 ciklus, akkor a 110.ciklusban az eddig szummázott adatokból kivonásra kerül a 100. lépésig szummázott összeg, így az eredmény a legutóbbi 10 adat összege lesz. Az egyik leggyakoribb elrendezés tartalmaz még egy decimáló fokozatot is, aminek kettős előnye van, mivel nincs szükség minden ciklusban eredményre, így az adatok tárolására használt regiszterek száma csökkenthető, illetve a túlmintavételezett jel miatt a kimeneti jel frekvenciája csökkenthető, ezzel megelőzve a PC túlterhelését.

A szűrő működése alapján az alábbi egyenlettel írható le [6]

$$y(n) = x(n) - x(n-D) + y(n-1) \tag{2.1}$$

ahol D a comb szakasz késleltetési nagysága, általában megegyezi a decimálási faktorral, R-el. Ennek a z-transzformáltjával megkapható az átviteli függvénye:

$$H(z) = \frac{1-z^{-D}}{1-z^{-1}} \tag{2.2}$$

Ebből az átviteli függvénye frekvencia szerint megkapható a $e^{j2\pi f}$ behelyettesítésével.

$$H_{CIC}\left(e^{j2\pi f}\right) = \frac{1-e^{j2\pi fD}}{1-e^{j2\pi f}} = \frac{e^{-j2\pi fD/2}\left(e^{j2\pi fD/2}-e^{-j2\pi fD/2}\right)}{e^{-j2\pi f/2}\left(e^{j2\pi f/2}-e^{-j2\pi f/2}\right)} \tag{2.3}$$

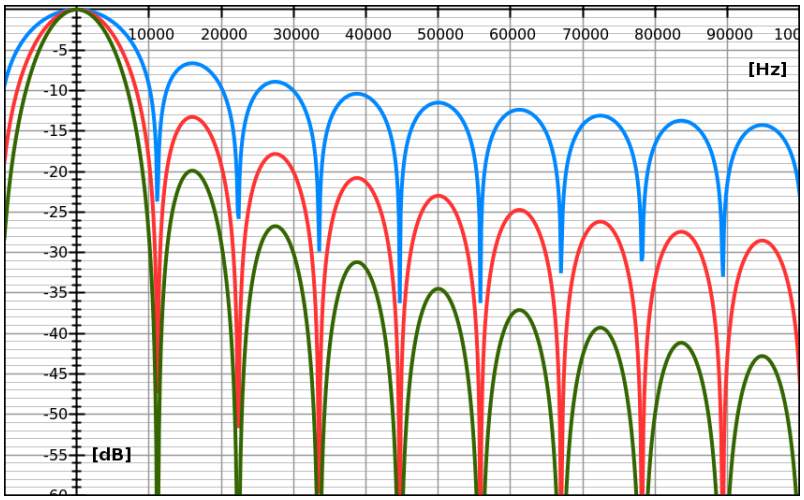
felhasználva, hogy $2jsin(\alpha) = e^{j\alpha} - e^{-j\alpha}$

$$H_{CIC}\left(e^{j2\pi f}\right) = \frac{e^{-j2\pi fD/2}}{e^{-j2\pi f/2}} \frac{2jsin(2\pi fD/2)}{2jsin(2\pi f/2)} = e^{-2j\pi f(D-1)/2)} \frac{sin(\pi fD)}{sin(\pi f/2)} \tag{2.4}$$

A CIC filter egyik nagy előnye, hogy az integráló és fésű fokozatok egymással kaszkádosíthatóak, így sokkal jobb karakterisztika érhető el az elnyomási tartományban.

$$H_{CIC,N-rendű}\left(\left|\frac{sin(\pi fD)}{sin(\pi f)}\right|\right)^N \tag{2.5}$$

Mint ahogy a CIC filter algoritmusából látszik is, az integráló és a fésű szakaszban lévő regiszterek értéke folyamatosan növekszik és nem kerül nullázásra, így előbb



2.4. ábra. Amplitúdó karakterisztika 1. 2., és 3. rendű CIC szűrő esetén 5,71MHz-es mintavételi frekvencián és 512-es decimálási arány mellett

utóbb bekövetkezik a túlcsordulás, más néven overflow jelenség. Az overflow nem jelent problémát, mivel a szűrő kimeneti értékét sosem a regiszterekben tárolt szám értéke, hanem az azok közötti relatív különbség határozza meg, ami overflow előfordulása esetén sem változik. Ha egy ciklus alatt akár az integráló vagy fésű szakaszban kettő vagy több overflow is előfordul már problémát jelenthet, így ezen szakaszoknak a méretét úgy kell megválasztani, hogy ilyen várhatóan ne következzen be [7]. A filterben az egyszerűség kedvéért minden szakaszon U32-es számformátumot választottam, de erre csak azért volt szükség, hogy egy nagy decimálási arány megadása ne jelentsen problémát.

A CIC szűrő jó választás olyan szempontból, hogy könnyen megvalósítható és nem igényel nagy területet az FPGA-n. A szűrő karakterisztikája ugyan elmarad a hagyományos FIR szűrők karakterisztikájától, mivel az átviteli tartományban is jelentős csillapítással rendelkezik, de egyszerűsége miatt jól használható előszűrőként. Az átviteli karakterisztika javítására használható egy kompenzáló FIR filter, ami az eredeti mintavételezés frekvenciáján nehéz lenne megvalósítani, de a decimálási aránnyal redukált frekvencián már könnyebben kivitelezhető. A projekt során ezzel nem foglalkoztam részletesebben, mivel a tesztek nagy része fix frekvenciákon történt, és az egyenletes széles-sávú átviteli karakterisztika nem volt ilyen esetben követelmény.

2.2.3. Kommunikáció, DMA FIFO

Az mikrofonokon mért adatok beolvasásánál fontos, hogy az adatok továbbítása folyamatos legyen, és ne legyen közben adatveszteség. Az FPGA és host-PC közötti kommunikációra több módszer is adott, de ezek a legtöbb esetben nem elegendően gyorsak, vagy nem biztosítják a veszteségmentes adattovábbítást. Ilyen esetekre az egyik leggyakoribb megoldás a FIFO (First In First Out) memória használata, mely afféle pufferként működve biztosítja az aszinkron működés mellett is a folytonos és veszteségmentes adattovábbítást.

A LabVIEW környezeten belül többféle FIFO is definiálható a különböző célokra, a host-PC-vel való kommunikációra DMA FIFO-t használtam. A DMA (Direct Memory Access) mint a neve is mutatja közvetlenül hozzáfér az board-on lévő memóriához, így gyors adattovábbítást tesz lehetővé. Hátránya, hogy csak korlátozott számú csatorna áll rendelkezésre (összesen 3), így nem kaphat minden mikrofon dedikált csatornát. A kommunikációt így egy csatornán keresztül oldottam meg, melyben az adatok a csatornákon egyszerre kerülnek begyűjtésre, majd a az pozíciójuk sorrendjében történnek beírásra a FIFO-ba. A módszer előnye, hogy így a DMA csatornák száma csökkenthető, viszont a kiolvasásnál ügyelni kell arra, hogy mindig a csatornák szám-szorosának megfelelő adat történjen kiolvasásra, más esetben előfordulhatna a csatornák sorrendjének az összekeveredése [8].

A DMA FIFO-ból való kiolvasást optimálisabb nagyobb tömbökben végezni, mivel a FIFO felkészítése az olvasásra jelentős processzoridőt vesz igénybe, így gyakori kiolvasás esetén jelentősen lassíthatja a mérést. A FIFO méretének a megválasztásában ökölszabályként az egyszerre bekérendő adatoknál úgy 10-szer nagyobb méretet választottam, illetve a pontos nagyságot úgy fixáltam, hogy a csatornaszám valamely számszorosa legyen, így a mikrofoncsatornák túlcsordulás esetén nem keverednek össze.

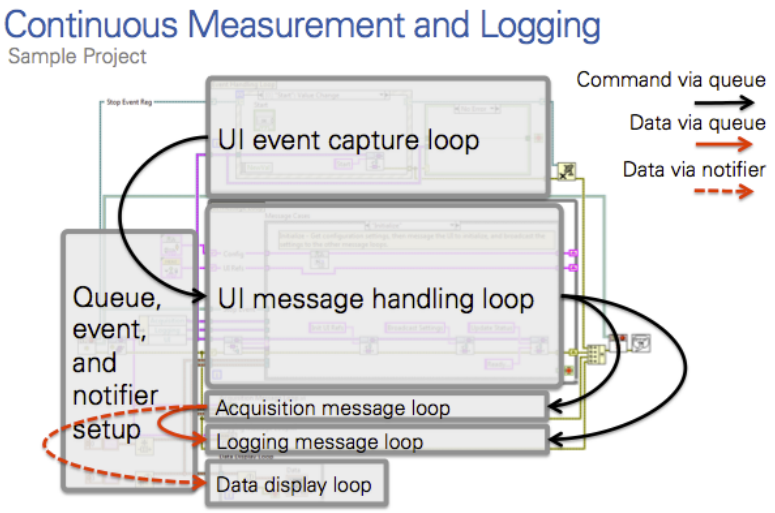
2.3. A host PC-n futó VI

A board és PC között a minden adatforgalom egy ethernet csatlakozón keresztül történik, beleértve a mérési adatok továbbítását és a board felkonfigurálásához szükséges információkat. A PC feladata elsősorban az adatok rögzítése, ezen túl opcionálisan az adatok megjelenítése és feldolgozása. Mivel az utóbbi funkciók jelentősen lassítják a számítógépet, így előfordulhat a FIFO túlcsordulása, ami az adatvesztés veszélye miatt igyekeztem elkerülni.

A program írásakor alapként egy a LabVIEW programhoz mellékelt Continous measurement and logging sablont választottam. A sablon saját dokumentációval rendelkezik, így a diplomamunkában ezzel külön nem foglalkoztam. A méréshez használt program működését tekintve egy állapotgép (2.5 ábra), amelyben a végrehajtásra kerülő állapotokat a felhasználói felületen történő interakciók határozzák meg. A program futása során három process fut párhuzamosan, a mintavételezés, az adatok rögzítése, illetve az adatok feldolgozása, melyek között egy Queued Message Handler (QMH) teremt kapcsolatot. A QMH szerepe kettős, megakadályozza a versenyhelyzet kialakulását a szálak között, illetve eltárolja a felhasználói utasításokat, így minden kiadott utasítás végrehajtásra kerül.

A példaprogram úgy lett megtervezve, hogy könnyen hozzá lehessen igazítani az adott mérőeszközhöz. A program sematikus rajzát a 2.5 tartalmazza.

A méréshez kapcsolódó subVI-okat az Aquisition könyvtár tartalmazza. A mérés vezérlését végző állapotgépben már előre definiálva van az állapotok közötti átmenetek, így a mérőeszköz működtetéséhez elegendő az adott állapotokat leíró VI-ok módosítása. Ezek sablonja a programban már előre definiálva van, névszerint a Acquire, Configure Hardware, Initialize Hardware References és a Stop VI. Mivel a kiolvasás spe-

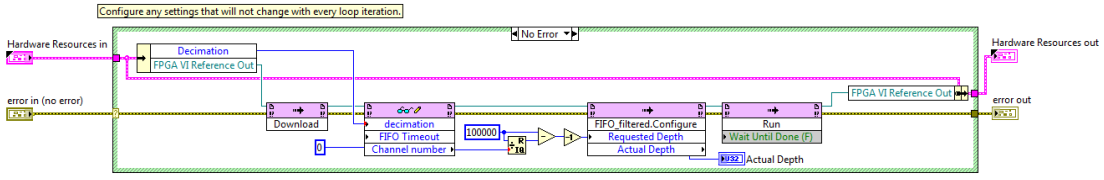


2.5. ábra. A főprogram sematikus rajza

ciális módon történik, így ezt egy külön erre a célra készített Read DMA FIFO VI-ban végeztem.

2.3.1. Configure Hardware VI

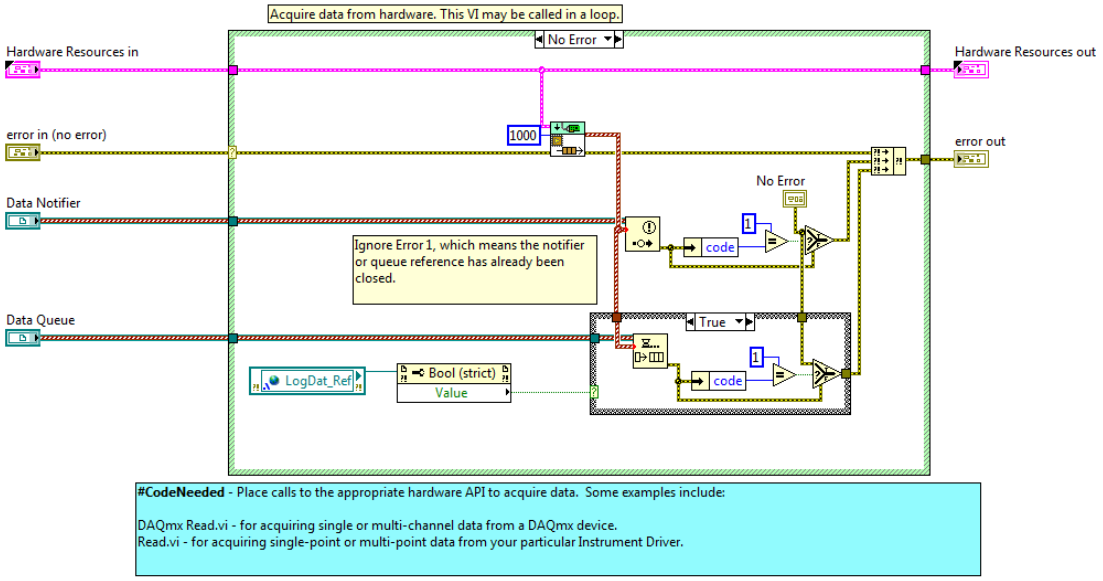
Ebben a VI-ban történik a mérőműszer felkonfigurálása. Az ehhez szükséges adatok az FPGA bitfile-ban lévő regiszterek tárolják. A konfiguráció a start gomb lenyomásakor történik, mérés közben ezek a paraméterek nem módosíthatóak. A konfiguráció során elsősorban a DMA FIFO mélysége és a CIC szűrő ablakszélessége állítható, a konfiguráció elvégeztével automatikusan indul a mérés.



2.6. ábra. Configure Hardware VI

2.3.2. Acquire VI

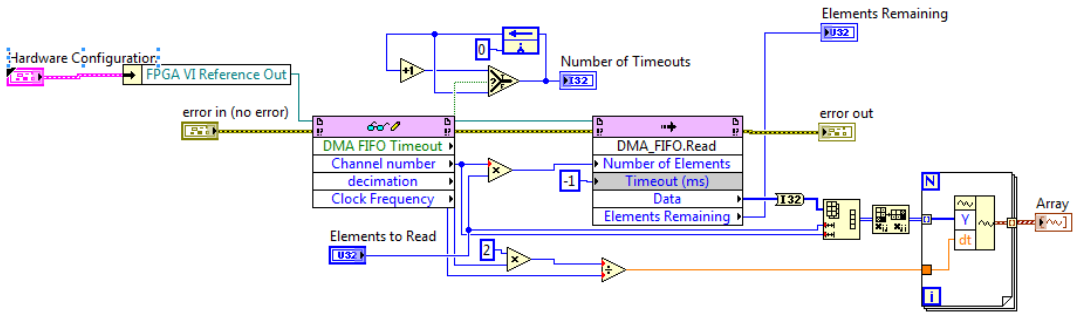
Ez a VI felelős az adatok kinyeréséért az FPGA memóriájából, melyet követően az adat továbbítódik a kijelzést és az adatrögzítést végző modulokhoz. Megadható, hogy mennyi adat történjen kiolvasásra csatornánként, erősen javasolt, hogy a kiolvasott adatok értéke magas (1000) legyen, ennek oka, hogy a memória felkészítése a kiolvasásra időigényes művelet, így érdemes a kiolvasást tömbösítve végezni. Az adatok a kiolvasást követően továbbítódnak a kijelzést és az adatrögzítést végzőt modulokhoz. Az adatok



2.7. ábra. Aquire VI

kijelzése folyamatos, de a rögzítés egy kikapcsolható Ennek oka, hogy a mérés indításakor a filterben lévő regiszterek 0-ról indulnak, és kell nekik egy kis idő, amíg felveszik az átlagos értéküket, így a ‚érés megkezdésekor még fals adatot szolgáltatnak.

2.3.3. Read DMA FIFO VI



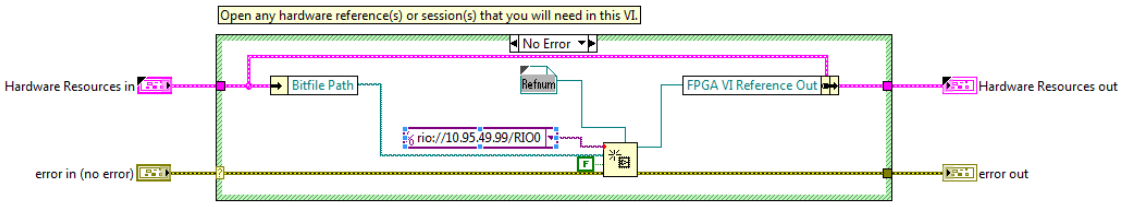
2.8. ábra. Read DMA FIFO VI

Az FPGA-ból történő kiolvasás a DMA FIFO read utasításával történik. A modul automatikusan elvégzi az adatok csatornánkénti szétválasztását a bitfájl paraméterei alapján, és a kimeneten már a csatornánként szétválasztott adat kerül továbbításra. A FIFO timeout paramétere megadja, hogy a program mennyi időt várakozzon (ms-ban) a megfelelő mennyiségű adatra, jelen esetben ez az érték -1, vagyis nincs várakozási

idő. Ebben az esetben a processzor folyamatosan ellenőrzi, hogy van-e elegendő adat, ami okozhatja a CPU túlterhelődését, de ilyesfajta problémát nem tapasztaltam. A beolvasott adatok kimeneti adatformátuma egy a LabVIEW-on belül használt waveform adattípusból álló tömb, mely az adatok mellett eltárolja mintavételek közötti időkülönbséget.

A mérés során ezen a VI-on keresztül ellenőrizhető, hogy a FIFO megtelt-e adattal, melyet a number of timeouts változó t. A FIFO túlcsordulása a *number of timeouts* változón keresztül ellenőrizhető, mely jelzi, hogy a működés során a FIFO hányszor telt meg.

2.3.4. Initialize Hardware References VI

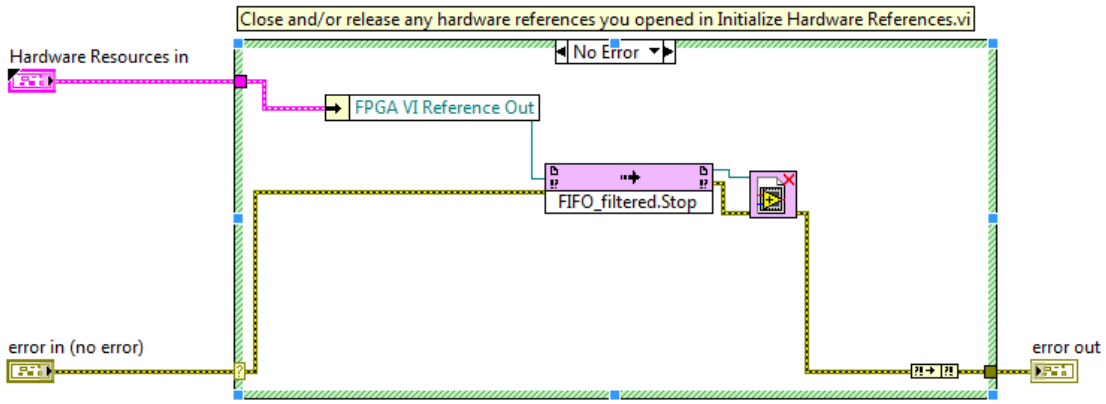


2.9. ábra. Initialize Hardware References VI

Az FPGA-n futó kód dinamikusan van betöltve, ezért inicializáláskor egy speciális változótípussal, a refnum-mal kell megadni, hogy a betöltött bitfájl milyen változókat fog tartalmazni. Ennek a megoldásnak az az előnye, hogy amíg a bitfájlban a kommunikációhoz használt változók azonosak, addig az alatta meghúzódó FPGA kód szabadon változtatható, így például haszbálhat más szűrőt, tartalmazhat extra változókat debuggoláshoz stb.

A változók, melyeket a refnumnak tartalmaznia kell:

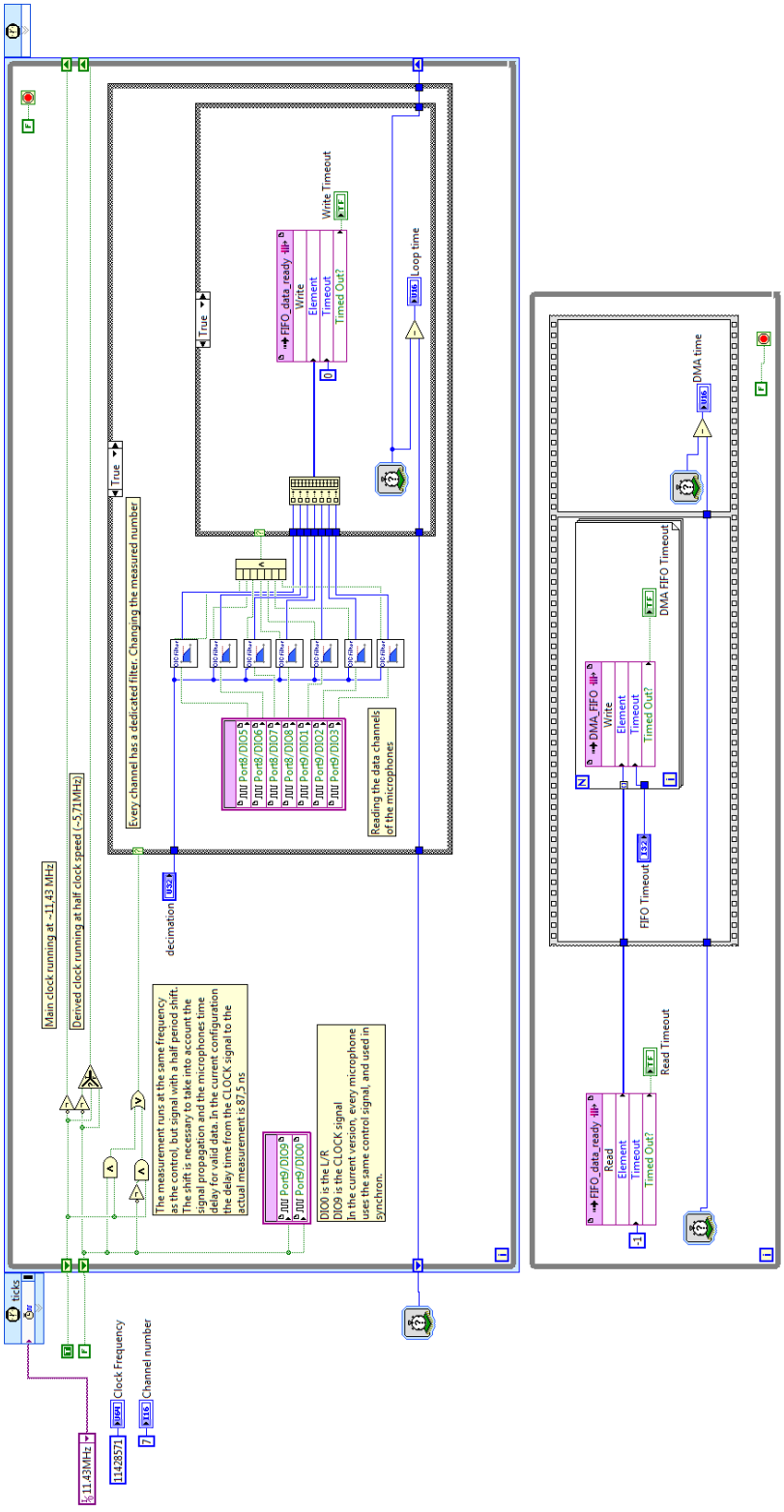
- DMA FIFO: ez szükséges az adatok továbbításáért, mivel a LabVIEW név alapján azonosítja a FIFO-t, így ez nem változtatható
- Number of channels: ez adja meg az FPGA által kezelt mikrofonok számát, a DMA FIFO-ból kiolvasandó adatok mennyiségének a meghatározásához szükséges
- Clock frequency: az FPGA fő órajelének a frekvenciáját adja meg (konstans, a programból nem módosítható)
- decimation: a decimálási arányt adja meg
- DMA FIFO Timeout: szükséges a FIFO ellenőrzéséhez, hogy megtelt-e



2.10. ábra. Stop VI

2.3.5. Stop VI

A VI megállítja az FPGA-t és egyben reseteli is (minden érték a default értéket veszi fel). Az újabb mérés megkezdésekor a program újra betölti a megadott FPGA fájlt, így lehetőség nyílik különböző FPGA bitfájlok futtatására is.



2.11. ábra. Az FPGA-ra írt teljes VI

3 | A szenzorrendszerek alapjai

A projekt célkitűzése az volt, hogy létrejöjjön egy olyan platform, melyen alkalmazható és tesztelhető a szenzorrendszerek körében használt módszerek és algoritmusok. Ebben a fejezetben ismertetem a szenzorrendszerek működési elvét, de mivel ez a témakör elég szerteágazó, így a diplomamunkámban csak azokra a területekre fókuszálok, melyeket a tesztelés során gyakorlatban is alkalmaztam.

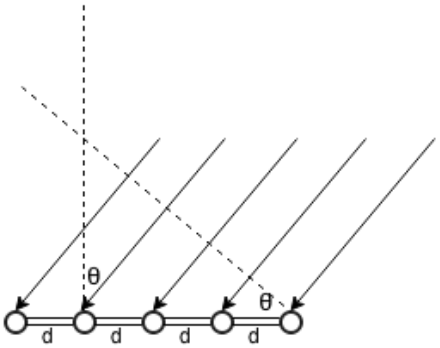
3.1. Elméleti bevezető

A szenzorrendszerekkel való mérés a nyalábformáláson alapul. A nyalábformálás során a detektorok különböző súllyal és fázissal eltolt jelei kerülnek összeadásra, mellyel kihasználva a konstruktív interferenciát, a vétel minősége javítható adott irányban. Az analógia megfordítható, ugyanazt a jelet megfelelő fáziskéséssel leadva a jelforrásokra elérhető, hogy az ily módon létrejövő hullámfront megfelelő irányítottsággal és alakkal rendelkezzen, így koncentrálva a jel intenzitását.

A nyalábformálás gyakorlati haszna, hogy izotrop iránykarakterisztikájú detektorok (pl. dipólantennák) összehangolásával a detektortömb érzékenysége adott térrészbe koncentrálható, így a tér többi irányából beszűrődő zavaró jelek kiszűrhetők. A környezet ismeretében ez a fajta optimalizáció megvalósítható, de gyakori eset, hogy a környezeti paraméterek a mérés során nem ismertek, és a feladat ezek meghatározása. Egyik ilyen paraméter a jelforrás irányának (DOA – Direction of Arrival), melynek meghatározása többféle módon lehetséges, a diplomamunkámban ezek közül vizsgáltam meg néhányat. A DOA megkeresése legtöbb esetben a nyaláb pásztázásával (pl. radar esetében), vagyis a jelek módszeres időbeli eltolásával és összegzésével érhető el. A detektortömbbel megvalósított pásztázás előnye, hogy az ilyen elven működő rendszerek nem tartalmaznak mechanikus mozgó alkatrészt, így sokkal gyorsabb működést biztosítva.

3.2. A megvalósított szenzorrendszer

A megvalósítás során szenzorrendszerek esetében a leggyakoribb és matematikailag legkönnyebben kezelhető elrendezés a soros elrendezés, melyet a diplomamunkámban is alkalmaztam.



3.1. ábra. Mikrofonsorok működésének az alapelve. A mikrofonsor felé érkező hullámfront elsőként a jobboldali mikrofonba érkezik be, majd a Δt időkésés múlva ezt sorban a többi követi.

Az jel a mikrofonokba különböző időkéséssel érkezik, mely arányos a beesési szöggel. Az időkésés az alábbi képlettel számolható:

$$\Delta t_i = \frac{(i-1)d \cdot sin(\Theta)}{c}, i = 1,2,\cdots,M \tag{3.1}$$

ahol c a bejövő jel sebessége

A megvalósított mikrofonrendszer paramétereit a 3.2 táblázat tartalmazza.

3.3. A hangforrás irányának (DOA) meghatározása

A hangforrás irányának a becslése többféle módon történhet. Egyik megközelítés a jelek közvetlen összehasonlításán alapul, ilyen célt szolgál a keresztkorrelációs függvény. Előnye, hogy FFT algoritmussal hatékonyan implementálható. Hátránya, hogy a módszer nem skálázódik jól, mivel ez a módszer csak két mikrofon jelét hasonlítja össze, így a mikrofonok számának a növelésével nő a vizsgálandó kombinációk száma, ami növeli a számításigényt és a komplexitást.

Másik megközelítés a nyalábformálás, amikor nem a mért jelek összehasonlítása történik, hanem azok meghatározott fázissal eltolt és súlyozott változatai. A nyalábformálás során ezen paraméterek úgy vannak megválasztva, hogy a mikrofonrendszer egy adott pillanatban mindig egy kitüntetett irányban legyen érzékeny, a jelek fázisának a manipulációjával afféle radarként a környezet letapogatható a különböző irányokba.

A jelek közötti fáziseltolódás meghatározása a diszkrét mintavételezés miatt nem valami pontos, mivel a jelek eltolása csak diszkrét időintervallumokban történhet. A soros elrendezés miatt a DOA arányos lesz a mikrofonokon mért időkéséssel (TDOA – Time Difference of Arrival), így a kiértékelés során a DOA-t a szög helyett ennek függvényében adtam meg.

Paraméter	Érték	Név	Számítás
d	20cm	A mikrofonok távolsága	–
f_{eff}	11160Hz	Effektív mintavételi frekvencia. Megadja a mérőrendszer hasznos (szűrést és decimálást követő) mintavételi frekvenciáját	$2f_{CLOCK}/D$
s_{min}	3cm	Minimális térbeli felbontás. Megadja, hogy két mintavétel között a hullámfront mekkora utat tesz meg	c/f_{eff}
Θ_{min}	8,62°	A hullámfrontnak az a minimális beesési szöge, melynél már két szomszédos mikrofonon detektálható a fáziskésés	$arcsin(s_{min}/d)$
f_{opt}	850Hz	Az optimális hangfrekvencia, melyre a mikrofonrendszer a legérzékenyebb, ekkor a mikrofonok éppen fél hullámhossznyi távolságra vannak egymástól	$c/2d$

3.1. táblázat. A mikrofonrendszer paraméterei. A változók: $f_{CLOCK} = 2,857MHz$ a mikrofon CLOCK órajel, $c = 340m/s$ a hang sebessége, $D = 512$ a teszt során használt decimálási arány.

3.3.1. Delay & Sum

A delay & sum (D&S) a bemutatott módszerek közül a legkevésbé számításigényes és a leginkább flexibilis. A használatához első lépésben 3.1 egyenlet felhasználásával ki kell számítani adott szög esetén a beérkező hullámfront becsült késését a mikrofonokon, majd ki kell számítani az ennek megfelelő késéssel eltolt jelek összegét.

$$y = \frac{1}{M}\sum_{i=1}^{M} x_i(t - \Delta t_i) \tag{3.2}$$

Az így összegzett jelben helyesen becsült irány esetén konstruktív interferencia jön létre. Mivel a teszt során ismertem a jel frekvenciáját, ezért az erősítés nagyságát a úgy határoztam meg, hogy vettem a jel teljesítményspektrumát, majd megvizsgáltam, hogy mekkora amplitúdó tartozik a 850Hz-nek megfelelő komponenshez.

3.3.2. A GCC és GCC PHAT

A keresztkorrelációs algoritmus előnye, hogy tetszőleges frekvenciájú jelre alkalmazható, viszont számításigényesebb és mivel csak két jel összehasonlítását teszi lehetővé kevésbé robusztusabb és skálázható.

Keresztkorreláció x_1 x_2 diszkrét valós jelre:

$$R_{x_1x_2}(m) = E\left[x_1(n) \cdot x_2^*(n-m)\right] \tag{3.3}$$

A keresztkorreláció direkt módon történő számítása a nagyszámú szorzás miatt meglehetősen számításigényes művelet, mely FFT (Fast Fourier Transform) alkalmazásával használatával jelentősen gyorsítható.

$$R_{x_1x_2}(m) = \mathscr{F}^{-1}\left(X_1(\omega)\cdot X_2^*(\omega)\right) \tag{3.4}$$

ahol $X_1(\omega)\cdot X_2(\omega)$ az x_1 x_2 jelek fourier transzformáltjai.

A módszer hátránya, hogy az eredményt több tényező is (hangvisszaverődés, zaj) erősen befolyásolja, ennek a megoldására használható az általánosított keresztkorrelációs módszer, melynél az inverz FFT előtt a jelek egy ablakfügvénnyel vannak szorozva, így a zavaró tényezők részben kiszűrhetők.

$$R_{x_1x_2}^{GCC}(m) = \mathscr{F}^{-1}\left(X_1(\omega)\cdot X_2^*(\omega)\cdot\psi(\omega)\right)) \tag{3.5}$$

ahol a $\psi(\omega)$ függvény az alkalmazott súlyozó függvény

Az egyik leggyakrabban használt súlyozó függvény a PHAT (Phase Transform), melynek előnye, hogy a keresztkorrelációs függvényben az amplitúdók 1-re vannak normálva, így az inverz transzformációkor csak a fázis ad járulékot, ami ideális esetben egy dirac-delta függvény lesz.

$$\psi_{PHAT}(\omega) = \frac{1}{|X_1(\omega)\cdot X_2(\omega)^*|} \tag{3.6}$$

A függvény maximuma megadja a becsült időkésést a két detektor között.

$$\widehat{d}_{PHAT}(i,j) = \underset{d}{argmax}\,\widehat{R}_{PHAT}(d) \tag{3.7}$$

ahol $\widehat{R}_{PHAT}(d)$ az inverz Fourier transzformáltja az előző egyenletnek

3.3.3. Multichannel cross correlation (MCCC)

Az MCCC algoritmus egy nagy számításigényű módszer, ami egyaránt használható szélessávú és fix frekvenciájú jelek vizsgálatához, és a detektorok számával is jól skálázódik. Soros elrendezésű, L mikrofont tartalmazó detektorrendszer esetén a módszer a következő:

Legyen f_l a mikrofonok közötti késést megadó függvény:

$$f_l = (l-1)\tau \tag{3.8}$$

ahol τ az időkésés két szomszédos mikrofon között. Az MCCC algoritmus alkalmazásához a mérési adatokból képezhető egy kiigazított idővektor, melynek az elemei a csatornákon mért jeleknek az elvárt f_l időkéséssel kiigazított jelei.

$$\mathbf{x}_{1:L}\left[x_1\left[n-f_L(m)+f_1(m)\right],\ x_2\left[n-f_L(m)+f_2(m)\right]\ \cdots\ x_L[n]\right]^T \tag{3.9}$$

ahol a mikrofonok közötti becsült időkésés $\widehat{\tau} = m/f_s$, f_s pedig a mintavételezési frekvencia. Az ennek megfelelő térbeli korrelációs mátrix

$$\mathbf{R}_{m,1:L} = \mathbb{E}\left\{\mathbf{x}_{1:L}\left[n-f_L(m)\right]\cdot\mathbf{x}_{1:L}^T\left[n-f_L(m)\right]\right\} = \begin{bmatrix} r_{m,11} & \cdots & r_{m,1L} \\ \vdots & \ddots & \vdots \\ r_{m,L1} & \cdots & r_{m,LL} \end{bmatrix} \tag{3.10}$$

melyben a két jel közötti keresztkorrelációt az alábbi egyenlet alapján kapható meg.

$$r_{m,kl} = \mathbb{E}\left\{x_k\left[n - f_l(m)\right] x_l\left[n - f_k(m)\right]\right\} \tag{3.11}$$

A térbeli korrelációs mátrix diagonalizálható

$$\mathbf{R}_{m,1:L} = \mathbf{D}\widetilde{\mathbf{R}}_{m,1:L}\mathbf{D} \tag{3.12}$$

ahol **D** diagonális mátrix

$$\mathbf{D} = \begin{bmatrix} \sqrt{\mathbb{E}\left\{x_1^2[n]\right\}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\mathbb{E}\left\{x_L^2[n]\right\}} \end{bmatrix} \tag{3.13}$$

a szimmetrikus mátrix pedig

$$\widetilde{\mathbf{R}}_{m,1:L} = \begin{bmatrix} 1 & \cdots & \rho_{m,1L} \\ \vdots & \ddots & \vdots \\ \rho_{m,L1} & \cdots & 1 \end{bmatrix} \tag{3.14}$$

melyben a keresztkorrelációs együttható $x_k\left[n - f_l(m)\right]$ és $x_l\left[n - f_k(m)\right]$ csatornák között

$$\rho_{m,kl} = \frac{\mathbb{E}\left\{x_k\left[n - f_l(m)\right] x_l\left[n - f_k(m)\right]\right\}}{\sqrt{\mathbb{E}\left\{x_k^2[n]\right\}\mathbb{E}\left\{x_l^2[n]\right\}}} \tag{3.15}$$

ahol k és $l = 1, 2, \cdots, L$.

két csatorna közötti keresztkorrelációs együtthatót a

$$\rho_{m,12}^2 = 1 - \det\widetilde{\mathbf{R}}_{m,1:2} \tag{3.16}$$

egyenlet adja meg. Hasonlóan, a többcsatornás keresztkorrelációs együttható definíciója

$$\rho_{m,1L}^2 = 1 - \det\widetilde{\mathbf{R}}_{m,1:L} \tag{3.17}$$

Ezt az egyenletet felhasználva a mikrofonok közötti időkésés becsülhető. Ehhez meg kell kreálni a mért jelekből a mintavételezésenként vett eltolt jelet, és ki kell számítani az így képzett térbeli korrelációs mátrixot, majd ennek kell számítani a keresztkorrelációs együtthatóját. A legvalószínűbb időkésés a keresztkorrelációs együttható maximumában, vagyis a térbeli korrelációs mátrix determinánsának a minimumában van.

4 | A mikrofonrendszer tesztje

A mérés célja az volt, hogy ellenőrizzem a megvalósított mikrofonrendszer működését, milyen pontossággal lehet megbecsülni az irányt, és hogy melyik módszer használható a leginkább.

4.1. A mérési elrendezés

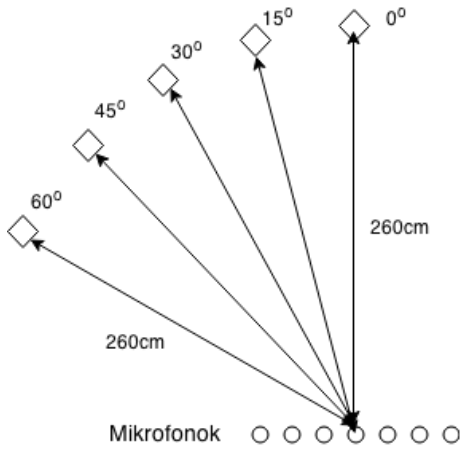
A tesztkörnyezet megtervezésekor figyelembe kellett vennem, hogy a mikrofonok a hangszóróhoz még viszonylag közel helyezkednek el, így a síkhullámos közelítés helyett a jel inkább egy pontforrás hullámfrontjához közelít. Ahhoz hogy az ebből fakadó hiba ne legyen túl nagy, a hangszóró távolságát úgy választottam meg, hogy a detektált hullámfront eltérése az ideális elméleti modelltől ne legyen nagyobb 1 mintavételnél. Ebből kifolyólag a 7 mikrofon helyett csak a középső 5 mikrofon jelét használtam a kiértékelés során, mivel a szélső mikrofonok jele már túlságosan eltértek az elméleti modelltől várt értékektől.

A kísérlet során a 3.2 táblázat szerinti beállításokat alkalmaztam.. A hangforrás és a középső mikrofon távolságát 260cm-nek választottam, mivel ennél a távolságnál a középső és a legszélső mikrofonba érkező hullámfront időkülönbsége nem nagyobb egy mintavételnél. A mérés során különböző szögből indítottam egy 850Hz-es teszt jelet, melynek beérkezési idejét vizsgáltam a mikrofonokon. A mérési környezet nem volt teljesen zajmentes, de a mérés során az nem okozott számottevő hibát.

4.2. Mérési eredmények

A mérés során a mikrofonokkal rögzítettem mintegy 5 másodpercnyi mintát, majd az adatokat PC-n kiértékeltem. Megoldható lett volna a mérés közbeni kiértékelés is, de elsősorban az eredmények reprodukálhatóságára törekedtem, ami egy folyamatos méréssel nem volt biztosítható.

A mikrofonokból beérkező jelet a 4.3 ábra tartalmazza. A jelen látható, hogy jelentős ofszettel rendelkezik (megközelítőleg 131072 = $D^2/2$). Ennek oka, hogy a ΣΔ ADC a bejövő DC jelet váltakozó 0-1 sorozattá alakítja, így egy 512 bitszélességű átlagolással előállított jel DC ofszetje 256-nál lenne. Mivel az FPGA tartalmaz egy másodrendű CIC szűrőt, így az offset ennek megfelelően a mozgóátlag négyzetével lesz arányos (a DC jel nagysága a szűrő rendjének a hatványával emelkedik).



4.1. ábra. A mérési elrendezés

A jeleket további feldolgozás előtt 0-s offset-re hoztam, és 1-re normáltam, így a kiértékelés során mindegyik jel ugyanolyan súllyal fog szerepelni. Ez a lépés egyébként nem feltétlenül szükséges a kiértékeléshez, a tapasztalatom az volt, hogy a Delay& Sum és az MCCC módszer működését nem befolyásolta jelentősen, csak a GCC algoritmus teszteléséhez volt ehhez feltétlenül szükség.

A kiértékelések során az X tengelyen nem a becsült beérkezési szög, hanem a becsült időkésés van feltüntetve, melynél 1 egység egy mintavétel idejének felel meg. Átszámítva 1 mintavételi különbség megközelítőleg 8°-os kitérési szögnek felel meg.

4.2.1. Delay& Sum algoritmus eredményei

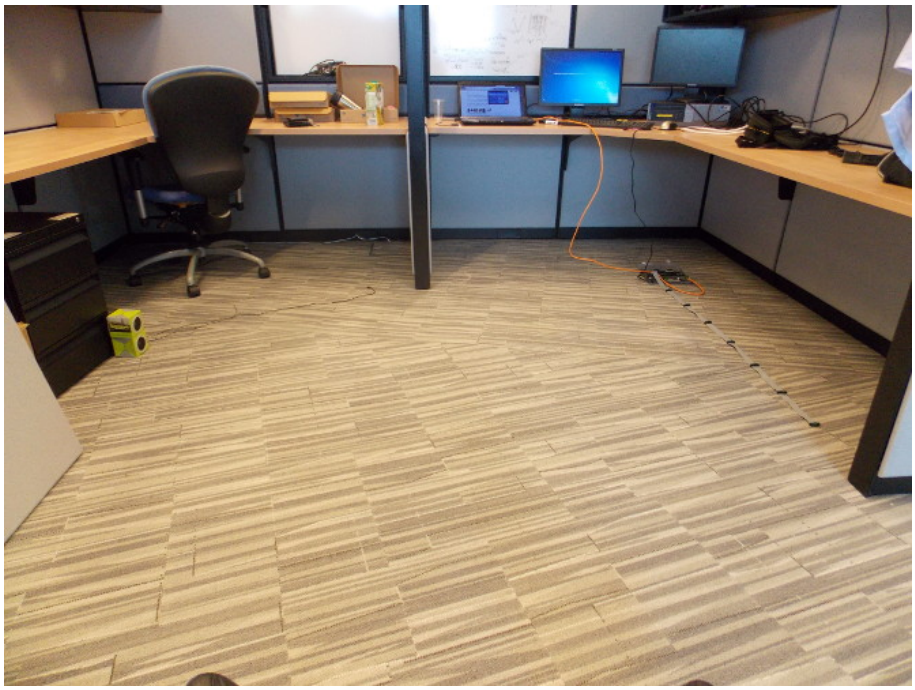
Az legegyszerűbb vizsgálati módszer a Delay& Sum módszer, melynél a csatornák időben eltolt jele kerül összeadásra. Az elvárás az, hogy a feltételezett DOA-knak megfelelő fáziskéséssel eltolva a jeleket és azokat összeadva konstruktív interferencia jön létre, melynek a maximuma a bejövő jel irányában lesz. Az interferencia nagyságát az összegzett jel fourier spektrumából állapítottam meg, melyben a jelnek megfelelő 850Hz-es csúcs nagyságát vettem alapul. Az eredményeket a 4.5 tartalmazza.

Az eredmények alapján jól latható, hogy az optimális időkésés jól elkülönül a többi eredménytől, így ezzel viszonylag jó becslés végezhető. A kiértékelést megismételtem 50 mintára, hogy megvizsgáljam mekkora a legvalószínűbb becsült érték a szórása, melyet a 4.6 ábra tartalmaz.

Az eredmények alapján ez a módszer a számításigényéhez és a komplexitásához képest elég jó eredményt ad, de hátulütője, hogy csak egy frekvenciára ilyen érzékeny, így más frekvenciájú vagy szélessávú jelre már nem lenne ilyen jó az eredmény.

4.2.2. GCC és GCC-Phat algoritmus eredményei

A GCC előnye a Delay& Sum módszerrel szemben, hogy nem csak fix frekvencián, hanem szélessávú jelre (pl emberi beszéd) is használható, viszont jóval számításigényesebb módszer. További hátránya, hogy ezzel a módszerrel egyszerre csak két jel ha-



4.2. ábra. A mérési elrendezés

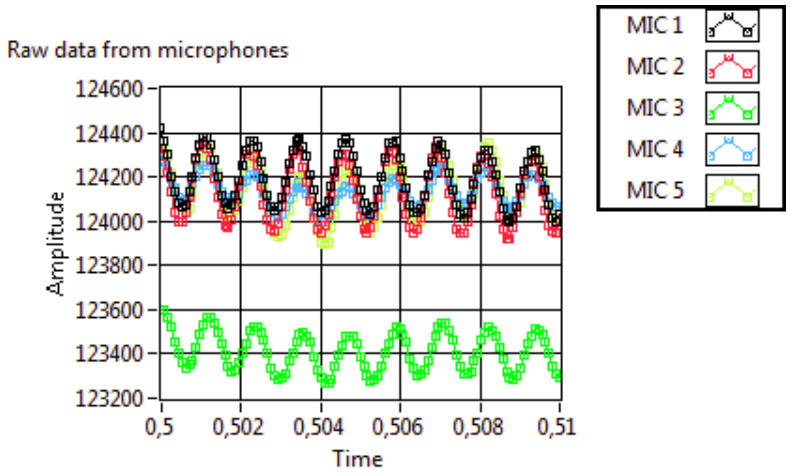
sonlítható össze, így több mikrofonos elrendezésben ennek a kivitelezése minden csatornára már nehézkessé válhat, de kis számú mikrofon esetén ez még nem jelent nagy problémát. Az összehasonlítás során vettem a szélső mikrofont, és keresztkorreláció számításakor ehhez viszonyítottam a többi mikrofon jelét. A mikrofonok keresztkorrelációs függvényei 30°-os beesési szög esetén (4.7).

A módszer megbízhatóságának a vizsgálatához a számítást kiértékeltem 100 különböző mintavételezett jelre 30°-os beesési szögből, és vettem ezek maximumát. Az eredményeket a 4.8 tartalmazza

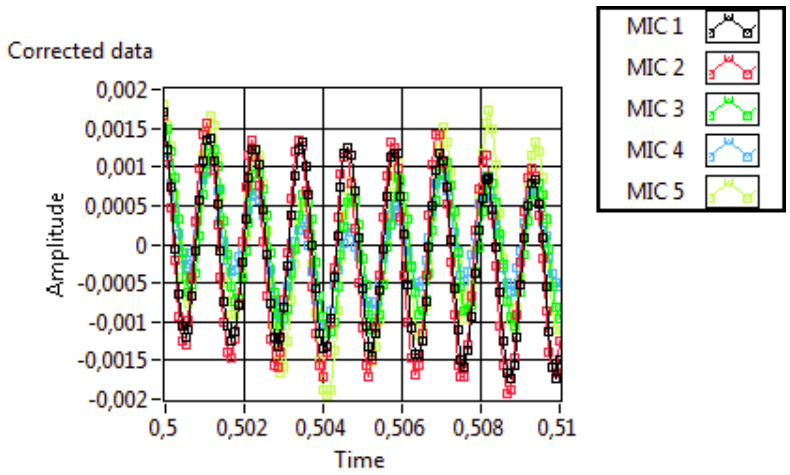
Az eredmények alapján megfigyelhető, hogy amíg a mikrofonok között alacsony az időkésés (szomszédos mikrofonok) a módszer jó eredmény ad, viszont a késleltetés növekedésével a módszer megbízhatatlanná válik. Ennek oka, hogy amikor a késleltetés eléri a hullámhossz felét, ami 6-7 mintavételnyi időnek felel meg, akkor a keresztkorreláció szimmetrikus volta miatt az időkésés "átlapolódik" az ellenkező irányba, ami a kiértékelés során annyit tesz, hogy +45°-os és a -45°-os feltételezett irányt az algoritmus nem tudja megkülönböztetni .

A GCC algoritmusból fakadó bizonytalanság elkerülésére használható a GCC-PHAT algoritmus, ami tekinthető egy szűrt GCC számításnak. Lényege, hogy a jel Fourier transzformáltjában a frekvenciakomponensek járuléka 1-re van normálva, így inverz transzformációkor csak a jelek fázisa ad járulékot, ideális esetben egy dirac-delta függvényt adva eredményül. Mivel ez az algoritmus elsősorban szélessávú jelek esetében használatos, így ennek ellenőrzésére végeztem egy tesztet beszédhanggal 45°-os beesési szögből.

Az eredmények alapján beszédhanggal az időeltolásnak megfelelő csúcsok jól kivehetőek, így szélessávú hangforrás esetében ez a módszer jól alkalmazható.



4.3. ábra. A feldolgozatlan mérési adat 0° os beesési szög mellett



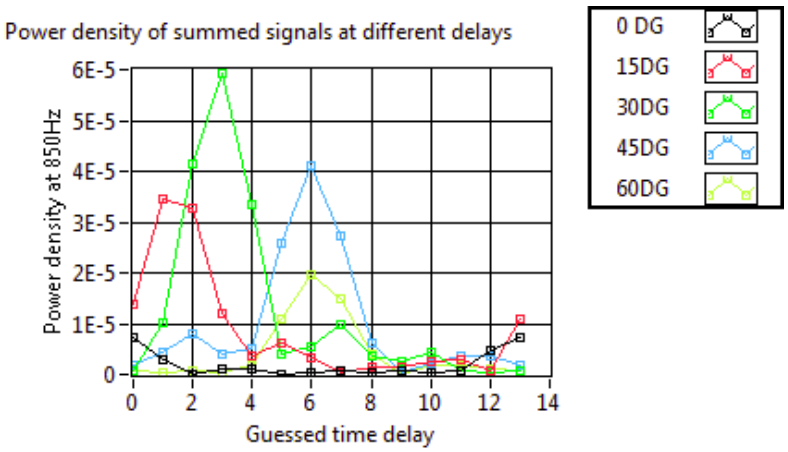
4.4. ábra. A korrigált mérési adat

4.2.3. MCCC algoritmus eredményei

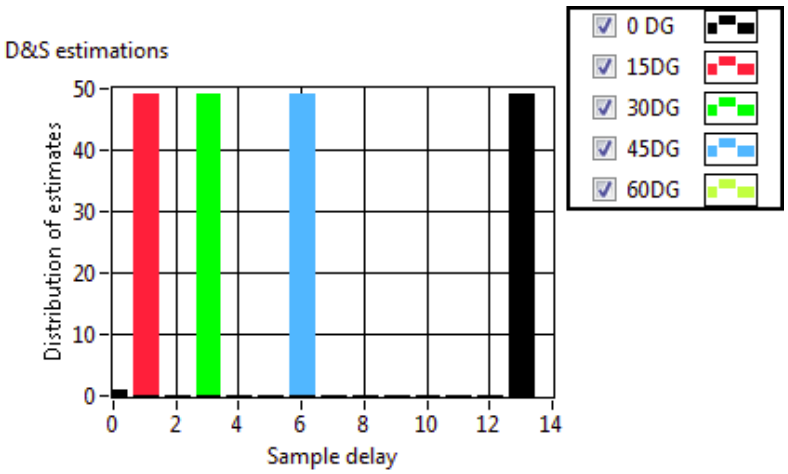
A tesztelt módszerek közül ez volt a leginkább számításigényes. Hasonlóan a Delay& Sum módszerhez a feltételezett iránynak megfelelően időben eltolt jelekkel dolgozik. A módszer lényege, hogy a csatornák eltolt jeléből képez egy idővektort, melynek nagysága a csatornák számával egyenlő és a csatornánként egy időpillanatban mért jeleket tartalmazza. Ebből a vektorból képezhető egy kovariancia-mátrix, melynek determinánsa a legvalószínűbb időkésénél minimumot vesz fel.

Az ábrán jól kivehető, hogy 45° szög környékén a becslés megbízhatósága jelentősen lecsökken, mivel minimum érték már nem különül el olyan jól, mint kisebb szögeknél. Az algoritmus megbízhatóságának az ellenőrzésére ennél is lefuttattam egy 100 mintából álló tesztet, melynek eredményeit a 4.11 tartalmazza.

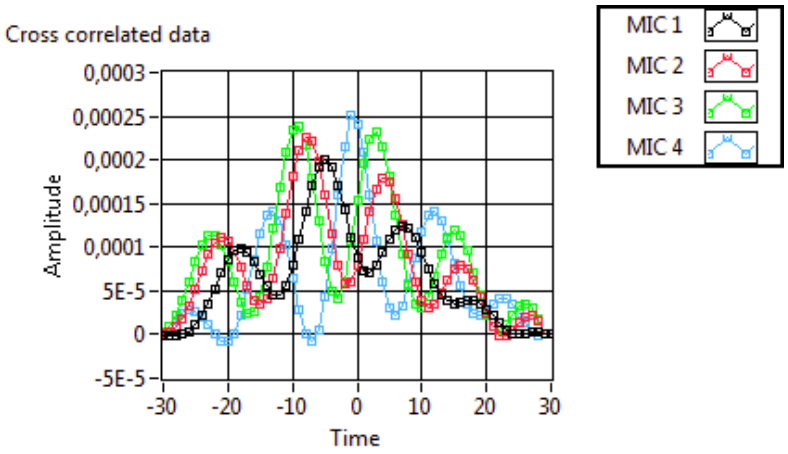
Az eredmények közül a 30°-os mért adatok egy kicsit eltér az elvárhatótól, de mivel a mérőeszköz bizonytalansága mintegy 8°, ezért ez még nem túlzottan nagy hiba.



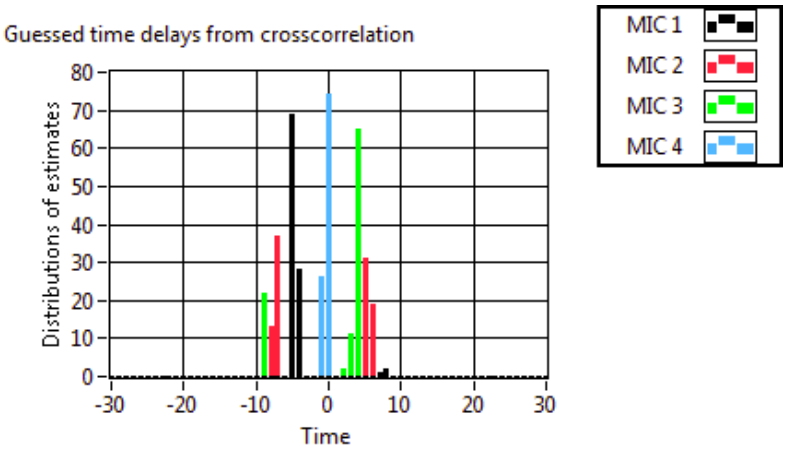
4.5. ábra. A Delay & sum módszerrel kapott eredmények különböző időkésésekre különböző beesési szög mellett



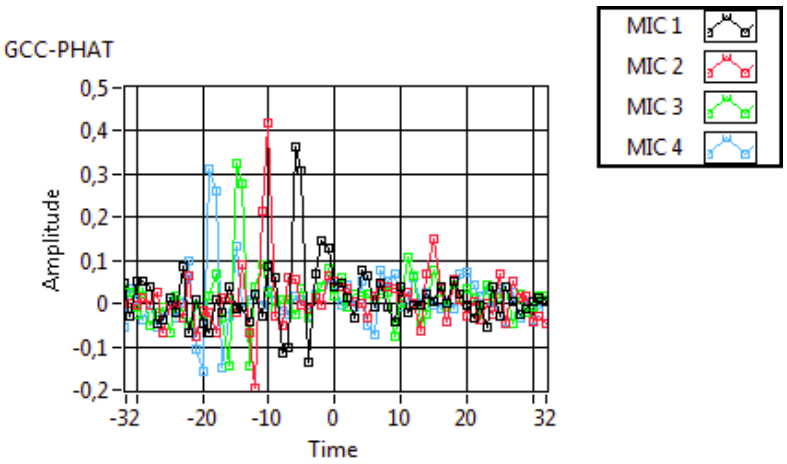
4.6. ábra. Delay & sum módszerrel becsült értékek megoszlása 50 minta mellett



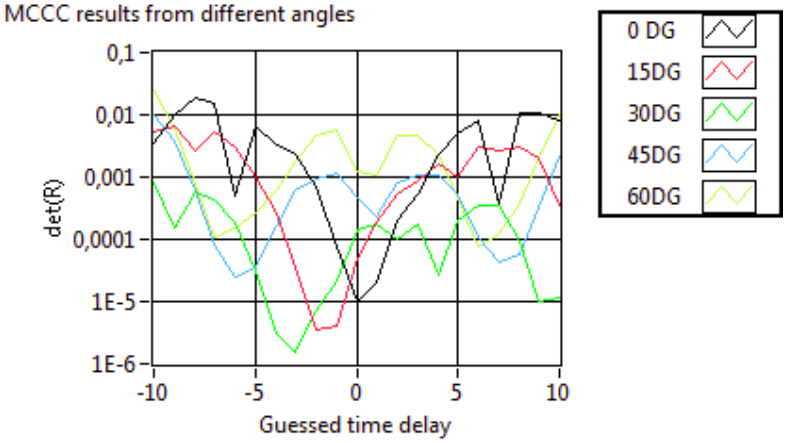
4.7. ábra. Keresztkorrelációs algoritmussal kapott eredmények 30°-os beesési szög mellett



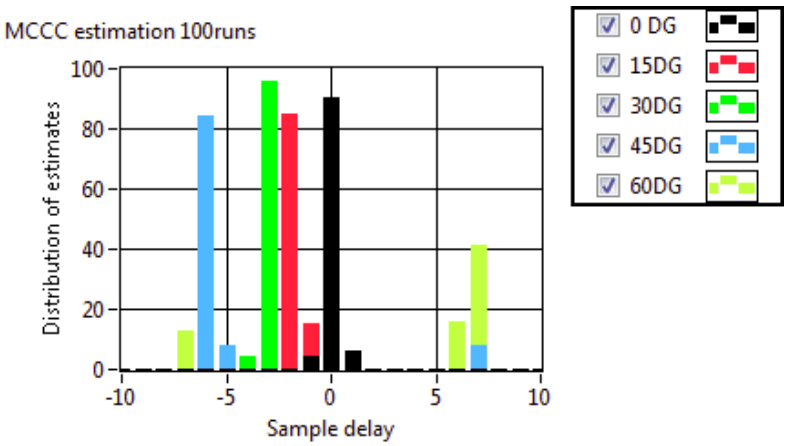
4.8. ábra. Keresztkorrelációs algoritmussal becsült időkésés megoszlása 100 mintával 30°-os beesési szög mellett



4.9. ábra. GCC-PHAT módszerrel kapott eredmények 45°-os beesési szög mellett



4.10. ábra. Az MCCC algoritmussal kapott eredmények különböző beesési szög esetén.



4.11. ábra. Az MCCC algoritmussal kapott eredmények eloszlása 100 kísérlet esetén.

5 | Összefoglalás

5.1. Eredmények

A diplomamunkám során létrehoztam egy mikrofonokból álló szenzorrendszert, melyen különböző, rádiózásban és ultrahangban is használt fázisvezérléses módszerek tesztelhetőek. A tervezés során szem előtt tartottam a modularitást és a flexibilitást, így a létrehozott rendszer hardveres és szoftveres szinten is könnyen módosítható. A mérőszoftvert úgy terveztem meg, hogy a mérés minél könnyebb és felhasználóbarátabb legyen, illetve hogy lehetőséget adjon különböző módszerek mérés közbeni insitu kipróbálására. A működőképességét azzal demonstráltam, hogy megvizsgáltam a mért jel alapján milyen pontossággal határozható meg a forrás iránya, így afféle hangradarként mennyire funkcióképes a rendszer, mely az elvárt pontosságon belül működőképesnek bizonyult. A kiértékelés során teszteltem néhány algoritmust, melyek különböző tulajdonságokkal (komplexitás, robusztusság, skálázhatóság) rendelkeztek, melyek jó kiindulási alapot jelentenek a rendszer továbbfejlesztése esetén.

5.2. Továbbfejlesztési lehetőségek

A mikrofonrendszer jelenlegi változatában alkalmas a hangradarként való üzemmódra. A felbontása jelenleg 8° környékén van, de ez a szűrőben használt decimálási arány csökkentésével, és ezzel az effektív mintavételezési frekvencia növelésével könnyen orvosolható. A limitáló tényezőt jelen esetben a gyenge PC jelentette, mivel elsősorban ennek lassúsága miatt kellett a decimálási faktort magasan tartani, ami még jócskán csökkenthető a felbontás veszítése nélkül.

A mikrofonok geometriai elrendezés még jelentősen lehet javítani. A fejlesztés során azért ezt a felépítést választottam, mert egyszerű megvalósítani, és matematikailag is könnyű kezelni, de korántsem jelent ideális megoldást. A tesztelés során bebizonyosodott, hogy a jelen elrendezésben 45°-nál beesési szöget A jelenlegi soros elrendezésben nem lehet megbízhatóan detektálni, így érdemes lehet másfajta elrendezést is fontolóra venni. Egyik ilyen megoldás lehet a síkbeli hatszögrács, mert ilyen kialakítással elérhető, hogy a detektortömb valamelyik síkja 45°-nál kisebb szöget zárjon be a hullámfronttal.

Az FPGA erőforrásai a 7 mikrofonos elrendezéshez elegendőnek bizonyult, az erőforrásoknak mintegy 30%-a volt kihasználva. Az FPGA-val való TDOA becslés a PC számára sok erőforrást felszabadítana, de ezzel az FPGA kód jelentősen bonyolódna,

amihez érdemes lehet a későbbiekben egy nagyobb FPGA-ra váltani. A delay & sum módszer egyszerűségéhez képest meglepően jó eredményeket ad, így ha a jövőben fix frekvenciájú jelekkel történne mérés, ez egy jó módszer az FPGA-ra való implementálásra. A többi módszer jóval bonyolultabb, és leginkább szélessávú jelek esetében jelentenek előnyt. Fejlesztés a jövőben mégis a szélessávú jelek detektálása felé tolódna el, akkor érdemes megfontolni a detektálása felé történne, érdemes lehet megfontolni a szűrő karakterisztikájának a javítását valamely FIR vagy SCIC filter implementálásával.

Irodalomjegyzék

[1] Sangil Park, *„Principles of Sigma-Delta Modulation for Analog-to-Digital Converters”* published by motorola
http://www.numerix-dsp.com/appsnotes/APR8-sigma-delta.pdf

[2] Bonnie Baker, *„How delta-sigma ADCs work, Part 1”* Analog Applications Journal, (3Q 2011)
http://www.ti.com/lit/an/slyt423/slyt423.pdf

[3] Bonnie Baker, *„How delta-sigma ADCs work, Part 2”* Analog Applications Journal, (3Q 2011)
http://www.ti.com/lit/an/slyt438/slyt438.pdf

[4] http://en.wikipedia.org/wiki/Pulse-density_modulation

[5] National Instruments „USER GUIDE NI sbRIO-9612XT/9632XT/9642XT Single-Board RIO OEM Devices”
http://www.ni.com/pdf/manuals/375052c.pdf

[6] Richard Lyons *„Understanding cascaded integrator-comb filters”*, Online article, Embedded Systems Programming
http://www.design-reuse.com/articles/10028/understanding-cascaded-integrator-comb-filters.html

[7] Matthew P. Donadio *„CIC Filter Introduction”* whitepaper 18 July 2000
http://home.mit.bme.hu/~kollar/papers/cic.pdf

[8] National Instuments whitepaper „Transferring Multi-Channel Data in DMA Applications (FPGA Module)”
http://zone.ni.com/reference/en-XX/help/371599H-01/lvfpgaconcepts/fpga_dma_fifo_interleave/

[9] Velasco, J, Taghizadeh, Mohammad J., Asaei, Afsaneh: *„Novel GCC-PHAT Model in Diffuse Sound Field for Microphone Array Pairwise Distance Based Calibration”*, Conference paper, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015
http://publications.idiap.ch/downloads/papers/2015/Velasco_ICASSP15_2015.pdf

[10] Udo Klein *„Direction-of-Arrival Estimation Using a Microphone Array with the Multichannel Cross-Correlation Method”* IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2012
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6621296&tag=1

[11] Knowles electronics „SPM1437HM4H-B Digital microphone datasheet”
http://www.knowles.com/download/file?p=SPM1437HM4H.pdf

Függelék

A mikrofonrendszer használata

A méréshez szükséges eszközök:

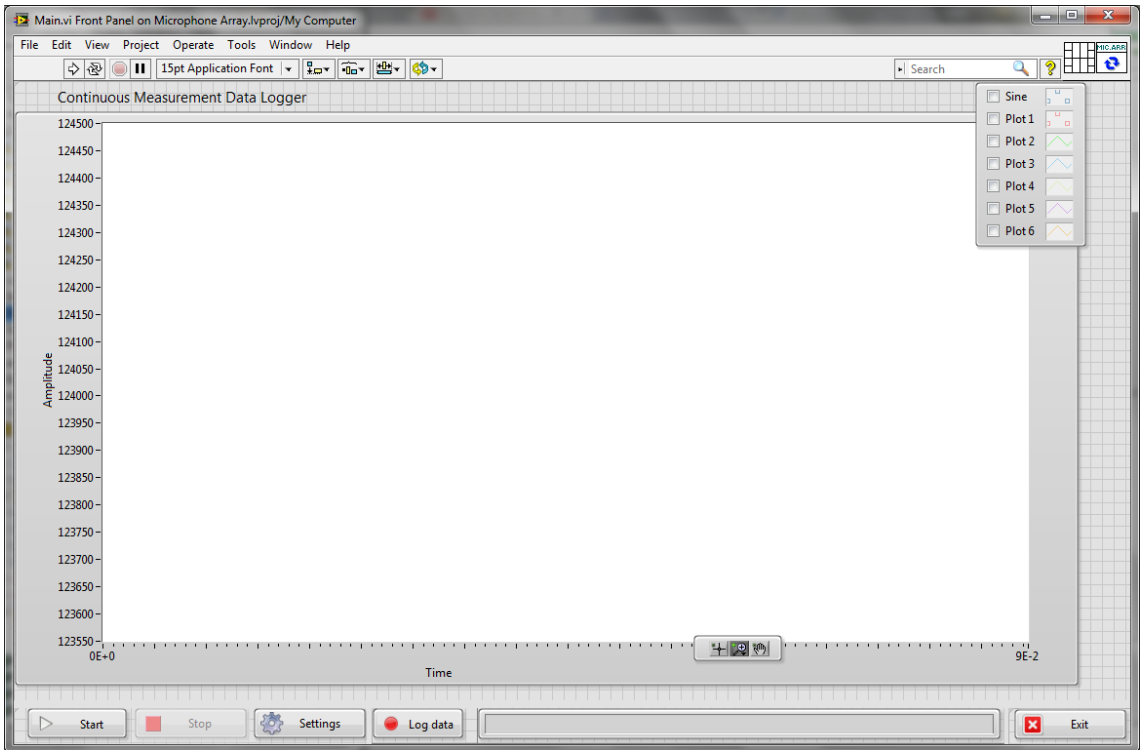
- PC a mérőszoftverrel
- UTP kábel, ami összeköti az sbRIO-t a PC-vel, (lehetőleg minél hosszabb a mérőeszköz könnyebb mozgatása végett)
- sbRIO board
- mikrofonsor, ami az sbRIO DIO kimeneteire kell kötni
- 19-30 VDC adapter sz sbRIO és a mikrofonok áramellátásához

A PC-n az alábbi szoftverek szükségesek a program használatához:

- LabVIEW
- LabVIEW Real-Time module
- LabVIEW FPGA module
- NI-RIO driver

Ezek hiánytalan megléte esetén összeállítása után és az sbRIO-t áram alá helyezve a mérőrendszer mérésre alkalmas. Ha a mérés indítása után a program hibaüzenettel kilép, érdemes lehet ellenőrizni az alábbi hibalehetőségeket:

- A sbRIO IP címe. Mivel az sbRIO csatlakoztatása a PC-re lokális hálózaton történik, ezért előfordulhat, hogy az sbRIO másik IP-címet kap, mint ami a programon belül eg van adva. Az IP cím a MAX-on keresztül ellenőrizhető, és a poject explorerben megadott értkre módosítható.
- A program megfelelően be van állítva, a decimálási arány nem 0 (default 512), illetve érvényes bitfájl és elérési útvonal van megadva.
- A board LEDje 1s-ként villog, ami azt jelenti hogy a board-nak nincs érvényes IP címe, NI MAX-on (Measurement and automation explorer) keresztül megfelelően beállítható, NIMAX->Remote Syystem->network settings (további segítség http://www.ni.com/getting-started/set-up-hardware/compactrio/first-use).



F.1. ábra. A program felhasználói felülete

- A board LED-je 2s-ként villogm ami azt jelenti, hogy a board nincs megfelelően felkonfigurálva, vagyis nincsenek rajta a működéséhez szükséges driverek. Ezek hiányában a board-on lévő LED mintegy 2s-ként villog, jelezve, hogy a board helytelenül van felkonfigurálva

A program használata

A gombok funkciói:

- PLAY: Ezzel a gommbal indítható az adatgyűjtés a mikrofonokról. A mérés során a mért jel ekkor csak kijelzésre kerül, de az adatok rögzítése még nem történik meg.
- STOP: A mérés ezzel a gombbal állítható le, mely után új paraméterek adhatóak meg a SETTINGS menüben, majd a PLAY gombbal a mérés újraindítható.
- SETTINGS: Itt állíthatóak be a mérésre vonatkozó paraméterek. A bitfájlok az FPGA Bitfiles mappában találhatóak.
- LOG DATA: A mérés indításakor még nem kerülnek rögzítésre az adatok (csak egy üres tdms fájl lesz megnyitva, amibe majd folyamtosan történik az adatok rögzítése). A LOG DATA lenyomásától számítva kerülnek rögzítésre az adatok a gomb újbóli megnyomásáig

- EXIT: bezárja a programot

A mérés elindítása után, mielőtt az adatok rögzítésre kerülnének, érdemes ellenőrizni, hogy valóban folyamatos -e az adatok kiolvasása, nincsenek -e túlterhelődve a FIFO-k. Ez esetben érdemes a decimálási arányt olyanra beállítani, amelynél már feldolgozható ütemben érkezik az adat a PC-re. További gyorsulás érhető el, ha a mért jel kirajzolása kikapcsolásra kerül. Az adatok rögzítése után a mérés leállításával (STOP) új mérési paraméterek adhatóak meg, így két mérés között nem szükséges a program bezárása.

Ábrák jegyzéke

Rövidítések

ADC	Analog to Digital Converter
BME	Budapesti Mszaki Egyetem
CIC	Combined Integratodr Comb
DAC	Digital to Analog Converter
DFT	Discrete Fourier Transformation
DMA	Direct Memory Access
DOA	Direction of Arrival
DSP	Digital Signal Processor
FFT	Fast Fourier Transformation
FIR	Finite Impulse Response (Filter)
FIFO	First In First Out
FPGA	Field Programmable Gate Array
GCC	Generalized Cross Correlation
GUI	Graphical User Interface
IIR	Infinite Impulse Response (Filter)
I/O	Input-Output
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LSB	Least Significant Bit
MCCC	Multi Channel Cross Correlation
MEMS	Micr-electro Mechanical System
PCM	Pulse Code Modulation
PHAT	Phase Transform
PDM	Pulse Density Modulation
SCIC	Sharpened CIC
SNR	Signal to Noise Ratio
TDOA	Time Delay of Arrival