



Polimertechnika Tanszék

TDK DOLGOZAT

ANYAGOK HANGJA

Készítette: Marczis László Péter

Herbay Máté

mechatronikai mérnök

BSc. hallgatók

Konzulens: Dr. Czigány Tibor

Egyetemi tanár

Dr. Lipovszki György

Egyetemi docens

TARTALOMJEGYZÉK

A dolgozatban használt jelölések, rövidítések	3
1. Bevezetés, a TDK dolgozat célja	4
2. Irodalmi áttekintés	7
2.1. Az emberi hallás	8
2.2. Az akusztikus emissziós mérés	9
2.2.1. Az akusztikus emissziós mérés története	9
2.2.2. Az akusztikus emisszió elvi háttere	9
2.2.3. Az akusztikus emisszió méréstechnikája	11
3. Az akusztikus emissziós feldolgozó program	14
3.1. A program koncepciója	14
3.2. A forrásfájl	15
3.3. A főprogram	16
3.3.1. A főprogram kezelőfelülete	16
3.3.2. A főprogram blokk diagramja	21
3.4. A program működése	30
4. Mérések	57
5. Futási eredmények	60
5.1. Beállítások hatásának bemutatása	60
5.2. Mérési eredmények kiértékelése	65
5.3. Konklúzió	71
6. Összefoglalás.....	72
7. Irodalomjegyzék.....	74
CD Melléklet	

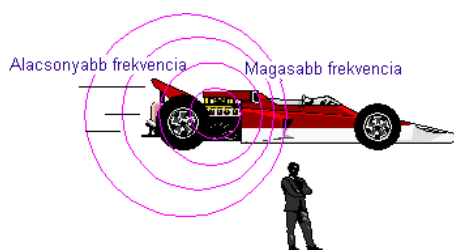
JELÖLÉS- ÉS RÖVIDÍTÉSJEGYZÉK

Jelölés	Mértékegység	Név
A	[dB]	amplitúdó decibelben
A'	[μV]	új amplitúdó mikro voltban
A _{max}	[mm]	harmonikus rezgés - amplitúdó
E	[μJ]	energia
r	[mm]	gömbhullám sugara
R ₀	[Ω]	műszer állandó
t	[s]	idő
U	[μV]	maximum amplitúdó
U ₀	[μV]	decibel alapérték
U _{alap}	[μV]	decibel alapérték, konstans 1
y	[mm]	harmonikus rezgés - részecske elmozdulása
φ	[rad]	harmonikus rezgés - fázisszög
ω	[rad/s]	harmonikus rezgés - körfrekvencia

Rövidítés	Jelentés
AE	akusztikus emisszió
PP	polipropilén
PET	polietilén-tereflatát
vi	„Virtual Instrument”, a labVIEW fejlesztői környezet programjának elnevezése, illetve kiterjesztése
subVI	LabVIEW program, mely egy másik programba lett beágyazva (alprogram)
Szamol.vi	A programban használt vi jelölése
időegység	A programban használt változó jelölése

1. Bevezetés, a TDK dolgozat célja

A hallás az emberi érzékelés egy fontos része, mely segítségével sok információhoz juthatunk. Mivel a hang bármely irányból hozzánk juthat (szemben a látással, mely hozzávetőleg 120 fokban érzékel [8]), ezért általában a hangok alkalmasabbak a nem várt veszélyek jelzésére. Filmekből ismerős, hogy a folyókon utazók általában a hangja alapján veszik észre először a vízesést; a közlekedésben, üzemekben gyakran alkalmaznak hangot figyelemfelkeltésére (szirénák, hajókürtök, dudák stb.). A zajokból azonban további információkhoz is juthatunk: a lesújtó villám fénye és a dörgés között eltelt időből becsülhetjük a becsapás távolságát, a mozgó járművek hangjának magasságából megállapíthatjuk tőlünk távolodik-e, de becslést adhatunk sebességéről is (Doppler effektus, 1-1. ábra). [3]



1-1. ÁBRA DOPPLER EFFEKTUS

Már ősidők óta ismert, hogy a különböző anyagok belső szerkezetének megváltozása hangokat kelthet, s ez jelzés értékű lehet. A fa ropogása-recsegése miközben összetörik; a hegyomlás előtti kőzetekből származó hangok (1-2. ábra); az üveg ropogása, mialatt reped; a fémek hangja miközben hül.



1-2. ÁBRA HEGYOMLÁSRA FIGYELMEZTETŐ TÁBLA

Az orvostudomány már régóta használja a sztetoszkópot diagnosztikai célokra, de az ultrahang is mindenki számára ismerős. A hangokat létrehozó folyamatok, illetve a hangok által hordozott információk megfejtése régóta foglalkoztatja az emberiséget. Hooke a XVII. század végén leírta, miként lehet következtetni a keletkezett hangokból az őket létrehozó testen belüli mozgásokra [11].

Az 50-es évektől kezdve már képesekké váltunk a hallható tartományon kívüli hangok észlelésére is, mely a hangok által hordozott információk megfejtéséhez egy nagy lépést

jelentett. 1950-ben a Müncheni Műszaki Egyetemen oktató Joseph Kaiser detektált először nem hallható hangot elektronikus készülékével, s egyik legfontosabb megfigyelése, hogy a keletkező hang irreverzibilis eseményt jelölhet [1, 11].

Ma már nagy pontosságú eszközeink, érzékelőink vannak, melyek bőven meghaladják az ember korlátait (elektronmikroszkóp, mely több nagyságrenddel kisebb dolgok észlelésére és megfigyelésére képes, mint az emberi szem; mikrométer, mely az emberi tapintásnál sokkal finomabban képes pozicionálni és tapintani, s így tovább). Azonban mindezek csak az emberi érzékelés kiterjesztésére szolgálnak, ugyanis a bejövő információk feldolgozásának a feladata a legtöbb esetben az emberre hárul. Ugyan mesterséges intelligenciákkal, s szakértői programokkal a műveletek jó részét automatizálhatjuk, de ezen szoftverek korlátai miatt még mindig sok helyen nem hagyható ki az ember a döntés, feldolgozás, értelmezés folyamatából. Ebből kifolyólag ezek az eszközök (mint minden eszköz) még mindig elsősorban az ember érzékelésének meghosszabbításai, s kiszélesítői. Emiatt az eszközök „nyelvét”, az érzékelők által szolgáltatott információkat át kell alakítani az ember számára könnyen feldolgozható adatokká (grafikus ábrázolás, diagramok, mozgó ábrák legtöbbször).

Az akusztikus emisszió témaköre foglalkozik a testekben keletkező rezgésekkel (nem hallható, ultrahang tartománybeli hangokkal), illetve azok detektálásával. A gyűjtött információt mindenképpen át kell „alakítani”, hogy értelmezhesük őket. Az egyik megoldás a grafikus ábrázolás. Ezen TDK azonban egy másik megközelítést próbál feltárni: mégpedig az érzékelt „hangok” hallhatóvá tételét. Ezzel az érzékelt jelek lejátszhatóvá válnak, mely sok várt, s még több feltárássra váró lehetőséggel kecsegtet. Egy üzemben a hallás útján történő kommunikáció általában gyorsabb, mint a szöveges, s esetleg jellegzetesebb (így gyorsabb a megértése) is, mint például a diagram általi ábrázolás.

Az akusztikus emisszió alkalmas anyagok megkülönböztetésére (mely már eszköz nélkül hallható tartományban is alkalmazható, hiszen másképp hangzik törés esetén egy fa, s egy acél gerenda). Azonban ezen módszerrel az anyag belső szerkezetéről is kaphatunk pontosabb képet, hiszen szélesebb skálán érzékeljük a keletkezett hangokat, melyek nem mindig, vagy csak részben esnek bele a hallható tartományba. Így akár arra is alkalmas lehet, hogy megvizsgáljuk, az adott anyagban van-e erősítés, de az anyag homogenitására is fényt deríthetünk. Továbbá már ma is használják hibakeresésre. A mérések hallhatóvá tételével gyorsíthatjuk a vizsgálatokat, hordozhatóbbá tehetjük az eszközt (nincs szükség esetleg kijelzőre), amivel egyúttal gazdaságosabbá is válhat.

A TDK dolgozat célja:

1. Az akusztikus emisszió által rögzített adatok feldolgozásának újfajta megközelítése. Mivel az anyagok összetételétől s belső szerkezetétől is függ az általuk kibocsátott hang, a mérési eredmények hallhatóvá tételével újabb információkat fedezhetünk fel, továbbá gyorsíthatjuk a mérés kiértékelését is.
2. Az anyagok által kibocsátott hangok hallhatóvá tétele (melyek ultrahang tartományban helyezkednek el, s ezáltal nem hallhatóak az emberi fül számára), illetve az erre alkalmas szoftver megalkotása LabVIEW programban.
3. A program olyan módon történő elkészítése, hogy több beállítási lehetőség segítségével, egy mérési eredmény többféleképpen is megvizsgálható legyen (különböző „erősítés”, kiemelés; különböző ablakozás stb.), ezzel lehetőséget hagyva az optimális beállítások megtalálására.
4. Különböző anyagfajták (polimerektől kezdve, acélokon át egészen a természetes anyagokig, mint például a fa) mechanikai terhelése közben keletkezett hanghullámok hallhatóvá tétele, kiértékelése, elemzése és összehasonlítása.



2. Irodalmi áttekintés

A fizikában az ismétlődő folyamatokat rezgéseknek nevezzük. Egy valamilyen közegben terjedő rezgést hullámnak hívunk, és amennyiben ez a rezgés mechanikai (azaz rugalmas alakváltozás terjed tovább), akkor hangnak nevezzük. A továbbiakban ilyen rezgésekkel foglalkozunk.

A rezgéseket többféleképpen csoportosíthatjuk. Lehetnek periodikusak és nem periodikusak. Periodikus rezgések speciális, egyszerű esete a harmonikus rezgés. Ekkor egy pontbeli elmozdulás szinuszos függvénnel írható le az idő függvényében (1):

$$y = A_{\max} \cdot \sin(\omega \cdot t + \varphi) \quad (1)$$

ahol y a részecske elmozdulása egy adott irányba, $A_{\max}$ az amplitúdó, ω a körfrekvencia, φ a fázisszög, t pedig az idő. Ekkor a rezgés láthatóan két jellemzővel leírható: az amplitúdójával és a körfrekvenciájával (a fázisszög a 0 időpont megválasztásától függ). A gyakorlatban azonban a rezgések ritkán harmonikusak. A periodikus rezgések legtöbbször több harmonikus összetevő szuperpozíciójaként állítható elő. Ez azt jelenti, hogy a hangok több különböző frekvenciát tartalmaznak egyszerre. Ha ezen összetevők amplitúdóját és (kör)frekvenciáját diagramon ábrázoljuk, akkor ezt hívjuk a rezgés frekvencia spektrogrammjának, az ezt leíró függvényt pedig frekvenciaspektrumnak. Nem periodikus rezgések lehetnek folyamatosak, illetve tranziensek. Ezek egymáshoz végtelenül közeli frekvenciakomponenseket tartalmaznak, így frekvenciaspektrumuk folytonos. Folyamatos nem periodikus rezgés például a zaj, tranziens pedig a dörej, csattanás. Ez utóbbi rendszerint hirtelen energia felszabadulást jelez, így az ilyen rezgések az akusztikus emissziós méréseknél nagy fontosságúak [2, 3, 7].

Másféle csoportosítás szerint lehet a hullám transzverzális (a közeg részecskéi a terjedési irányra merőlegesen mozognak el) és longitudinális (a részecskék a terjedési iránnyal párhuzamosan mozognak). Transzverzális hullám terjedésének feltétele a közegben létrejövő nyírófeszültség, emiatt ez csak szilárd közegben jöhet létre. Szilárd anyagban tehát az akusztikai hullámok mindkét formában terjedni tudnak, bonyolult rezgéseket kialakítva. Ezzel szemben folyékony és gáznemű közegben csak longitudinális hullámok jöhetnek létre, ekkor a hullám a közeg sűrűsödésének és ritkulásának tovaterjedését jelenti [1, 7].

2.1. Az emberi hallás

Az ember füle a dobhártyára érkező nyomáshullámokat érzékeli. A fülben a rezgés ingerületté válik, és az idegeken keresztül az agyba jut. Az agy feldolgozza ezt, aminek következtében létrejön bennünk a hangérzet. Ezen hangérzet összefüggése az eredeti rezgés fizikai tulajdonságaival azonban korántsem triviális. A hangerő érzete alapvetően a hangnyomás amplitúdójával, illetve a hang intenzitásával (felületre vett teljesítményével) van kapcsolatban. Ez a kapcsolat közelítőleg logaritmikus jellegű, ezért a hang erősségét decibelben (dB) fejezik ki, legtöbbször a hangintenzitás vagy teljesítmény alapján. A hangerő érzetét kis mértékben befolyásolja ezen kívül a hang magassága is [2, 7].

A legkisebb ember által érzékelhető hangnyomás (hallásküszöb) körülbelül $2 \cdot 10^{-5} \text{ N/m}^2$, míg egy 20 N/m^2 nyomású hangot már fájdalomként érzékelünk (fájdalomküszöb). Hangintenzitás tekintetében ezek az értékek 10^{-16} és 10^{-3} W/cm^2 . A hangra használt dB skála alapértékét emiatt $2 \cdot 10^{-5} \text{ N/m}^2$ -nek választották meg, egy ekkora nyomású hanghullámot tekintünk tehát 0 dB-nek. Néhány tipikus hang erőssége [3, 11]:

- | | |
|--|--------|
| • falevél susogása: | 20 dB |
| • beszélgetés: | 60 dB |
| • közlekedés: | 80 dB |
| • légkalapács: | 100 dB |
| • sugárhajtású repülőgép felszállása 100m-ről: | 128 dB |

A hangmagasság a frekvenciával van összefüggésben, de ez is logaritmikus kapcsolat. Például ha egy hang frekvenciája kétszerese a másikénak, akkor ezt egy oktáv különbségként érzékeljük. Ha egy ennél is kétszer akkora frekvenciájú hangot veszünk, akkor ugyanekkora hangközként, oktávként halljuk ez előzőhöz képest. Az emberi hallás tartománya nagyjából 16... 20 000 Hz között van, de 2000 Hz körül a legérzékenyebb. Az ennél alacsonyabb hangokat infrahangnak, a magasabbakat ultrahangnak nevezzük. Ezeket az emberi fül nem érzékeli. A mi esetünkben az ultrahangoknak lesz nagy szerepük, ugyanis az AE jelek legnagyobb része ezen tartományba esik [3, 7, 11, 18].

A hangszín nehezen megfogható fogalom. A hang frekvenciaspektrumával van összefüggésben, de a pontos viszony egyelőre tudományosan nem tisztázott. Egy szinuszos hanghullámot "tisztá" hangként érzékelünk, míg bonyolultabb hullámformák esetében a frekvenciaspektrum domináns összetevője alapján érzékeljük a hang magasságát, míg a többi

komponens (felharmonikusok) a hangszín kialakításában vesz részt. Folytonos, "elmosódott" spektrum esetén nem érzékelünk egyértelmű hangmagasságot, hangszínt azonban igen. Jól elkülönülő frekvenciasávok esetén több különböző magasságú hangot halljunk egyidejűleg, azaz összhangzatot [7, 18].

2.2. Az akusztikus emissziós mérés

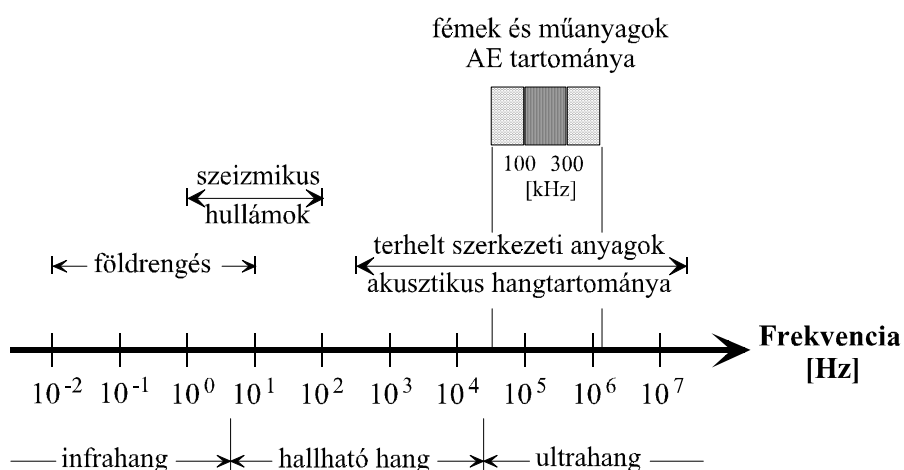
Ebben a fejezetben áttekintjük az akusztikus emissziós mérési módszert, annak múltját és jelenét, elvi hátterét és méréstechnikáját.

2.2.1. Az akusztikus emissziós mérés története

Az emberiség már rég felfedezte, hogy az anyag által kibocsájtott hangokból következtetni lehet annak belsejében lejárvó folyamatokra. Robert Hooke 1681-ben írta: *"Az állati, növényi vagy ásványi testek belső mozgásai és folyamatai felfedezhetők az általuk keltett hangokból"*. Ennek tudományos vizsgálatára azonban a XX. századig várni kellett. Az 1900-as évek elején kezdtek kísérleteket végezni, melyekkel kapcsolatot fedeztek fel egy terhelt szerkezet deformációja és a terhelés kiváltotta hangok, recsegések között. Az igazi áttörést az jelentette, mikor 1950-ben Joseph Kaiser olyan műszerrel vizsgálta az anyag által kibocsájtott hangokat, mely ultrahang tartományban is képes volt mérni. Fényt derített többek közt egy terheléskor végbemenő irreverzibilis folyamatra, melyet azóta Kaiser-effektusként ismerünk. Emellett lefektette az akusztikus emissziós mérés és anyagvizsgálat alapjait. A módszer a 60-as években kezdett elterjedni, mint roncsolás mentes anyagvizsgálati módszer, a 70-es években pedig már széles körben alkalmazták [1, 11].

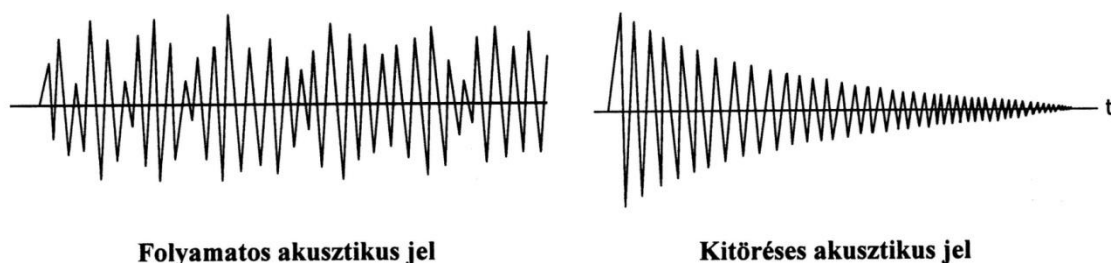
2.2.2. Az akusztikus emisszió elvi háttere

Az akusztikus emisszió (röviden: AE) egy szilárd testben keletkező és terjedő rugalmas hullám. Forrása az anyagban tárolt energia felszabadulása valamilyen fizikai vagy kémiai hatás következtében. AE sokféle anyagban és sokféle kiváltó okból létrejöhet, pl. földtani folyamatok során, nagynyomású folyadék- vagy gázrendszerek szivárgásánál, vagy terhelésnek szilárd testek tönkremenetelekor [1]. A különböző forrásból származó akusztikus emissziós rezgések jellemző frekvenciája más-más tartományokat ölel föl (2. ábra):



2. ÁBRA HANGOK CSOPORTOSÍTÁSA FREKVENCIÁJUK SZERINT

A továbbiakban mi a fémekben, illetve polimerekben keletkező AE-vel foglalkozunk. Ezek frekvenciája jellemzően 100 kHz és 1 MHz közé esik, amplitúdójuk pedig legtöbbször igen kicsi, így az emberi fül számára ritkán észlelhetők. A fémekben és polimerekben keletkező AE-kat kétféle csoportra oszthatjuk: folytonos és kitöréses (burst) típusú rezgésekre (3. ábra).



3. ÁBRA AKUSZTIKUS EMISSZIÓK TÍPUSAI

A kitöréses típusra jellemző egy rövid felfutó, majd ezt követően egy hosszabb lecsengő szakasz. Ez a típus egy tranziens jel, így a korábban leírtak szerint legtöbbször valamilyen viszonylag jelentős, nagy energiájú folyamathoz kapcsolódnak, és emiatt amplitúdójuk is sokkal, akár 10-14 nagyságrenddel nagyobb lehet a folytonos jelekhez képest. Ezen okokból kifolyólag akusztikus emissziós méréseknél elsősorban a kitöréses típusú jelekkel szoktak foglalkozni [1].

A fémekben az AE-s forrásokat négy csoportra oszthatjuk: diszlokációs mozgások, fázisátalakulások, sűrűlódások, repedés-kialakulás és -növekedés [1, 19]. Polimerek esetén általában szálerősítésű polimer kompozitokat vizsgálunk. Ezek olyan szerkezetek, ahol

nagymerevségű szálak (pl. üveg, bazalt, szén, PP) biztosítják a megfelelő szilárdságot, a befoglaló anyag, azaz mátrix (pl. epoxi, PP) pedig összetartja ezeket a szálakat, elősegítve azok együttdolgozását, és megakadályozva a nyomásra igénybevett szálak kihajlását. Kompozit például az üvegszál erősítésű polipropilén, de annak tekinthető a farostlemez is. Polimer kompozitoknál az AE jellemző forrásai: mátrix képlékeny alakváltozása, mátrix tépődése, száلكihúzóda, rétegelválás, szál szakadása [12, 17].

Ha a több forrásból egyszerre képződik akusztikus emissziós jel, akkor ezek összeadódnak, halmozódnak, nagy energiájú hanghullámokat képeznek. Ilyen történik például szakadáskor is.

Az akusztikus jeleket nem tudjuk közvetlenül érzékelni. AE mérésakor egy vagy több, az anyag felszínére helyezett mikrofonnal érzékeljük a rezgéseket. A hullámok, mivel forrásuk közelítőleg pontszerű, gömbhullámként kezd el terjedni, melynek sugara r . Ezek a forráshoz közel r^2 -tel, távolabb r^4 -nel, a felszínen pedig $r^{1/2}$ -nel csökkennek. A közegben való haladás során részben elnyelődnek a disszipáció hatására, a határfelületeken visszaverődnek, az inhomogenitások, anyaghibák és zárványok pedig torzítják azokat, mielőtt eljutnának a szenzorig [1]. Amennyiben az AE forrását meg akarjuk határozni, ezekkel mind valamilyen mértékben számolni kell. Ebben a dolgozatban ezzel nem foglalkozunk, így a hullámterjedés matematikai leírására ennél részletesebben nem térünk ki, a témáról bőséges anyagot találhatnak az érdeklődők a szakirodalomban: [1, 7].

2.2.3. Az akusztikus emisszió mérés technikája

Akusztikus emissziós érzékelők feladata az anyag rezgéseinek villamos jellé alakítása. Erre a feladatra többféle elven működő szenzorok is léteznek: elektromágneses, kapacitív, fotoakusztikus, piezorezisztív stb. elven működő érzékelők, de a leggyakrabban piezoelektromos érzékelőket használják a célra. A többi típus általában speciális körülmények között, pl. nagy hőmérsékleten használják, általános célra viszont a piezoelektromos műszereket egyszerűségük, alacsony áruk, kis zajuk és nagy érzékenységük ideálissá teszik.

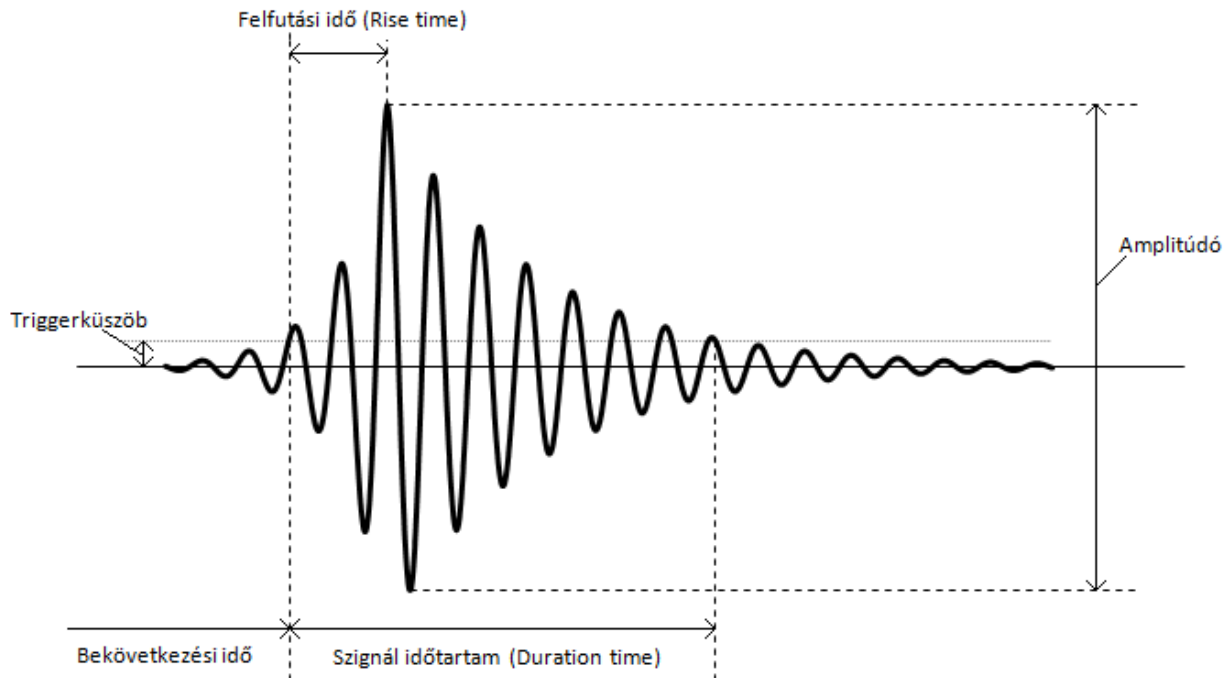
Az érzékelőt valamilyen módszerrel a testhez szorítják, hogy az közvetlenül átvegye a szilárd közeg rezgéseit. Az érzékelő és az anyagfelület közti hézagot úgynevezett akusztikus csatolóanyag (pl. olaj, víz, zsír, megfelelő ragasztó) tölti ki. Az érzékelő legtöbb esetben a felületre merőleges elmozdulásokat tudja érzékelni. Mind a longitudinális, mind a transzverzális,

mind a felületi (Rayleigh- vagy Lamb-féle) hullámok rendelkeznek ilyen komponenssel, így általában mindegyik érzékelése lehetséges AE érzékelővel.

Fontos szempont a szenzor kiválasztásában az érzékelni kívánt frekvenciasáv. Ez általában 30 kHz és 1 MHz közötti sávban található. Piezoelektromos szenzorok kialakításuktól (elsősorban mechanikai ágyazásuktól, csillapításuktól) függően lehetnek rezonáns, illetve szélessávú típusúak. A rezonáns típusú érzékelők sávszélessége kicsi, a szélessávú típusúakat több-rezonanciás kialakítással hozzák létre. [1, 4]

Másik fontos szempont az érzékenység, mivel kevésbé érzékeny eszköz jeléhez nagyobb erősítés szükséges, ami a zajokat is növeli. A megengedhető hőmérséklet szintén mérvadó lehet. Piezoelektromos anyagok elvesztik tulajdonságukat a Curie-pont feletti hőmérsékleten, így az ezen az elven működő szenzorok alkalmazhatóságát ez bekorlátozza. Általános célra használatos AE detektorok általában 120 °C-ig használhatóak. Léteznek speciális, magas hőmérsékletű szenzorok is, melyek kb. 550 °C-ig működtethetőek, de ezek érzékenysége jelentősen alacsonyabb [1]. A szenzor által létrehozott villamos jelet általában rövid, maximum 1 m-es koaxiális kábellel vezetik tovább egy előerősítőbe. A rövid kábel azért szükséges, mert a gyenge, erősítetlen jelre zajok szuperponálódhatnak, és ezek mennyisége a kábel hosszával nő. Az előerősítő ezután a zajokat is azonos mértékben felerősíti. Néha az előerősítőt magába a szenzorba beépítik, minimálisra csökkentve ezzel az ilyen típusú zajokat. Az erősített jel már kevésbé érzékeny a zajokra, mivel amplitúdója rendszerint jóval nagyobb azoknál, így jól elkülöníthető tőle. Ezután a jel a számítógépbe jut, ami feldolgozza, megjeleníti, illetve tárolja az adatokat. A gép nem magát a beérkező jelet tárolja, hanem azt elemi hangkitörésekre, ún. eseményekre osztja, és ezen események paramétereit regisztrálja. A piezoelektromos érzékelők tulajdonságaiból fakadóan egy elemi akusztikus emisszió impulzusra adott válasz egy kitöréses (burst) típusú jel lesz. Ennek jellegzetessége egy kezdeti rövid felfutó szakasz, majd egy hosszabb lecsengés. [1, 11]

A mérés során megadnak egy trigger küszöbértéket (threshold). A számítógép csak azokat az eseményeket regisztrálja, amelyek amplitúdója ezt az értéket átlépi. Ez egy elsődleges zajszűrési funkciót lát el. Szokásos megoldás az úgynevezett lebegő küszöbérték használata, mikor a pillanatnyi küszöbértéket az egyenirányított zajjel-átlagra (AVR) állítják be. Ha az esemény regisztrálásra kerül, akkor a következő értékeket tárolja le a gép: az esemény kezdetének időpontja; az esemény maximális amplitúdója; felfutási idő; az esemény teljes időtartama; rezgés-, vagy beütésszám (4. ábra).



4. ÁBRA EGY ESEMÉNY PARAMÉTEREI

A számítógép ezután ezekből más, származtatott mennyiségeket is ki tud számolni. Ilyen lehet például az esemény energiája (a jel négyzetének integrálja az esemény időtartama alatt), az úgynevezett signal strength (a jel burkológörbéjének integrálja) vagy a MARSE (az egyenirányított jel integrálja). Ezek számításának leegyszerűsítéséhez gyakran háromszöggel közelítik a jel burkológörbéjének alakját, ekkor a háromszög magassága a maximális amplitúdó, alapja pedig az esemény időtartama. Gyakran számolják még az időegységre jutó átlagos eseményszámot vagy oszcilláció számot, ez jól tükrözi az akusztikus aktivitás mértékét. Több detektor esetén megfelelő módszerrel kiszámítható az események helye is [1, 4, 11].

Az AE mérés technikájáról bővebb információk találhatóak [4, 13, 14, 15] számú szakirodalmakban.

3. Az akusztikus emissziós feldolgozó program

A programot a LabVIEW grafikus fejlesztői környezetben készítettük el, melyet a National Instrument nevű cég fejleszt. A LabVIEW programot két részre oszthatjuk: kezelő felületre, s blokk diagramra. Program indításakor a kezelő felület bukkan fel először, itt találhatjuk az esetleges beállításokat, s itt lesznek kijelezve az eredmények. A Blokk Diagramon láthatjuk a „programkódot”, s itt is programozhatjuk. Mivel ez a környezet grafikus nyelvet használ, ezért a programot tulajdonképpen „meg kell rajzolni”, azaz a már létező függvényeket és alprogramokat (ú.n. subVI-okat) ikonokként helyezzük el a blokk diagramon, majd vezetékekkel kötjük össze a különböző subVI-ok kimeneteit, s bemeneteit.

A „változók” ezeken a vezetékeken jutnak el ez egyik függvény kimenetétől a másik függvény bemenetére. (Ezen felül lehet lokális, illetve globális változókat is definiálni, melyekre vezetékek nélkül hivatkozhatunk.) A LabVIEW az adatfolyam programozást használja, azaz balról jobbra haladva hajtja végre az egyes program végeket, de csak akkor ha az összes „bekötött” bemenetére megérkezett az adat.

A program elkészítése során a LabVIEW Help menüje, a szakirodalom [5, 9, 10], illetve Dr. Lipovszki György volt a segítségünkre.

3.1. A program koncepciója

A program célja, hogy az akusztikus emissziós mérési adatokból hallható hangot hozzon létre. Célunk az volt, hogy az eredeti, nem hallható hangok karakterét minél jobban visszaadjuk, az ember számára jól érzékelhető módon. Szempont volt még, hogy a program sokféle beállítási lehetőséget hagyjon a felhasználó számára. Sok funkció és beállítás van még a felhasználó munkájának megkönnyítésére, ilyenek a felugró ablakok, az automatikus fájlkezelés, vagy a konvertált forrásfájlból való futtatás lehetősége.

Forrásunk olyan adatfájl, mely az AE események számszerű adatait tartalmazza. Ezeket az adatokat kell olyan hangokká alakítani, melyek az emberi érzékelés tartományába esnek. Ehhez az események három paraméterét kell megváltoztatni: az amplitúdót, a frekvenciát és az időtartamot.

A kész hangban minden eseményhez egy beállítható nagyságú időegység hosszúságú hang tartozik. Ezzel elérjük, hogy az eredeti hang rövidsége ellenére is halljuk azt. Mivel így a kész hangban minden eseményhez egyforma hosszúságú hang tartozik, így az amplitúdójuktól függ, melyik esemény mennyire lesz hallható a kész hangban.

Az amplitúdót az esemény maximum amplitúdójából, vagy az energiából lehet számolni. Az első lehetőség adja magát, míg az utóbbi mögött az a logikai megfontolás, hogy hiába nagy egy esemény amplitúdója, ha az nagyon rövid. Az energia viszont mind a maximális amplitúdót, mind az esemény időtartamát magába foglalja. Az amplitúdót normalizálással, illetve többféle skálázási móddal (arányos, négyzetgyökös, logaritmikus) transzformálhatjuk hallható tartományba.

A frekvenciát egy (egynél kisebb) konstanssal való szorzással csökkenthetjük le a hallható tartományba. Mivel az emberi hallás logaritmikus, ezért ez a transzformáció megtartja az eredeti hangközöket. Például, ha a szorzószámot felére csökkentjük, akkor a hang egy oktávval mélyebben fog szólni, de a "dallama" változatlan marad, megőrizve a hang eredeti jellegét.

A kész hanghullámot „.wav” formátumra konvertáljuk, így bármilyen hangmédia programmal könnyen meghallgatható. Ezen kívül ki is rajzolhatjuk a hanghullámot, amely segíti a hangok gyors (grafikus) összehasonlítását. A hangfájlt hangszerkesztő programmal tovább módosíthatjuk a lényeg kiemelésére, illetve a jobb hallhatóság érdekében, pl. zajszűrést hajthatunk végre, vagy zengetőt adhatunk a hanghoz.

3.2. A forrásfájl

Az akusztikus emissziós gép, illetve a hozzátartozó program az érzékelt jeleket egy „.txt” típusú fájlba exportálja ki. Az exportált adatokat, szóköz karakterek segítségével rendezi egymás alá. A következő adatok találhatóak meg így (oszlopos elrendezésben, balról jobbra ebben a sorrendben):

1. Bekövetkezési idő (mértékegysége: ms)
2. Amplitúdó (mértékegysége: dB)
3. Count (mértékegysége: db)
4. Risetime (mértékegysége: ms)
5. Duration Time (mértékegysége: ms)

A LabVIEW program ezen „.txt” fájl feldolgozását célozza meg.

3.3. A főprogram

A következőkben röviden összefoglaljuk a főprogram részeit, s működésének alapjait, hogy a későbbi részletezés alkalmával már lehessen egy átfogó kép, mi miért szükséges.

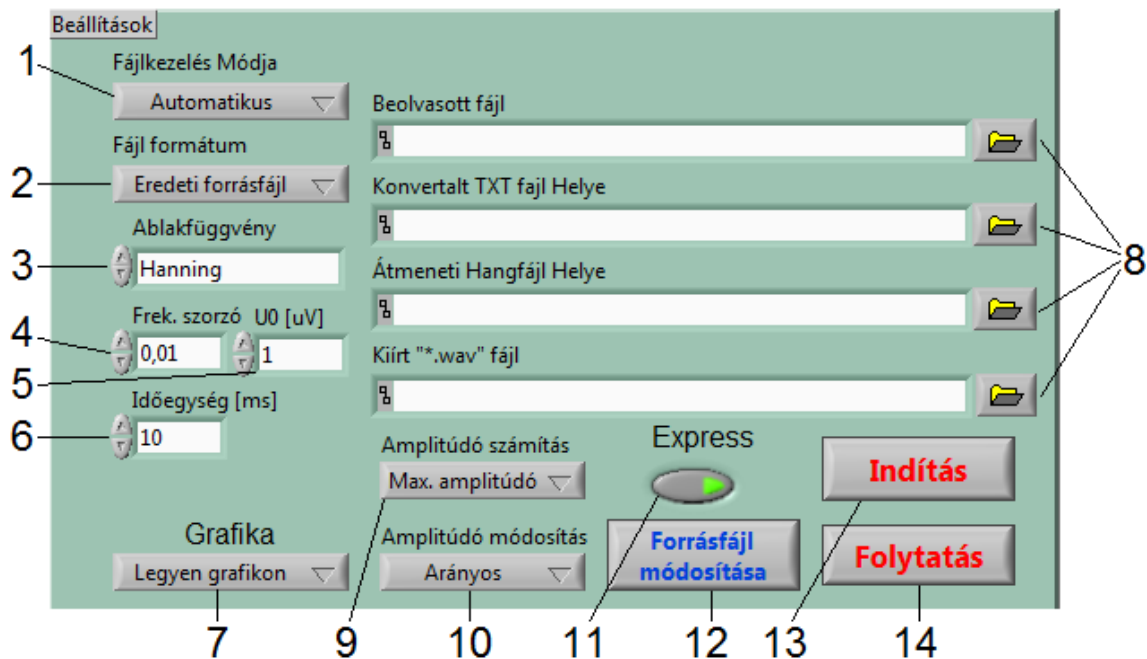
3.3.1. A főprogram kezelő felülete

Elindítás után a kezelő felület jelenik meg (melynek három részét külön-külön tárgyaljuk). Itt állíthatjuk be a program futásának egyes paramétereit.

A kezelő felület 3 részre osztható: a bal felső részen a beállításokat végezhetjük el, a jobb felső részben különböző információkat kapunk a feldolgozásról, illetve a mentett fájlokról, s végül az alsó részen egy diagram látható, melyen a hanghullám lesz látható (amennyiben ez van beállítva).

Beállítások (6. ábra)

A kezelő felület bal felső szegmensében, világoskék háttér előtt találjuk a program paramétereinek beállítására szolgáló felületet. A következőben ezt felületet kívánjuk bemutatni, illetve az egyes elemek funkcióját leírni. Itt található a beállításokra szolgáló rész:



6. ÁBRA BEÁLLÍTÁSOK

1: Fájlkezelés módja:

Manuális vagy Automatikus opciója létezik.

Manuális beállítás esetén a futás közben mentésre kerülő fájlok neveit, s útvonalait nekünk kell megadni, vagy a Beállításoknál látható úgynevezett „Path Control”-nál, vagy (ezeket üresen hagyva) a mentésnél megjelenő párbeszédpanelben.

Automatikus esetben a program az előre meghatározott neveken és helyekre (de mindig abba a mappába, ahol a program található) menti a fájlokat.

(Alapértelmezett: Automatikus)

2: Fájl formátum:

Itt is két lehetőségünk van. Amennyiben „Eredeti Forrásfájl” beállítást választjuk (alapértelmezett), akkor a beolvasott „.txt” fájlban nem végez konvertálást. Ha a másik opciót, a „Konvertált”-at választjuk, akkor a konvertálást kihagyja a program. Ha a konvertált forrásfájlt ismét konvertáljuk, akkor a végeredmény nem fog változni.

(Alapértelmezett: Eredeti Forrásfájl)

3: Ablakfüggvény:

Itt az elemi hullámok ablakozását állíthatjuk be. Számos különböző ablakfüggvény közül választhatunk: Rectangle, Hanning, Hamming, Blackman-Harris, Exact Blackman, Blackman, Flat Top, 4 Term B-Harris, 7 Term B-Harris, Low Sidelobe, Blackman Nuttall, Triangle, Bartell-Hanning, Bohman, Parzen, Welch, Kaiser, Dolph-Chebyshev, Gaussian.

Ezek tulajdonságairól a szakirodalomból tájékozódhatnak: [10, 20, 21]

(Alapértelmezett: Hanning függvény)

4: Frek. szorzó:

Ez a frekvenciaszorzó, mely a frekvencia módosításához kell.

(Alapértelmezett érték: 0,01)

5: U_0 [uV]:

A logaritmikus amplitúdó módosításhoz az alapérték. Dimenziója mikrovolt.

(Alapértelmezett érték: 1)

6: *Időegység [ms]:*

Az elemi hullámok hossza, továbbá a `Sampling Info` is ez alapján lesz beállítva. Milliszekundumban kell megadni.

(Alapértelmezett érték: 10 ms)

7: *Grafika:*

Beállítható, hogy a program kirajzolja-e a hanghullám grafikonját. Nagy minták esetén a kirajzoláskor memóriaproblémák léphetnek fel, ezért érdemes lehet kikapcsolni.

(Alapértelmezett: Legyen grafikon)

8: *Path Control rész:*

Ezekben a mezőkben adhatjuk meg a fájlok útvonalait (névvel együtt természetesen). Amennyiben ezek üresek, párbeszédablakok fognak felugrani ezek megadására.

(Alapértelmezett: üres)

9: *Amplitúdó számítás:*

Ezzel állíthatjuk be, hogy milyen módszerrel számítsuk ki az elemi hullámok amplitúdóit. "Max. amplitúdó", és "Energia" közül választhatunk.

(Alapértelmezett: Max. Amplitúdó)

10: *Amplitúdó módosítás:*

Itt választhatjuk ki, mely módszerrel változtassuk meg a hullámok amplitúdóit. "Arányos", "Négyzetgyökös" és "Logaritmikus" a módszerek közül választhatunk.

(Alapértelmezett: Arányos)

11: *Express:*

Itt kapcsolhatjuk a programot express módba, ekkor az alapértelmezett beállításokat használja, nem jelenít meg párbeszéd ablakot a beállítások módosításához, automatikus a fájlkezelés és „.wav” fájl is létrehoz.

(Alapértelmezetten be van kapcsolva, azaz Express módban fog futni a program.)

12: Forrásfájl módosítása nyomógomb:

Miután elindítottuk a LabVIEW programot, s már van egy forrásfájlunk, ezen gombra kattintva módosíthatjuk a feldolgozandó fájlt, ameddig le nem nyomjuk az Indítás gombot.

13: Indítás nyomógomb:

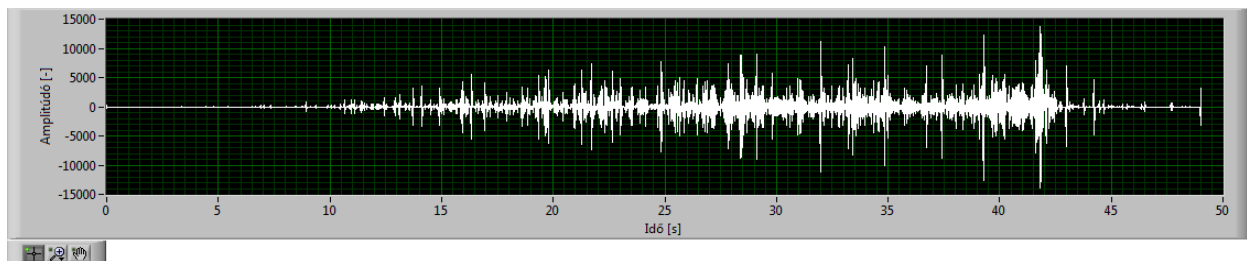
A program indításánál, miután már megadtunk egy forrásfájlt, ezen gomb lenyomására vár a program, hogy a feldolgozás elinduljon.

14: Folytatás nyomógomb:

Amennyiben nem express módban van a program, megjelenít egy ablakot a beolvasott adatok amplitúdó és frekvencia szélsőértékeivel. Ezen felugró ablaknál dönthetünk úgy hogy módosítjuk a beállításokat, ezen gomb lenyomásával folytathatjuk (módosítások után), a program futását.

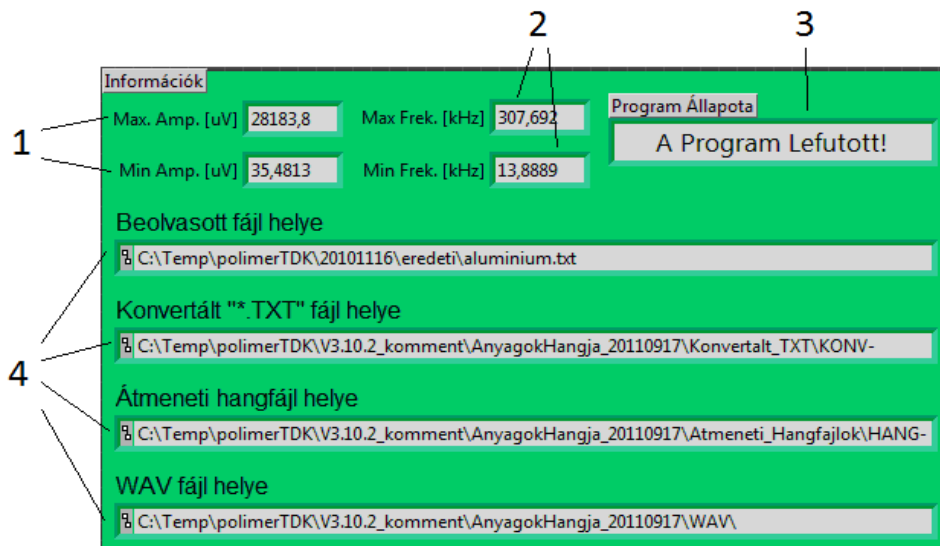
Grafikon (7. ábra)

A kezelő felület alsó felében találjuk a diagramot, itt láthatjuk a létrehozott hanghullámot (amennyiben nem tiltottuk le a megjelenítést). A függőleges tengelyen az Amplitúdó, vízszintesen az Idő látható. Az idő mértékegysége másodperc, az amplitúdóhoz azonban nem tudunk konkrét mértékegységet rendelni, így ez csak az arányokat mutatja meg, és különböző minták összehasonlítására se alkalmas. A bal alsó sarokban található palettákkal kinagyíthatjuk a grafikon egyes részleteit.

**7. ÁBRA GRAFIKON (FUTÁS UTÁN)**

Információk (8. ábra)

Itt olvashatjuk le a beolvasott adatok szélső értékeit, a minimális, illetve maximális amplitúdót (1) és frekvenciát (2). Továbbá a „Program Állapota” mezőben (3) olvashatjuk, hogy a program melyik állapotban van, továbbá a forrásfájl, s a mentett fájlok útvonala is látható (4). Ezek az első futtatás előtt üresek.



8. ÁBRA INFORMÁCIÓK (FUTÁS UTÁN)

Program állapotok:

- Indítható
- Könyvtár létrehozása
- Beolvasás
- Számolás
- Elemi Hullám Számolás
- Transzformálás
- Elemi Hullám Generálás
- Összefűzés
- Hangfájl Kiírás
- KészWav Létrehozására
- (Wav Fájl Készítés) – ez opcionális
- A Program Lefutott!

3.3.2. A főprogram blokk diagramja

A blokk diagramon láthatjuk a program "kódját". A különböző subVI-okat vezetékekkel kötjük össze, hogy a számított értékeket egymásnak átadhassák. A különböző színek különböző típusú adatokat jelölnek: kék - integer (egész) típusú szám; narancs - double (lebegő pont) típusú szám; rózsaszín - string (szöveg) típusú adat; zöld - boole (logikai) típusú adat; sötétzöld - path (elérési útvonal); barna fehér csíkos - itt hanghullámok. A vékonyabb vonalak egy adatot „szállítanak”, míg a vastagabbak tömböket.

A programot egy „Flat Sequence” nevű struktúrában írtuk meg. Ezen struktúra tulajdonsága, hogy csak az egyes „film kockákban” lévő összes subVI („függvény”, „alprogram”) lefutása után lép tovább a következő (tőle balra lévő) kockába. Ezzel blokkosítottuk a programot, mely könnyítette a programozást, egyszerűsítette a hibakeresést.

A kis sárga színű dobozokban megjegyzéseket olvashatunk.

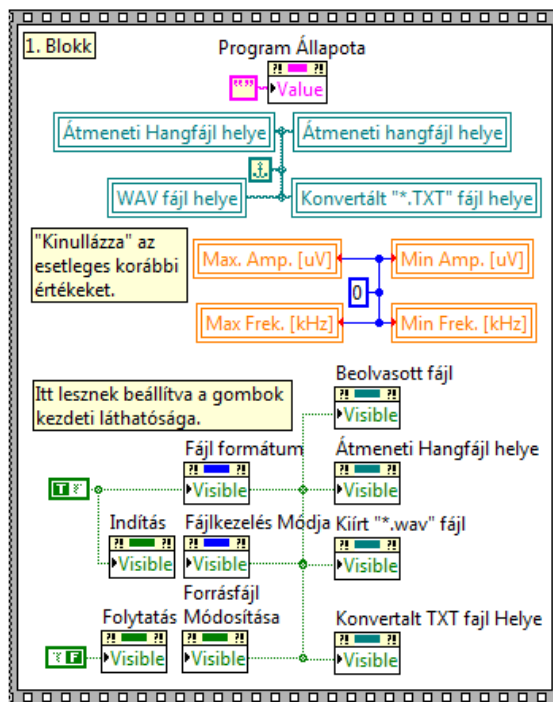
Az egyes blokkok (avagy kockák, a struktúra filmszalagra való hasonlatossága miatt) felső részében rózsaszínű vezetékekkel van összekötve egy „Program Állapota” doboz, s egy minden kockában más szöveggel rendelkező szövegdoboz. Ez felel a kezelő felület „Program állapota” mezőt frissíti, mikor a blokk aktív lesz.

A programban a fontosabb részegységeket subVI-okba rendeztük. Helyenként ezek nagyon egyszerű műveleteket takarnak, illetve olyan bemeneteket vagy kimeneteket tartalmaznak, amelyeket nem használunk. Ennek oka a főprogram átláthatóbbá tétele, és a későbbi módosítások, továbbfejlesztések megkönnyítése.

A következőkben röviden összefoglaljuk az egyes blokkok főbb tulajdonságait, a részek pontosabb leírását később fejtjük ki. Az egyes blokkok láncszerűen, sorban követik egymást. Mikor tehát az egyik blokk teljesen lefutott, akkor kezdi meg a működését a következő.

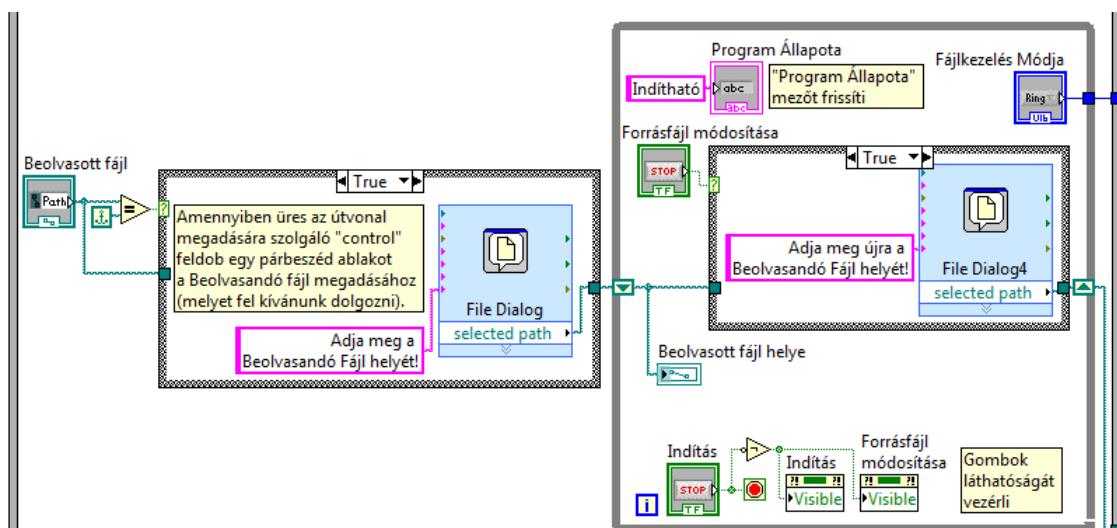
1.blokk (9. ábra):

Az első kockában kitöröljük a kezelőfelületen található mezőket, hogy ne az esetlegesen korábban bennmaradt értékeket olvassuk az új futásnál. (Útvonalak, szélső érték információk.) Továbbá beállíthatja a kezdeti láthatóságot. (A „Folytatás” gombon kívül minden más kezelő szerv látható lesz.)



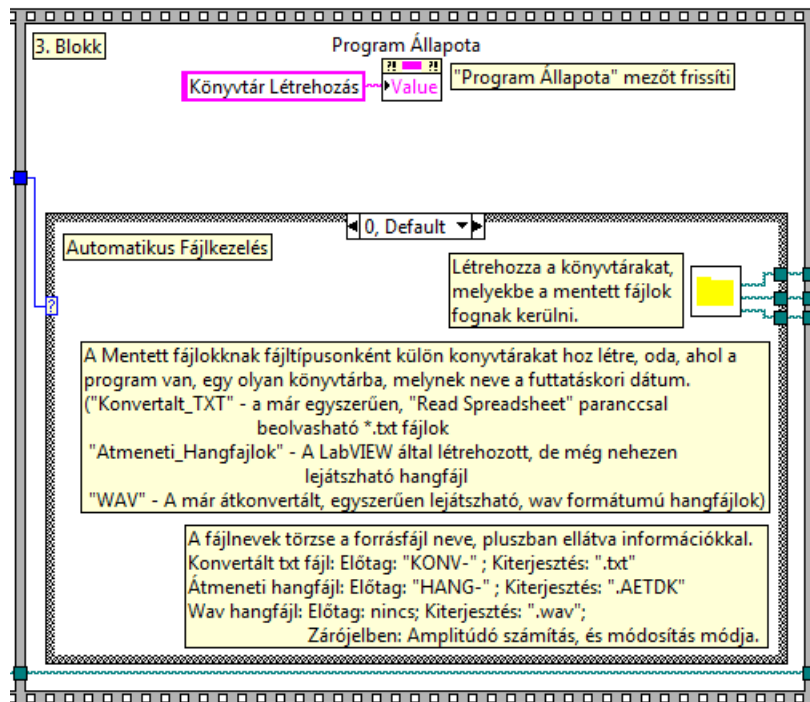
2.blokk (10. ábra):

A második kockában (amennyiben üres a „Beolvasott fájl” nevű path control, megjelenik egy párbeszédablak, hogy adjuk meg a feldolgozni (beolvasni) kívánt „txt” fájlt. Miután ezt megtettük még van lehetőségünk a forrásfájl módosítására, a kezelő felület megfelelő gombjának megnyomásával. („Folytatás” nyomógomb.) Innen a „Mehet” gombra kattintva folytatódik a program.

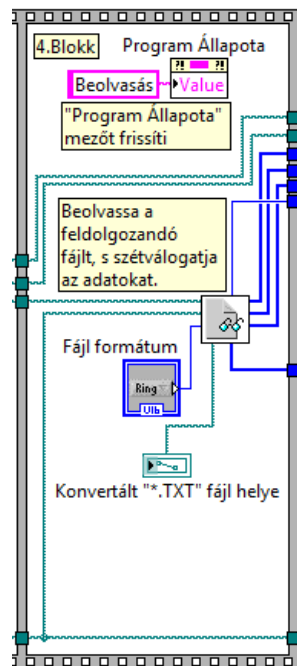


3.blokk (11. ábra):

A harmadik kocka valósítja meg manuális esetben a további „path control”-ok beolvasását, illetve automatikus esetben a megfelelő könyvtárak létrehozását. Az útvonalakat adja eredményül.

**11. ÁBRA 3. BLOKK****4.blokk (12. ábra):**

A negyedik kockában található a „Beolvasott fájl”-ként megadott forrásfájl beolvasása, s feldolgozása. Amennyiben a „Fájlformátum” „Eredeti forrásfájl” beállításon van, át is konvertálja a „.txt” fájlt, s elmenti azt. Másik opció esetén nem konvertál, hanem csak beolvassa az adatokat, amelyeket külön tömbökben ad eredményül. Az adatokon végez egy előszűrést. Ezt a **Beolvasas.vi** nevű alprogram végzi el, s az adatokat egész típusú tömbökben tárol. Az átkonvertált „.txt” fájl helyét feltünteti a kezelő felületen.



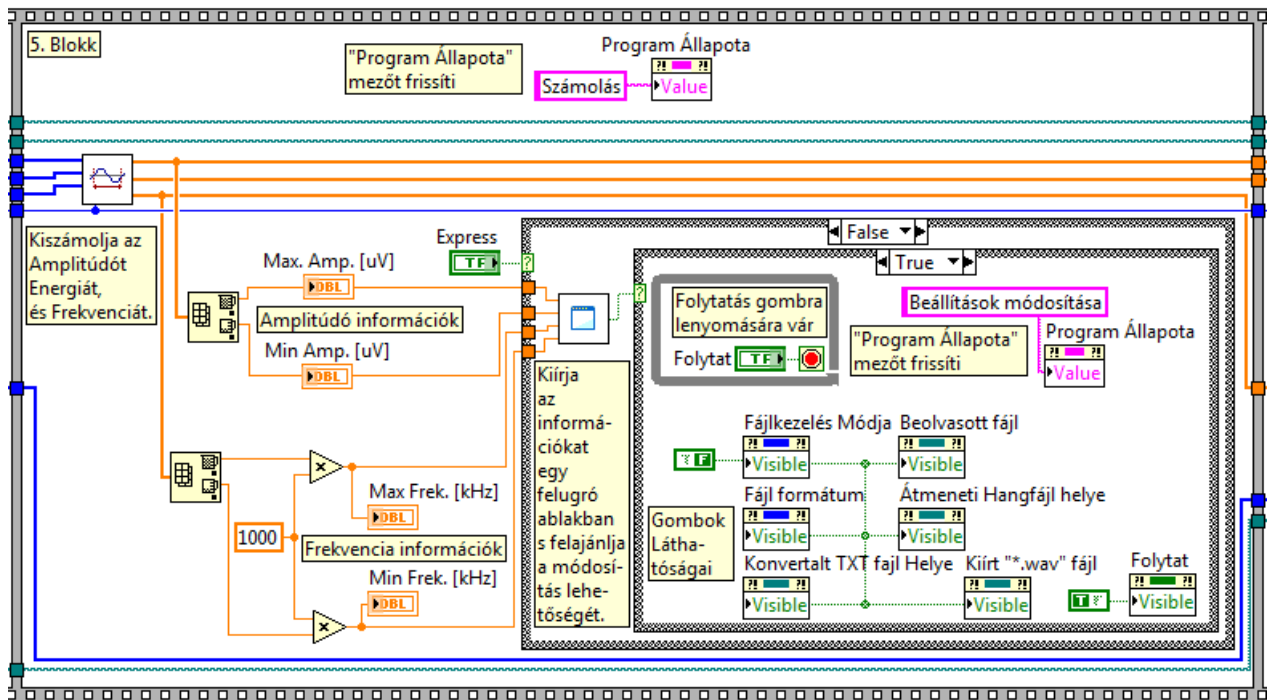
12. ÁBRA 4. BLOKK

5.blokk (13. ábra):

Ebben a fázisban határozza meg a program az Amplitúdó, a Frekvencia, s Energia változók értékeit. amelyekkel az elemi hullámokat határozzuk meg. (Azonban az amplitúdó még dB érték, ezért ezt még módosítani kell). Ezeket a műveleteket végzi el a **Szamol.vi**, s az értékeket lebegőpontos tömbökben tárolja. Mivel bizonyos beállításokat befolyásolhatnak a kiszámolt adatok szélső értékei, a program (nem Express módban) egy ablakot jelenít meg a Maximális és Minimális Amplitúdó, illetve Maximális és Minimális Frekvencia értékekkel, s felajánlja a módosítás lehetőségét.

Amennyiben a módosítást választjuk, a változtatások után a „Folytatás” gombra kattintva folytathatjuk a programot.

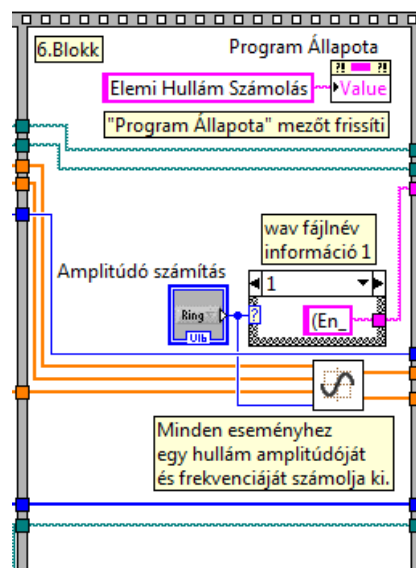
Különböző vezérlők láthatóságát is vezérli a program (módosítás esetén), mivel nem mindegyik lenne hatással a továbbiakra. Így ekkor eltűnnek az „Útvonal vezérlők”, a „Fájlformátum”, „Fájlkezelés módja” ikonok a kezelő felületről, s láthatóvá válik a „Folytatás” gomb.



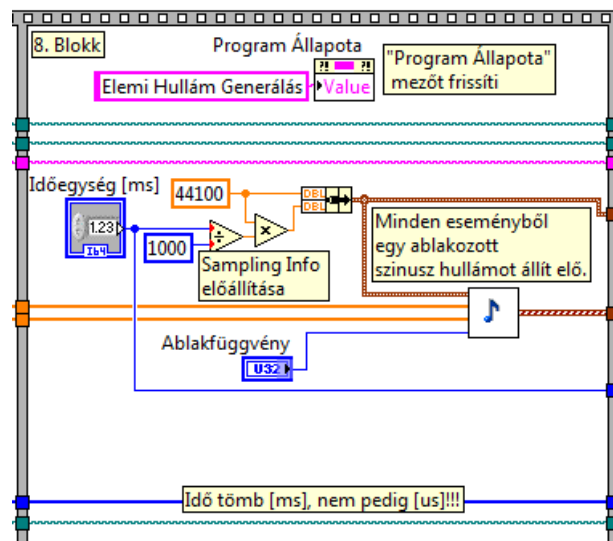
13. ÁBRA 5. BLOKK

6.blokk (14. ábra):

Ebben a blokkban az elemi hullámok (szinuszos hullámok) végleges frekvenciáit, s amplitúdóit. Az amplitúdó kiszámolásának módját határozzuk meg az „Amplitúdó számítás” kiválasztó elemmel. Továbbá ez utóbbi választásról az információ, amely a „wav” fájl nevébe fog kerülni itt lesz kiválasztva. Az „Elemi hullam szamolas.vi” fogja végrehajtani, s az eredményt lebegő pontos tömbökben tárolja.



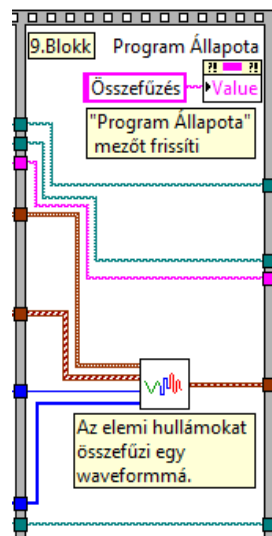
14. ábra 6. blokk



16. ÁBRA 8. BLOKK

9.blokk (17. ábra):

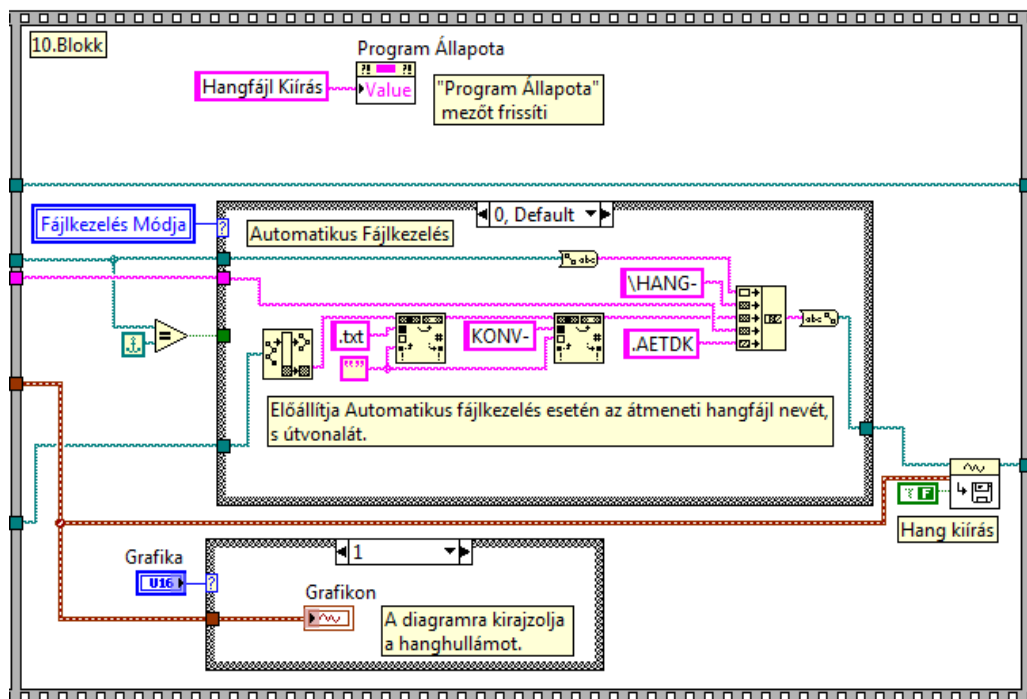
Ezen program blokkban fűzzük össze az elemi hullámokat egy végső hangfájllá. Azonban ez a hangfájl még nem alkalmas lejátszásra. (Ez lesz az Átmeneti Hangfájl, melynek kiterjesztése „AETDK”) Egy adatként továbbítja a hangfájlt.



17. ÁBRA 9. BLOKK

10.blokk (18. ábra):

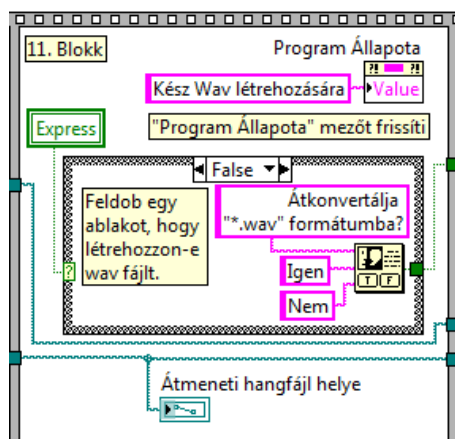
A tizedik program blokkban a „Grafika” kiválasztó elem beállításától függően jelenítjük meg a hangfájlt, a kezelőfelület grafikonján. Amennyiben nem adtunk meg a „Path Control”-nál az átmeneti hangfájlnak nevet, s útvonalat, illetve nem is automatikus fájlkezelésben futtatunk, feldob egy párbeszéd ablakot ennek pótlására.



18. ÁBRA 10. BLOKK

11. blokk (19. ábra):

Amennyiben nem Express módban futtatjuk a programot, ennél a blokknál megjelenik egy ablak, ahol eldönthetjük, átkonvertáljuk-e a hangfájlt „wav” formátumba. Express beállításnál a fájl mindenképpen átkonvertálásra kerül.

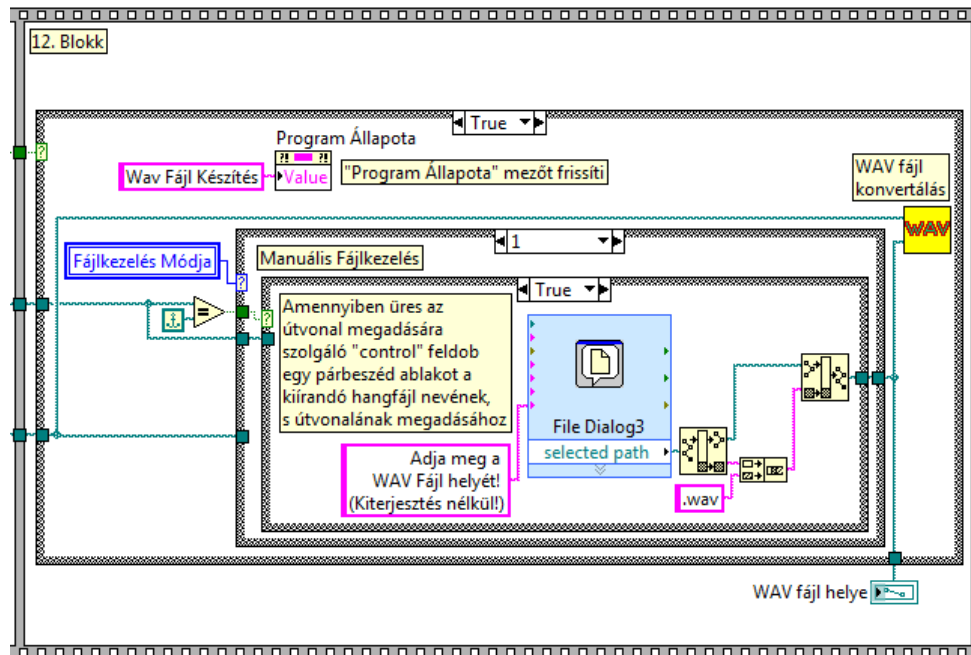


19. ÁBRA 11. BLOKK

12. blokk (20. ábra):

Itt kerül sor az átkonvertálásra, amennyiben ez lett megadva (az előző kocka párbeszéd ablakában, vagy Express módban).

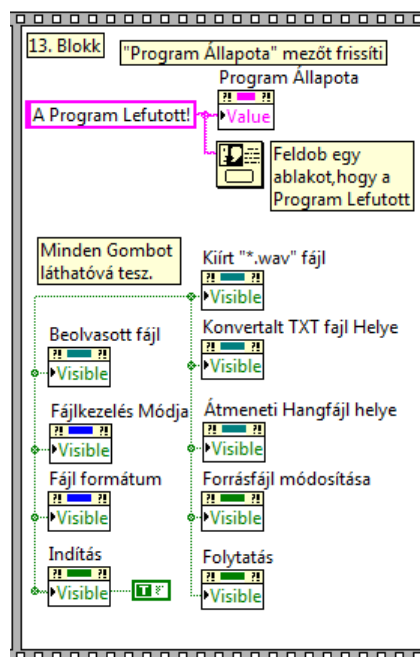
A HangfajlKonvertalas.vi önmagában is használható már meglévő waveform-fájlok „.wav” fájlá konvertálására.



20. ÁBRA 12. BLOKK

13.blokk (21. ábra):

Ebben a program blokkban egy ablak jelenik meg, hogy közölje a program lefutott, továbbá az összes vezérlő láthatóvá válik.



21. ÁBRA 13. BLOKK

3.4. A program működése

A LabVIEW programokat a felső menüben a nyíl szimbólummal tudjuk elindítani (addig tervező fázisban van a program, azaz ekkor szerkeszthetjük, „programozhatjuk”). Ekkor fog futni a program, kezdve az első blokk végrehajtásával. A továbbiakban részletezzük az egyes blokkok működését.

1. blokk:

Itt különböző, úgynevezett lokális változókat találunk, melyek a kezelőfelületen található megjelenítőkhöz (Indikátorokhoz) tartoznak, illetve a gombokhoz tartozó, úgynevezett „property node”-ok, amelyeket is az adott gomb megjelenítésének beállítására használunk. A lokális változókkal tudunk az egyes indikátoroknak a program futása közben újabb és újabb értéket adni. Ebben az első blokkban minden indikátor kiürítünk, azaz a szöveges (string) megjelenítőnek üres stringet, az útvonal megjelenítőnek (path indicator-nak) üres útvonalat, s a szám kijelzőknek nulla értéket adunk. A gombokhoz tartozó „property node”-okon most a láthatóságot tudjuk beállítani, mégpedig egy Boole, azaz logikai változóval. Indításkor az összes gomb, illetve vezérlő látható, kivéve a „Folytatás” gombot.

2. blokk:

Beolvassa a „Beolvasott fájl” nevű vezérlő értékét, amit a tervező módban tölthettünk volna ki. A beolvasott értéket (a kitöltés hiánya esetén üres útvonalat, azaz path-t) a program megvizsgálja, hogy üres-e. A vizsgálat eredménye egy logikai változó, amely az úgynevezett case structure, azaz eset struktúra kiválasztó (selector) részébe lesz betöltve. A case structure-ben több különböző alprogramot helyezhetünk el, amelyek közül a selector segítségével választhatjuk ki, éppen melyik fusson le. A selector-t logikai változóval (két lehetőség), vagy számmal (több lehetőség) is vezérelhetjük. A struktúra keretében felül középen tekinthetjük meg, éppen melyik eset programját látjuk. Amennyiben nem volt üres az útvonal vezérlő (path control), az onnan beolvasott értéket egyszerűen tovább adja a kimenetén (grafikusan a bemenet, s kimenet egyszerűen egy vezetékekkel össze van kötve). Ha üres volt, akkor a program feldob egy párbeszéd panelt, melyben megadhatjuk a fájl útvonalát. Ezt a módszert a mentési útvonalaknál is felhasználtuk, az elv ugyanaz.

Ezután az útvonal egy úgynevezett while ciklus bemenetére (egészen pontosan egy `shift register`-be) kerül. Ez a ciklus mindaddig fut, ameddig a kezelő felületen meg nem nyomjuk az „Indítás” gombot. A gomb hamis (`false`) logikai értéket ad egészen a lenyomásig, s ez az igaz (`true`) érték fogja leállítani a ciklust. Az újra, s újra futás esetén a program a bemenő értékeket mindig a ciklus széléről újra és újra beolvassa. `Shift register` esetén azonban az érték változhat, ugyanis, amit a `shift register`-be betöltünk a jobb oldalon, azt fogjuk a következő ciklusnál a baloldalon beolvasni. (Kivéve a legelső ciklusnál, mikor is a cikluson kívülről, baloldalra bekötött értéket fogja beolvasni.) Az útvonalat így a futtatás lenyomása előtt még megváltoztathatjuk. A mindenkor útvonal betöltődik egy útvonal kijelzőbe (`path indicator`-ba), így mindig az aktuálisan érvényben lévő útvonalat látjuk kijelezni a kezelő felületen. Az útvonalat átvezetjük egy újabb `case structure`-ön, melyet a „Forrásfájl módosítás” gombbal vezérelhetünk (ami szintén hamis értéket ad egészen a lenyomásáig). Mikor ezt a gombot lenyomjuk, akkor az egyszerű átvezetés helyett felugrik egy párbeszédablak, ahol ismét megadhatjuk az útvonalat. Továbbá minden ciklusban betöltődik az „Indítható” felirat a „Program Állapota” elnevezésű szöveges megjelenítőbe (`string indicator`-ba) ami az azonos nevű mezőt frissíti a kezelő felületen. (Itt az eredeti indikátort láthatjuk, a többi ellenben csak ennek a lokális változója, mint az első blokkban is). Az alsó felén láthatjuk a ciklus megszakítására szolgáló rész mellett az `Indítás`, s a `Forrásfájl módosítása` nevű gombok láthatóságáért felelős „`property node`”-okat. Ezeknek az „Indítás” gomb ad értéket, ami lenyomásig minden ciklusban hamis értéket ad, de ez az érték egy logikai negálás után érkezik meg a `node`-okba. Így lenyomáskor fognak hamis értéket kapni, ami miatt ez a két gomb a program futásának végéig nem lesz látható.

Itt kerül beolvasásra a „Fájlkezelés Módja” is, melyet legördülő menüből választhatjuk ki. A két választási lehetőség az „Automatikus”, s a „Manuális”. (Ezekről részletesebben később).

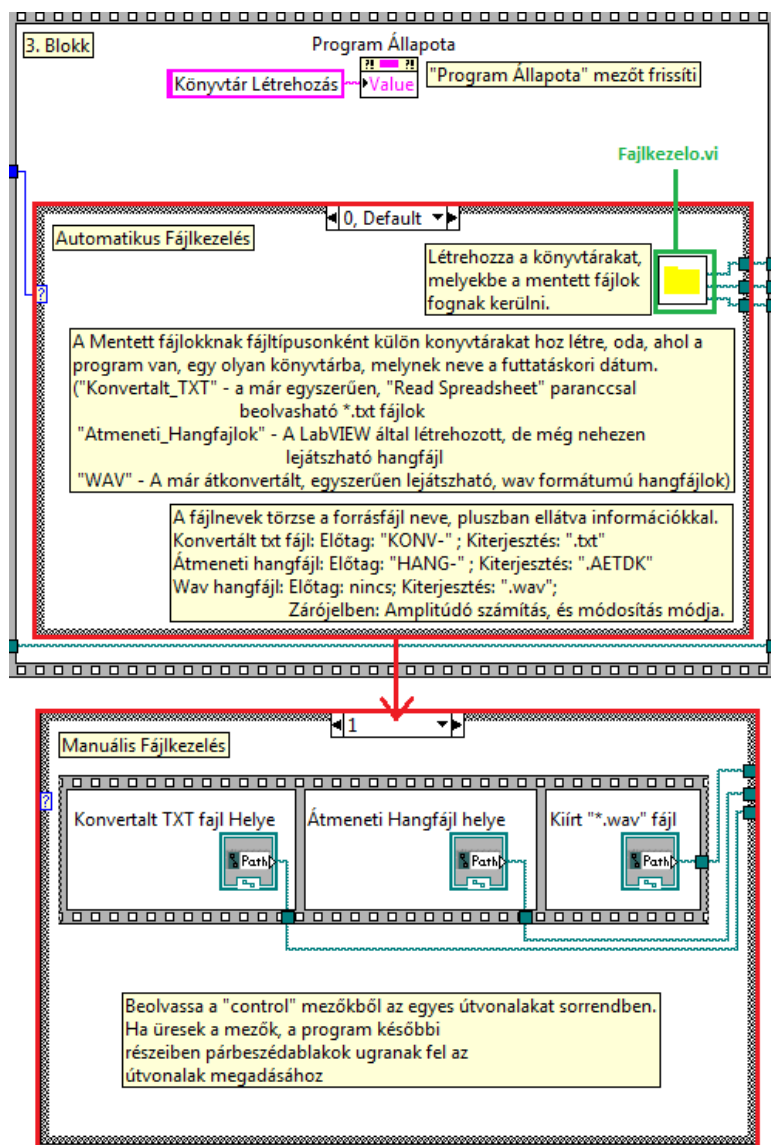
Mivel eddig csak a forrásfájl helyét olvastuk be (a fájlkezelés módját mindig újra beolvassuk a ciklusokban), ezért minden kezelő felületen látható vezérlő értékét a kívántra beállíthatjuk. Az „Indítás” gomb lenyomásával megszakítjuk a ciklust, s a program a következő blokkba lép.

3. blokk:

Ez a blokk felel a fájlkezelésért. A blokk felső részében frissül a „Program Állapota” mező („Könyvtár létrehozás” jelenik meg).

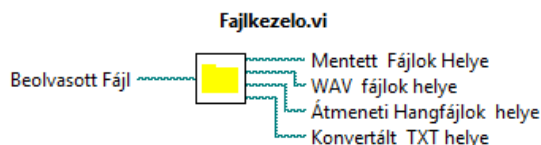
Egy case struktúrát láthatunk alatta, amit a „Fájlkezelés” beállítás vezérel. „Manuális” fájlkezelés során sorrendben beolvassa a többi három útvonal vezérlőbe beírtakat. (Amennyiben üresek, a program később párbeszédablakot dob fel a megadására).

„Automatikus” fájlkezelés során a program meghívja a *Fajlkezelő.vi* nevű subVI-t, amely a mentendő fájlokhoz hoz létre könyvtárakat. A case struktúra kimenetén a három útvonal jelenik meg, s ezeket vezetjük tovább (22. ábra).



22. ÁBRA 3. BLOKK

Fajlkezelő.vi (23.ábra):



23. ÁBRA FAJLKEZELŐ.VI

Ez a subVI létrehoz egy könyvtárat azon a helyen, ahol a program is található, „AnyagokHangja_20110927” néven, amennyiben még nincs ilyen nevű mappa. A számok mindig a futtatáskori dátumot tartalmazzák. Utána sorrendben létrehoz ebbe az új mappába (amennyiben még nincsen) egy KONVERTALT_TXT, ATMENETI_HANGFAJLOK, s egy WAV nevű mappát, amelyekbe értelemszerűen fognak kerülni a mentendő fájlok. Továbbá létrehozza a konvertált „.txt” fájlok teljes elérési útvonalát (azaz, fájlnevvvel együtt). A „.txt” fájl neve a következő lesz: „KONV-*.txt”, ahol a csillaghelyére kerül az eredeti forrásfájl neve (kiterjesztés nélkül).

A másik két mentendő fájl esetében azért nem itt lesz létrehozva a teljes útvonal, s csak a mappa elérési útja, mert olyan plusz információk is belekerülnek azokba a fájlnevekbe, amik ezután még változhatnak.

A VI, mint a főprogram, szintén szekvenciális, elsősorban a jobb átláthatóság kedvéért. Minden mappa, illetve a hozzájuk tartozó útvonal létrehozás külön-külön kockában foglalnak helyet. Az első kockában (24-1. ábra) jobb oldalt látható, úgynevezett „Get Date/Time String” nevű subVI kiolvassa az aktuális dátumot, amiből a pontokat kiveszi a következő háromszor lefutó for ciklus. Shift regiszter, valamint a „Search and Replace String” nevű függvény hajtja végre ezt a műveletet.

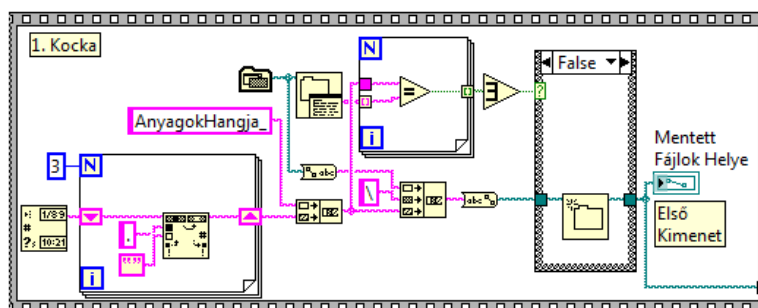
A következő lépésben az „AnyagokHangja_” tartalmú string-hez hozzáfűzi ezt a dátum információt. Ezek felett láthatjuk az „Application Directory” elnevezésű subVI-t, amely a program mappájának elérési útvonalát adja eredményül path-ként, azaz útvonalként. A már korábban létrehozott mappanév elé fűzi ezt az útvonalat, így ide lesz létrehozva ez a könyvtár (amennyiben még nincs). Az útvonal segítségével és a „List Folder” nevű alprogrammal kiolvassuk, ezen helyen milyen könyvtárak találhatók. A mögötte elhelyezkedő for ciklus megnézi, létezik-e már olyan nevű

könyvtár, melyet létre kívánunk hozni, s amennyiben nincs egyezés, a case struktúra létrehozza ezen a néven.

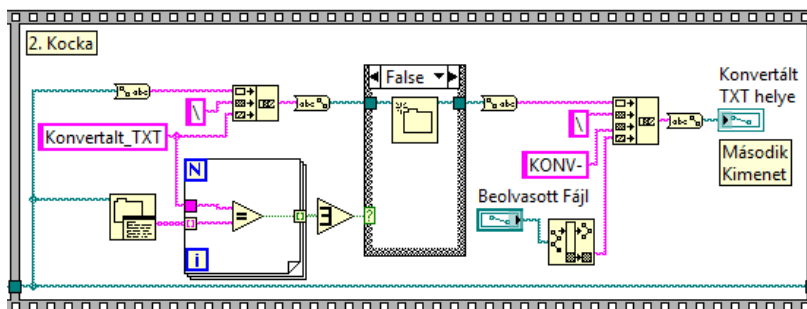
A könyvtár útvonalát kimentként kivezetjük, valamint a következő kockába is kivezetékezzük, mivel a további könyvtárakat ebbe a könyvtárba fogjuk létrehozni. Az útvonal típusú változókat át kell alakítani szöveggé, hogy szöveggént operálhassunk velük, majd vissza kell alakítani őket, hogy a program elérési útvonalakként értelmezni tudja őket. Erre a célra szolgál a „Path to String”, valamint a „String to Path” nevű beépített subVI. A többi kockában is hasonlóképpen lesz ellenőrizve, hogy létezik-e már olyan nevű mappa, s ha nincs, akkor létrehozza.

A második kockában (24-2.ábra) az eltérés, hogy a könyvtár létrehozása után, még hozzáfűzi a létrehozandó átkonvertált „.txt” fájl nevét is, mégpedig úgy, hogy a Beolvasott fájl útvonalából a fájlnevet lecsatoljuk („Strip Path” nevű algoritmussal), ezen név elé fűzzük a „KONV-” kiegészítést. (Mivel a beolvasott fájl is „.txt”, így a kiterjesztést nem kell változtatni.)

Kimenetként a mappák útvonalait adja, melyek a mentésnél lesznek felhasználva.



24-1. ÁBRA FAJLKEZELO.VI BLOKK DIAGRAMJA (ELSŐ KOCKA)



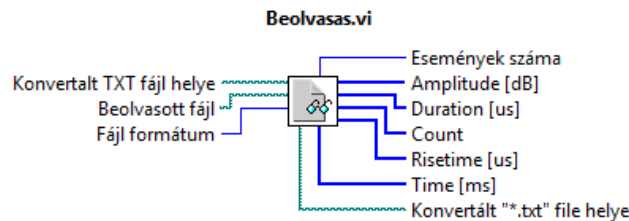
24-2. ÁBRA FAJLKEZELO.VI BLOKK DIAGRAMJA (MÁSODIK KOCKA)

4.blokk:

A felül látható, „Beolvasás” szöveggel ismét frissül a „Program Állapota” mező. Az előző blokk útvonalaiból csak a konvertált „.txt” fájl elérési útvonala lesz felhasználva, mely a „Beolvasas.vi” bemenetére kerül. A másik két útvonal csak át lesz vezetve ezen a blokkon.

Beolvasásra kerül a „Fájlformátum” beállítás (melynek értéke kiindulásként nulla, azaz „Eredeti forrásfájl” beállítás, 1-es ha már korábban konvertált „.txt” fájl olvasatunk be), s a feldolgozandó, valamint a konvertált „.txt” fájl útvonalával együtt a már említett „Beolvasas.vi” bemenetére kerülnek.

Beolvasas.vi (25. ábra):



25. ÁBRA BEOLVASAS.VI

Ez a subVI valósítja meg a kiinduló „.txt” fájl feldolgozását, amelynek eredményeként az adatok tömbökben fognak továbbhaladni.

A bemenetként bejövő „Fájlformátum” értéke vezérli a case struktúrát. Ha a „.txt” fájl eredeti (még nem konvertált), akkor egy „Konvertalo.vi” nevű alprogram fogja konvertálni a „.txt” fájl, melynek két bemenete a forrásfájl, s a konvertált „.txt” fájl útvonala, kimenete a konvertált „.txt” fájl útvonala. (Erről részletesebben később olvashatunk.)

Ha a beolvasandó fájl már konvertált, úgy a case struktúra a megadott „Beolvasott fájl” útvonalát adja eredményül. Ezt az útvonalat a kijelzés miatt kimenetként is kivezetjük. Egy „Read From Spreadsheet” nevű subVI segítségével táblázatként olvassuk be a „.txt” fájlt (ezen művelet végrehajtásához kell a konvertálás). Így kapunk egy mátrixot, ahol minden sor egy eseményt, s minden oszlop más-más adatot tartalmaz (sorrendben: Time, Amplitude, Count, Risetime, Duration). Ebből a mátrixból kiolvassuk az egyes oszlopokat az „Index Array” nevű alprogrammal, amelybe bevezetjük a mátrixot, s hogy hányadik

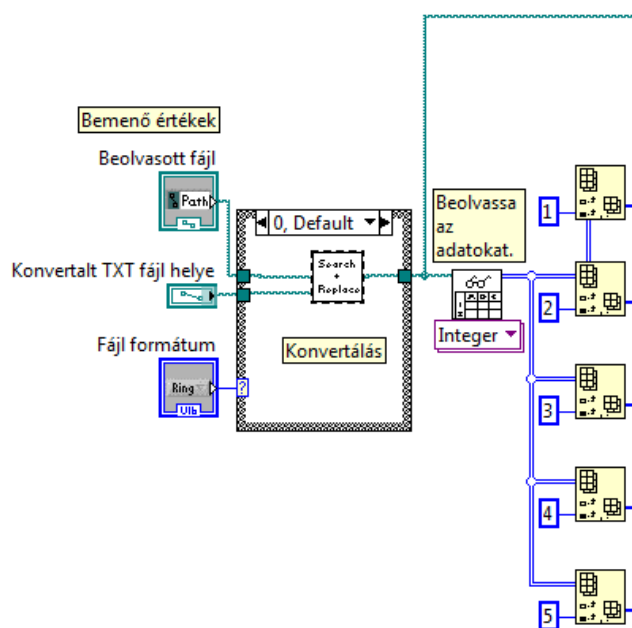
oszlopot kívánjuk kiolvasni (26-1. ábra). Ezek után egy while ciklus következik, mely kiszűri a „0” idejű sorokat. Ilyenek úgy jöhetnek létre, ha a „.txt” fájlban üres sorok is voltak, tehát ezzel információt nem veszünk, de a kimenő tömbök mérete csökkenhet, amivel memóriát, s gépidőt nyerhetünk. (Az egyes tömböket a ciklus shift registereibe vezetjük be.) A szűrésnél a „Search 1D Array” subVI-t használjuk, melynek bemenete az átvizsgálandó tömb (Time), s a keresett érték Eredményül megadja az első találat sorszámát. Ezt a sorszámot bevezetjük az egyes oszlopokkal együtt egy „Delete From Array” nevű alprogramba, mely a megadott tömbből kitörli a megadott sort. A tömb belekerül a shift registerbe, s a ciklus újratekődik. A „Search 1D Array” negatív értéket ad eredményül, ha a tömbben nem talált egyezést a keresett értékkel. (Mivel a sorszám csak természetes szám lehet.) Így a ciklus leállítása akkor következik be, ha ez az érték kisebb lesz nullánál.

Ezután az új tömböt bevezetjük egyszer egy „Array Size” nevű VI-ba, mely a tömb méretét adja eredményül (ami az események számával egyezik meg), s ezt is kivezetjük.

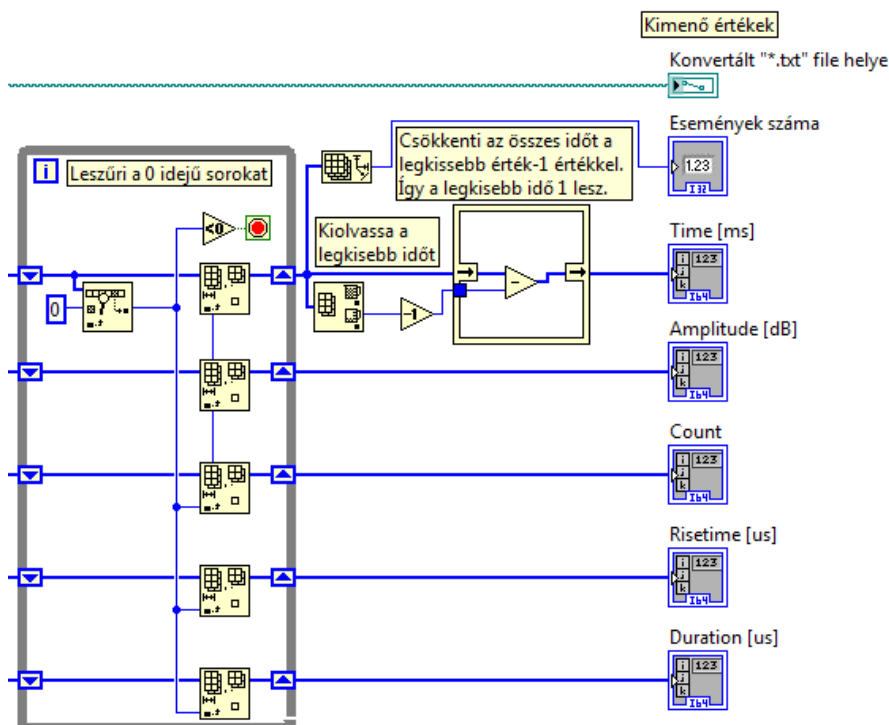
Másrészt a Time tömböt (mely az események bekövetkezésének időit tartalmazza) átformáljuk, hogy az első idő 1 legyen. (A későbbiek folyamán ezzel gépidőt s memóriát spórolunk a vizsgálatoknál, ciklusoknál amik az idő által vannak vezérelve. Információt nem veszítünk, hiszen csak az összes időt eltoljuk ugyanannyival, azaz, mintha az adatok rögzítését 1 milliszekundummal az első esemény bekövetkezése előtt kapcsoltuk volna be.)

Ehhez az „Array Max & Min” subVI segítségével kiolvassuk a legkisebb értéket a Time tömbből, ezt az értéket eggyel csökkentjük, s bevezetjük a tömbbel együtt egy „In Place Element Structure” nevű struktúrába. Ez a struktúra egy for ciklushoz hasonlóan működik, a tömb összes elemére elvégzi ugyanazt a műveletet, de kevesebb memória felhasználásával, s valamivel rövidebb idő alatt.

Ezután a Time tömböt is kivezetjük a többi tömbhöz hasonlóan. Az egyes tömböknek angol neveket adtunk, ezzel jelezve, hogy ezek a beolvasott értékek (26-2. ábra).



26-1. ÁBRA BEOLVASAS.VI BLOKK DIAGRAMJA (ELSŐ FELE)



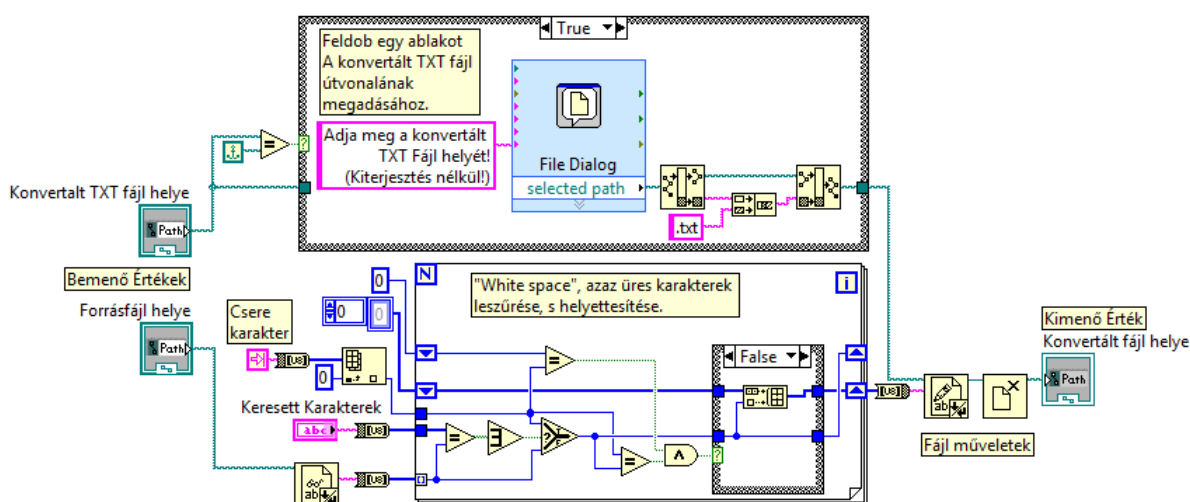
26-2. ÁBRA BEOLVASAS.VI BLOKK DIAGRAMJA (MÁSODIK FELE)

Konvertalo.vi (27.ábra)**27. ÁBRA KONVERTALO.VI**

Ez a VI felelős azért, hogy a „.txt” fájlt táblázatként beolvashassuk (28. ábra). A probléma, hogy az adatrögzítő program annak érdekében, hogy az adatok oszloponként jobbra legyen rendezve, változó hosszúságú szóközöket helyezett el. A „Read From Spreadsheet” paranccsal így feldolgozhatatlan mátrixot olvasnánk be. Ezért ezeket a változó mennyiségű szóközöket le kell cserélnünk valami egyébre. Mi egy tabulátorral választottuk el egymástól az oszlopokat. Ezért a műveletért felelős az alsó fele a programnak. A bemenő értéként kapott „Beolvasott fájl helye” útvonal alapján szöveggént beolvassa ezen „.txt” fájlt, majd át is alakítja string (azaz szöveges) formátumból „unsigned byte array” formába. Így minden karaktert a neki megfelelő kóddal helyettesít, s egy tömbbe helyezi őket (minden karakter kódot külön sorba). A programban alapértelmezetten, kívülről nem módosítható módon szerepelnek a keresett karakterek, melyeket le kívánunk szűrni, s a csere karakter, melyre a keresett karaktereket le kívánjuk cserélni. (A több egymás után következő keresett karaktert csak egy csere karakterre cseréli le.) Ezeket is átalakítjuk „unsigned byte” típusúvá. Az átalakításra a gépidő csökkentése végett van szükség, mivel nagyon sok adatunk van, melyek szöveggént való feldolgozása nagyon számolás igényes.

Az átalakított karaktereket, tömböket bevezetjük egy for ciklusba. A while ciklustól annyiban különbözik, hogy mindig megadott mennyiségű ciklust futtat le, s nem egy feltétel teljesülésére vár. Ha bemenetként tömböket vezetünk be a ciklusba, akkor a legnagyobb tömb hosszával megegyező számú ciklust fog elvégezni, s (ha csak át nem állítjuk), a tömbökből a ciklus sorszámaival megegyező sorszámu elemét olvassa be a tömbből. Így az átalakított szöveg meghatározza, hányszor fusson le és mindig ahányadik futásnál tart, annyiadik elemét olvassa ki, s végez vele műveletet. A ciklus

megvizsgálja, hogy az adott karakter megegyezik-e a keresett karakter bármelyikével. (Ezért felel az egyenlőség jel, s mivel több keresett karakter esetén azokat tömbként vizsgálja, így tömbös logikai értéket kapunk, amit a szummázással összesítünk. A szummázás eredménye igaz, ha legalább egy értéke igaz.) Ezen vizsgálat eredménye alapján egy „Select” függvény kiválasztja, hogy az adott karakter, vagy a helyettesítő karakter fusson tovább. Továbbá elvégez egy összehasonlítást az előző karakterrel is (felső egyenlőség). Ha nem a csere karakter fut tovább (alsó egyenlőség), vagy a mostani karakter nem egyezik meg az előző karakterrel, akkor a case struktúra a kimenő tömbhöz hozzáfűzi a karaktert. Amennyiben ismét egy csere karakter lenne hozzáfűzve, úgy a case struktúra a karakter nem fűzi hozzá a tömbhöz. A ciklus futása után a kimenő tömböt (azaz a szűrt szöveget unsigned byte-ként) visszaalakítjuk szöveggé, s a meghatározott útvonalra kiírjuk a „Write To Text File” VI segítségével (s be is zárjuk a kiírt fájlt, a „Close File” VI-al). A kiírási útvonal a VI felső felében lesz megvizsgálva. A módszer már korábban is szerepelt, amennyiben üres a bemenő útvonal feldob egy ablakot, melyben megadhatjuk (kiterjesztés nélkül, a „.txt”-t a program hozzáfűzi). Ha viszont nem üres a bemenő útvonal fog tovább futni (a fájl kiírásáért felelős VI-ba). A Fajlkezelő.vi már teljes útvonalat hozott létre, azaz a fájlnev is szerepel benne. Kimenő érték a Konvertált fájl helye lesz.

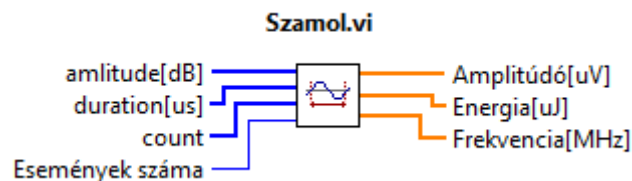


28. ÁBRA KONVERTÁLO.VI BLOKK DIAGRAMJA

5. blokk:

Az mérési adatokat tömbökként kapja meg **Szamol.vi**, ezenkívül megkapja ezen tömbök méretét is. Kivételt képez a **Time** tömb, ami az események kezdő időpontjait tárolja. Ez nem szükséges a **Szamol.vi** számára, ezért közvetlenül a következő blokkba megy tovább a vezetékek. Kimenetnek a maximum Amplitúdó μV -ban számított értékét, az esemény Energiáját, és a Frekvenciát adja ki.

Szamol.vi (29. ábra):



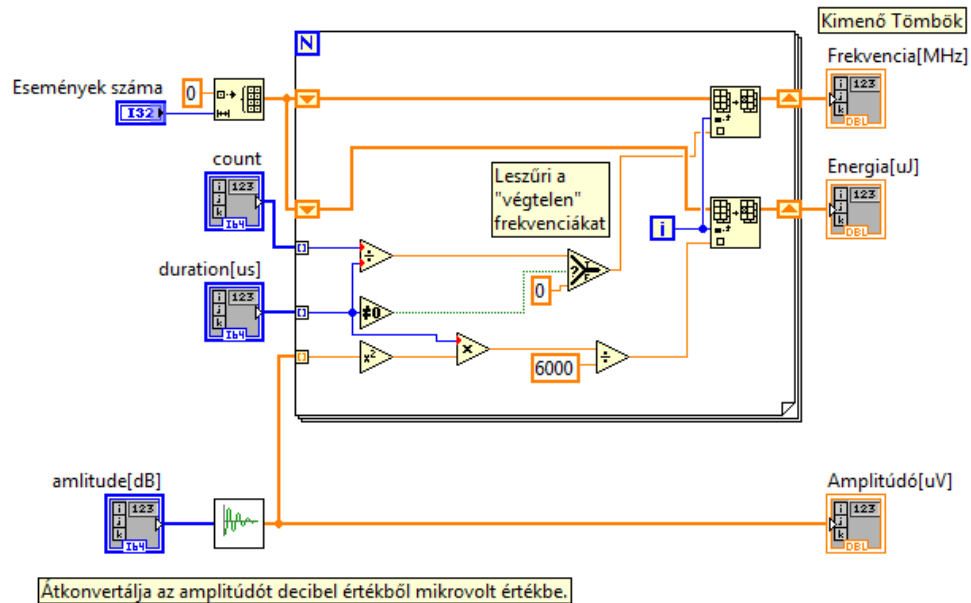
29. ÁBRA SZAMOL.VI

A frekvenciaszámításnál egy `for` ciklussal megyünk végig a tömb elemein. Ekkor minden ciklusban a ugyanazon eseményhez tartozó adatokból fogunk számolni. A frekvencia számításánál feltételezzük, hogy az esemény frekvenciája állandó. Ekkor a számítást egyszerűen a (2) képlet szerint végezhetjük:

$$\text{Frekvencia} = \text{count} / \text{duration} \quad (2)$$

ahol a *count* az esemény beütésszáma, a *duration* pedig az esemény időtartama. Mivel a *duration*, mikroszekundumban van megadva, ezért a kiszámolt frekvencia MHz-ben értendő.

Előfordulhat, hogy a *duration* értéke (például mérési hiba folytán) 0. Ez esetben az osztás nem elvégezhető. Ezeket a hibás eseményeket logikai kapuval szűrjük ki, és frekvencia, illetve energia értéküket 0-ra állítjuk be (30. ábra).



30. ÁBRA SZAMOL.VI BLOKK DIAGRAMJA

A maximum Amplitúdó átszámítása dB-ről μV mértékegységre egy külön VI-ban, a dBkonvertalo.vi-ban történik.

dBkonvertalo.vi (31. ábra):



31. ÁBRA DBKONVERTALO.VI

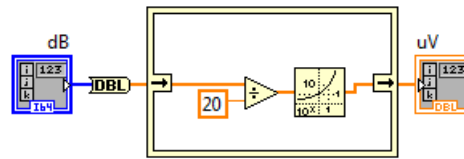
A dB számítás a (3) képlet alapján történik:

$$A = 20 \cdot \lg \frac{U}{U_{alap}} \quad (3)$$

Ezért visszaszámítása (4):

$$U = U_{alap} \cdot 10^{\frac{A}{20}} \quad (4)$$

ahol U a maximum Amplitúdó μV -ban értve, A az amplitúdó dB értéke, U_{alap} pedig a decibel alapérték, ez esetben $1 \mu V$. Mivel μV -ban számolunk, ezért az U_{alap} -pal való szorzást elhagyhatjuk. A blokk, ami körülveszi, egy In Place Element Structure. Ennek funkciója memóriaspórolás a számítás során (32. ábra).



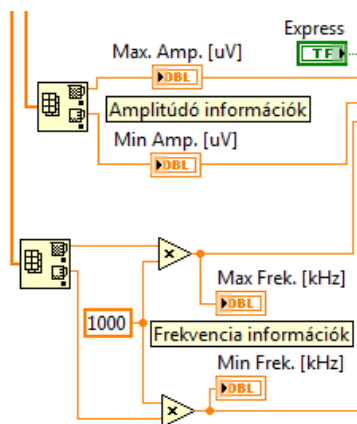
32. ÁBRA DBKONVERTALO.VI BLOKK DIAGRAMJA

Az energia számítása esetén a hullám burkológörbáját háromszög alakúként közelítjük. Ezzel az egyszerűsítéssel élve az energia képlete (5):

$$E = \frac{U^2}{6 \cdot R_0} \cdot duration \quad (5)$$

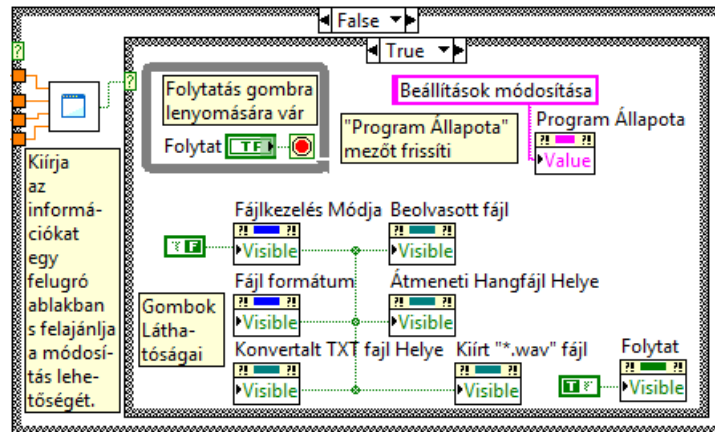
ahol E az energia μJ -ban mérve, az U az előzőekben kiszámolt μV -ban értendő maximum amplitúdó, R_0 egy műszerre jellemző állandó, értéke 1000Ω , a $duration$ pedig az esemény időtartama.

Miután a Szamol.vi elvégezte a számítást, a főprogram kiválasztja az amplitúdók és a frekvenciák legnagyobb, illetve legkisebb értékeit, és kijelzi azokat az előlapon. (33. ábra) Amennyiben nincs bekapcsolva az "Express" beállítás, akkor ezen információk alapján még módosíthatjuk a beállításokat. A frekvenciát 1000-rel szorozza, és kHz-ben írja ki.



33. ÁBRA FREKVENCIA ÉS AMPLITÚDÓ INFORMÁCIÓK KIÍRÁSA

Az alábbi rész kezeli a felugró ablakot, és az elemek láthatóvá, illetve láthatatlanná tételét (34. ábra):

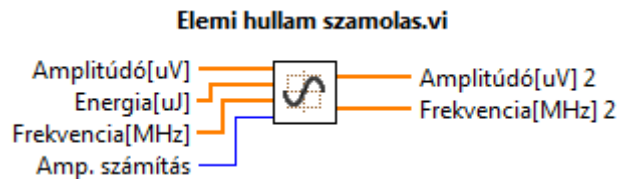


34. ÁBRA FELUGRÓ ABLAKOK ÉS LÁTHATÓSÁGOK KEZELÉSE

6. Blokk

A blokk fő részét az Elemi hullam szamolas.vi képezi.

Elemi hullam szamolas.vi (35. ábra):

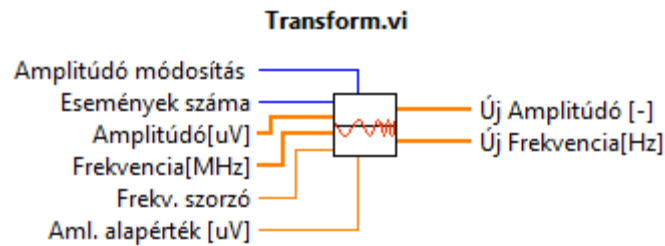


35. ÁBRA ELEMI HULLAM SZAMOLAS.VI

Ez számítja ki az elemi hullám amplitúdóját az alapján, hogy a beállításokban maximum amplitúdót, vagy energiát adtunk meg az amplitúdó számítás módjának.

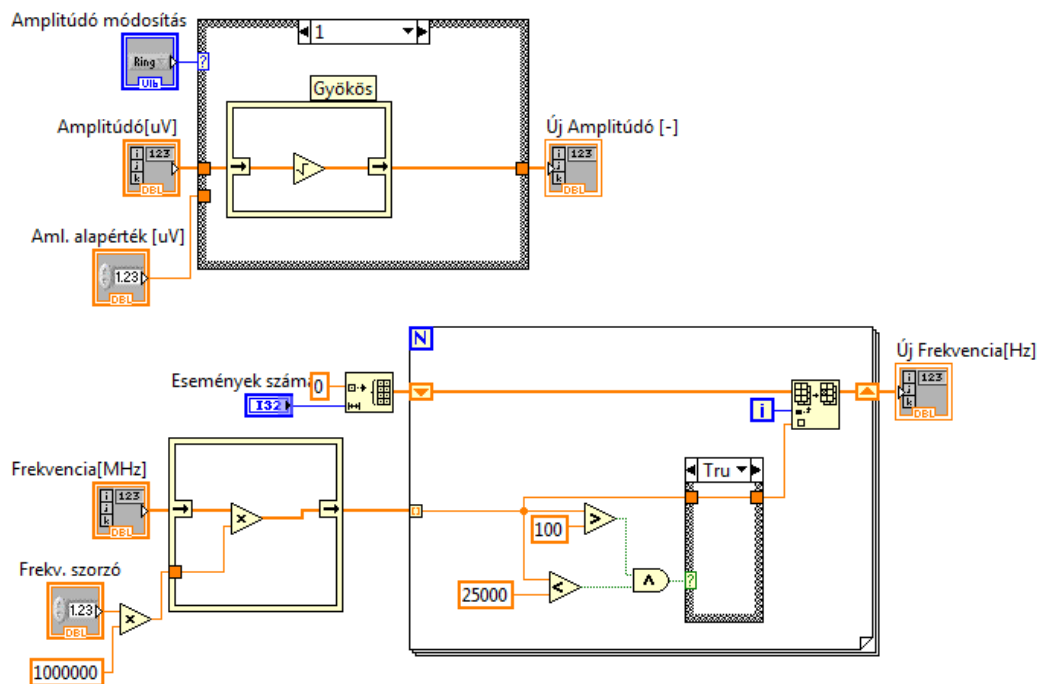
Az energia-alapú számítás esetén szükséges lenne még egy szorzószám, amivel az energiát amplitúdóra számítjuk át, de mivel a program a végén normalizálja a teljes jelet, ezért a konstans szorzónak nincs jelentősége. Az energiát tehát közvetlenül beköthetjük mint kimenő amplitúdó ($\text{Amplitúdó[uV]} \cdot 2$). Maximum amplitúdó alapú számításnál szintén közvetlenül kötjük össze a bemenetet a kimenettel, ugyanezen okból (37. ábra).

Transform.vi (39. ábra):



39. ÁBRA TRANSFORM.VI

Bemenete az előző blokkban kiszámolt Amplitúdó és Frekvencia, ezen kívül szükségesek még a módosítások paraméterei: a frekvencia szorzószáma, az amplitúdó módosítás típusa (arányos, négyzetgyökös, vagy logaritmikus), az amplitúdó alapértéke. Szükséges még ezen kívül a sorok száma is. Kimenatként kapjuk az eseményeket reprezentáló elemi hullámok végleges amplitúdóját (Új Amplitúdó [-]) és frekvenciáját (Új Frekvencia) (40. ábra).

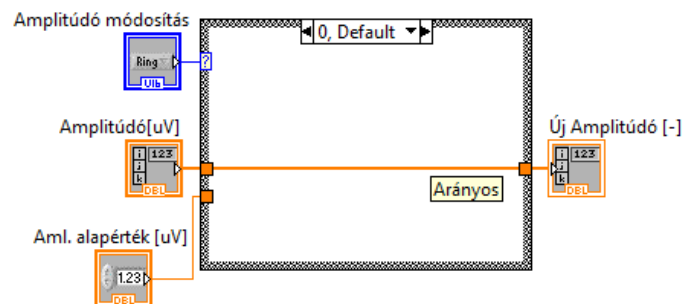


40. ÁBRA TRANSFORM.VI BOKK DIAGRAMJA

A program háromféleképpen tudja módosítani az amplitúdókat: arányosan, négyzetgyökösen és logaritmikusan. Az arányos megtartja az eredeti arányokat. Mivel azonban gyakori, hogy néhány kiugró jel mellett a többi esemény amplitúdója (vagy

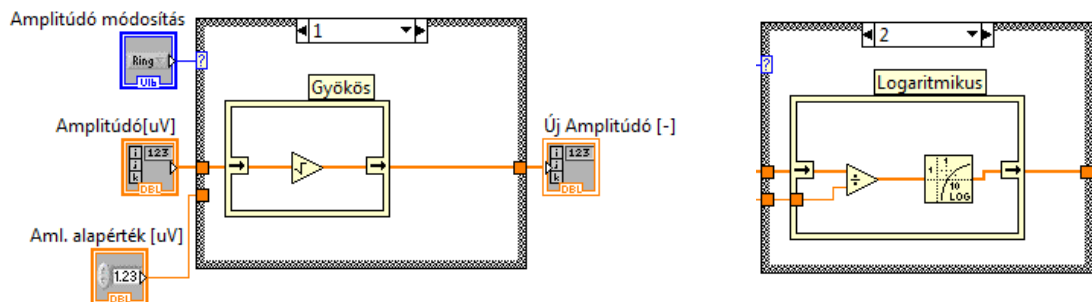
energiája) nagyságrendekkel kisebb, ezért a nagyok mellett a kisebb jelek alig lesznek hallhatóak. Ezen segít, hogy ha logaritmikusan skálázzuk az amplitúdót, ez azonban a különbségek túlzott "elkenését" eredményezheti. A kettő közt képez átmenetet a gyökös módszer.

Arányos amplitúdó módosítás esetén a kimenő amplitúdó egyenlő lesz a bemenő amplitúdóval. Arányszámra, fentebb részletezett okokból, a normalizálás miatt nincs szükség (41. ábra).



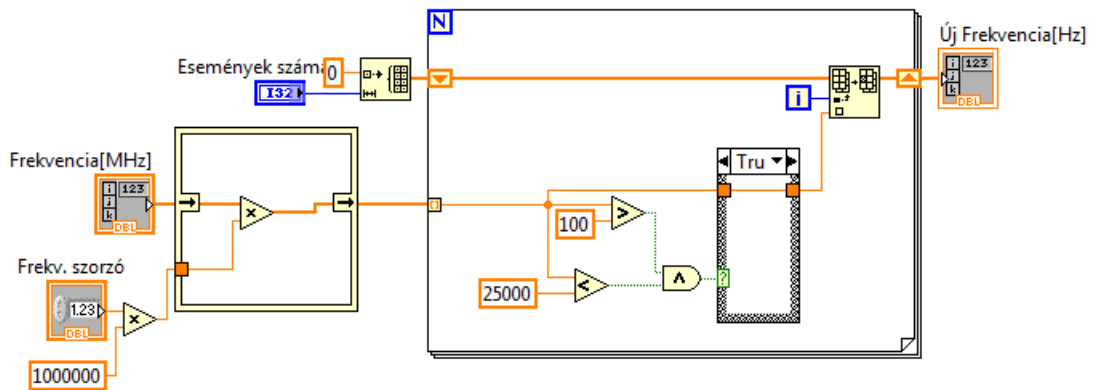
41. ÁBRA ARÁNYOS AMPLITÚDÓMÓDOSÍTÁS

Gyökös számítás esetén a program négyzetgyököt von az amplitúdóból, logaritmikus esetben pedig természetes alapú logaritmust. Ez utóbbi esetben még előtte elosztja az amplitúdót a megadott alapértékkel (42. ábra).



42. ÁBRA NÉGYZETGYÖKÖS ÉS LOGARITMIKUS AMPLITÚDÓMÓDOSÍTÁS

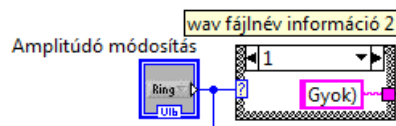
A frekvenciaszámításhoz először át kell váltanunk az értéket MHz-ről Hz-re, amit egy 1 000 000-val való szorzással végezhetünk el. Ezután még meg kell szoroznunk azt a frekvenciaszorzóval. A szorzások sorrendje a programban más, a számítás optimalizálása érdekében (43. ábra).



43. ÁBRA FREKVENCIAMÓDOSÍTÁS ÉS SZŰRÉS

Ezután még egy szűrés található: logikai kapuk segítségével 0-vá tesszük a 25 kHz-nél nagyobb, vagy 100 Hz-nél kisebb frekvenciákat. Ezek az emberi fül számára már biztosan nem hallhatóak, a hangkártya és a hangszórók számára azonban a nagyon nagy, és nagyon kicsi frekvenciák problémákat okozhatnak, emiatt kell ezeket "kinullázni". A számításokat In Place Element Structure-be helyeztük, a memória felhasználás javítása céljából.

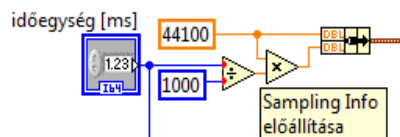
A program itt is hozzátesz egy tagot a fájlnevhez, az Amplitúdó módosítás módszere alapján értelemszerűen "Ar", "Gyok", vagy "Log" tagokat. (44. ábra)



44. ÁBRA FÁJLNÉV UTÓTAG BEÁLLÍTÁSA

8. Blokk

A 8. blokk feladata az elemi hullámok, valamint Sampling Info előállítása (45. ábra).



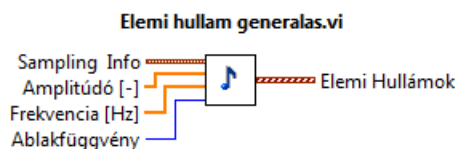
45. ÁBRA SAMPLING INFO ELŐÁLLÍTÁSA

Ez egy cluster típusú változó, mely tartalmaz egy mintavételezési frekvenciát, és a minták darabszámát. A mintavételezési frekvenciát szabványos értékre, 44100-ra állítottuk.

Az időegységet 1000-rel szorozva megkapjuk annak másodpercben mért értékét. Ha ezzel elosztjuk 44100-at, akkor olyan mintaszámot kapunk, amit felhasználva mindig valós idejű lesz a hang, azaz, olyan hosszú lesz egy elemi hullám, mint egy időegység. Ezután a két adatot egy **Bundle**-lel cluster-ré egyesítjük.

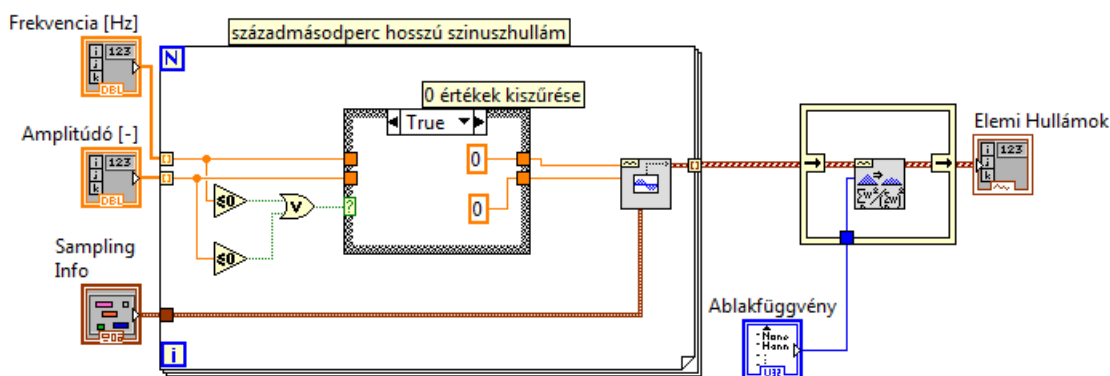
Az **Elemi hullam generalas.vi** készít hullámokat a megadott frekvencia és amplitúdó értékekből. A kimenet az elemi hullámokból álló waveform típusú tömb (**Elemi Hullámok**). A frekvenciákon és amplitúdókon kívül bemenet még a **Sampling Info**, és az **window** nevű ring típusú változó (**Ablakfüggvény**).

Elemi hullam generalas.vi (46.ábra)



46. ÁBRA ELEMI HULLAM GENERALAS.VI

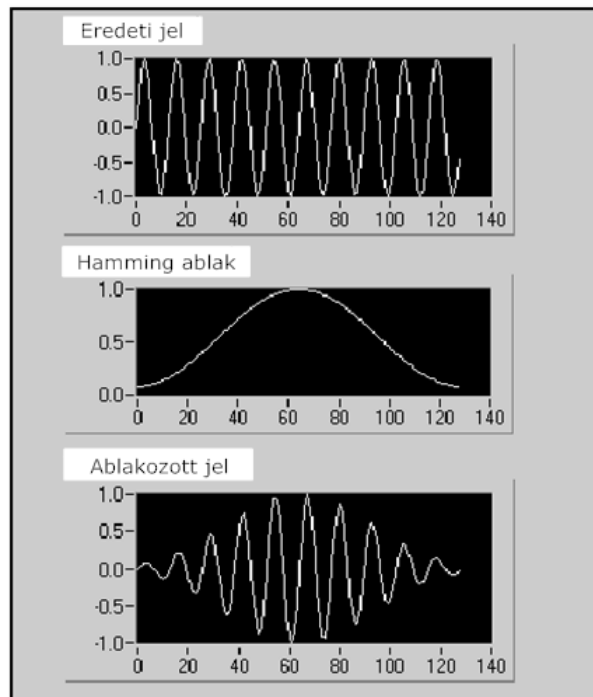
Az **Elemi hullam generalas.vi** egy for ciklussal egyenként hozza létre az amplitúdó-frekvencia párokhoz tartozó hullámokat a **Sine Waveform.vi** nevű VI-jal. Ehhez szükséges a **Sampling Info**. Előtte még egy szűrés történik, a 0 amplitúdójú vagy frekvenciájú pároknál mindkét adatot lenullázza, így az ezekhez tartozó hullám egy "üres", csupa nulla waveform lesz. A for ciklus tömböt hoz létre az elemi hullámokból (47. ábra).



47. ÁBRA ELEMI HULLAM GENERALAS.VI BLOKK DIAGRAMJA

Ezután következik az ablakozás, melyhez egy beépített VI-t alkalmazunk. Az ablakozó függvényt megadva a tömb összes tagját megszorozza a kiválasztott ablakozó függvénnyel. Sokféle különböző függvény használható erre a célra. Közös jellemzőjük, hogy mindegyik folytonos, és az adott intervallumon 0-ban kezdődik, és 0-ban végződik (vagy egy ahhoz közeli értéken), azon kívül pedig mindenhol nulla. Ezzel elérjük, hogy az elemi hullámok mindegyike szintén 0-ban kezdődjön, és 0-ban érjen véget (vagy annak közelében), elkerülve, illetve csökkentve a folytonossági hiány okozta zajt (48. ábra). [6]

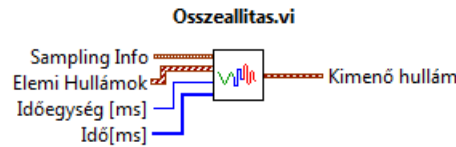
Az ablakozásról és annak LabVIEW-ban való megvalósításáról további információ található a [6, 10, 20] számú szakirodalmakban.



48. ÁBRA ABLAKOZÁS

9. Blokk

Itt található az *Osszeallitas.vi*, mely az előbb elkészített elemi hullámokból összerakja a kész waveformot. Bemenetnek kell még neki a *Sampling Info*, a *Time* tömb, és az időegység.

Osszeallitas.vi (49. ábra)

49. ábra Osszeallita.vi

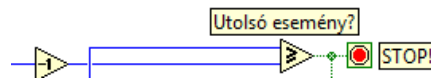
A VI működésének alapja, hogy az teljes időtartamot felosztja a beállított időegység méretű intervallumokra. Azok az események, amelyek azonos egységbe esnek, az azokhoz tartozó elemi hullámok összeadódnak, összhangzatot hozva létre. Természetesen, ha egy egységbe csak egy esemény jut, akkor abban az időben csak egy hang lesz hallható, ha pedig egy sem, akkor nem lesz hallható hang. Ez abból a megfontolásból lett így, hogy az egymás után gyorsan következő hangokat a fül úgy érzékeli, mintha egyszerre szólaltak volna meg. Ezen kívül ezzel elértük azt is, hogy az eredeti hangok rövideége ellenére hallhatóak lesznek a hangok. Kihasználjuk továbbá a technika fejlődése adta lehetőségeket is, ugyanis a 2001-es TDK-ban még nem volt meg a lehetőség egyszerre több hangfrekvencia lejátszására [11]. Ez a legbonyolultabb subVI, így a blokkdiagram bőséges kommentet tartalmaz a működés megértését segítő.

A VI legnagyobb részét egy while ciklus teszi ki. Előtte azonban még el kell végezni néhány fontos műveletet. Az elemi hullámok tömbjének méretét le kell kérdezni, mivel ez szolgáltatja majd a leállítási feltételt. Létre kell hozni a "reset" és a "gyűjtő" waveform-ok kezdeti értékét. Ezt egy **Sine Waveform.vi**-jal oldjuk meg, amplitúdónak 1-et, frekvenciának 0-t adunk meg, ezzel egy csupa nullát tartalmazó waveform-ot hozunk létre. Fontos még, hogy a `Sampling Info` bemenetre ugyanazt kell megadni, mint amit az elemi hullámoknál is használtunk, különben a hullámok nem lesznek kompatibilisek. Létre kell hozni még a kimenő hullám kezdeti értékét is. Ez egy waveform konstans, mely egy, nulla értékű mintát tartalmaz. Ezután a kompatibilitás értékében be kell állítani a dt (mintavételezési idő) paraméterét. Ezt a `Sampling Info` bemenet mintavételezési frekvenciájának reciprokja.

A cikluson belül több szál fut:

- "gyűjtő" waveform: ez gyűjti össze az egy időegységbe eső elemi hullámokat azáltal, hogy egymásra szuperponálja őket
- "reset" waveform: időegységnyi hosszú, csupa nulla waveform, mellyel "lenullázzuk" a gyűjtő waveformot, ha végére értünk az aktuális időegységnek
- esemény waveform: az aktuális eseményhez tartozó elemi hullámot hordozza
- kimenő waveform: ennek végére csatoljuk minden időegység végén a gyűjtő waveform tartalmát; a ciklus lefutása után ez a VI kimenete
- esemény számláló: egész típusú változó, mely az aktuális esemény indexét tartalmazza; mivel tömböt indexel, ezért 0-ról kezdődik
- időszámláló: egész típusú változó, mely azt jelzi, hogy hányadik időegységnél tart a program; 1-ről kezdődik

Minden ciklusban meg kell vizsgálni, hogy az aktuális esemény az utolsó-e. Ez esetben kell állítani a ciklust (50. ábra).

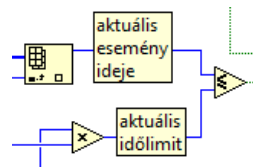


50. ÁBRA CIKLUS LEÁLLÍTÁSI FELTÉTEL ELLENŐRZÉSE

Ha a tömb mérete kisebb-egyenlő, mint az aktuális eseményszámláló, akkor a tömb végére értünk. A -1 azért kell, mert az eseményszámláló 0-ról indul.

Három fontos esetet kell megkülönböztetni: időegységen belül vagyunk, időegység végéhez értünk, vagy az utolsó eseményhez értünk. Első esetben az aktuális eseményt össze kell adni a gyűjtő waveform jelenlegi tartalmával, ezzel rászuperponálva a többi ehhez az időegységhez tartozó hangra. Ezután a következő eseményre kell lépni, az időegység azonban ugyanaz marad, tehát az eseményszámlálót növelni kell eggyel, míg az időszámlálót változatlanul kell hagyni. Második esetben a gyűjtő waveform tartalmát hozzá kell tenni a kimenő waveform végére, majd "ki kell üríteni". Ekkor a következő időegységre léphetünk (növeljük eggyel az időszámlálót), viszont mivel az aktuális eseménnyel nem foglalkoztunk, ezért ezt a következő ciklusra kell hagyni (az eseményszámlálót nem szabad növelni).

Harmadik esetben mindkettő szükséges: az aktuális eseményt hozzá kell adni a gyűjtő waveformhoz, majd ezt hozzátenni a kimenő waveform végére. Ezután a ciklus véget ér. Emiatt a számlálókkal fölöslegesen foglalkozni. Hogy melyik eset az aktuális, annak eldöntésére a program logikai kapukat használ (51. ábra):



51. ÁBRA IDŐLIMIT ELÉRÉSÉNEK ELLENŐRZÉSE

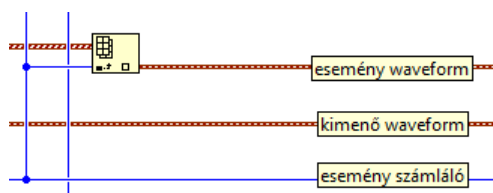
Az idő tömb eseményszámlálóval indexelt eleme lesz az aktuális eseményhez tartozó időpont. Az időszámláló és az időegység szorzásával megkapjuk az aktuális időegység felső határát. Ha az esemény ideje kisebb, mint ez a határ, akkor az időegységen belül vagyunk. Ekkor a kimeneti érték 1, azaz True. Különben átléptünk a következő egységbe, ekkor az érték 0, azaz False. Ezt egy case struktúrához használjuk (52. ábra).



52. ÁBRA A CASE STRUKTÚRA KÉT ÁLLAPOTA

Ha az érték False, akkor 0-t adunk kimenetnek. Ha True, akkor két eset lehetséges, melyet egy **Select** függvénnyel választunk szét: ha az utolsó eseménynél tartunk (a bemenet True), akkor 2 a kimenet, ha nem (False), akkor 1. A használt logikai érték számítását fentebb tárgyaltuk. Tehát az egyes esetekhez rendelt számok: az első eseté 1, a másodiké 0, a harmadiké pedig 2. Ezek után a program futása a következőképp zajlik:

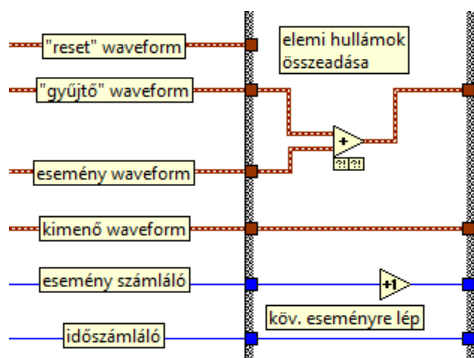
Először kivesszük az elemi hullámok tömbjéből az aktuális eseményét. Az indexet az eseményszámláló szolgáltatja (53. ábra).



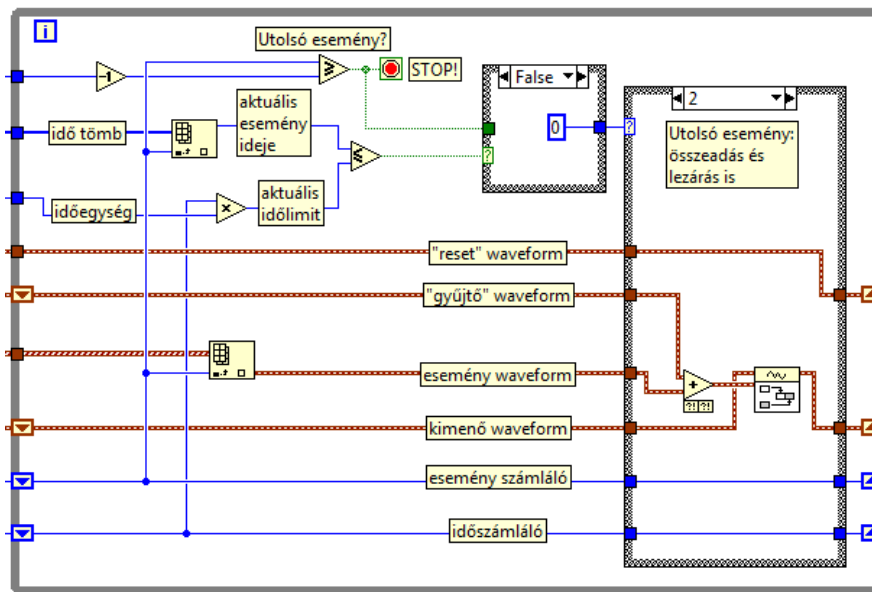
53. ÁBRA AKTUÁLIS ESEMÉNY KIVÁLASZTÁSA

Ezután a különböző szálak egy Case struktúrába futnak, melyeket a három esetnek megfelelően háromféleképp kezel.

Első eset: időegységen belül vagyunk (1) (54. ábra). Az esemény hullámát rá kell szuperponálni a gyűjtő waveform tartalmára. Ezt a LabVIEW intelligens műveletkezelése miatt egy egyszerű összeadással megoldhatjuk. Ez pontonként adja össze a hullámokat, de csak akkor működik, ha a hosszuk és a mintavételezési idejük egyforma. Ezért volt fontos az elején, hogy ugyanazt a Sampling Info-t használjuk. Ezután a gyűjtő waveform egy Shift Register-be fut bele, mely tárolja az értéket a következő ciklusig. Ha abban a ciklusban is ez az eset lesz érvényes, akkor ehhez lesz hozzáadva a következő esemény is. Így gyűjti össze az egy időegységbe tartozó elemi hullámokat a gyűjtő waveform. A kimenő waveform változatlanul fut bele egy másik Shift Registerbe. Az esemény számlálót növeljük, az időszámlálót azonban nem. A számlálók is Shift Register-ben tárolódnak ciklusról ciklusra.



54. ÁBRA ELSŐ ESET: IDŐEGYSÉGEN BELÜL VAGYUNK



56. ÁBRA HARMADIK ESET: UTOLSÓ ESEMÉNYHEZ ÉRTÜNK

11. Blokk

A program blokk felső részében ismét frissül a Program Állapota mező („Kész WAV létrehozására”), valamint legalul megtörténik az Átmeneti hangfájl útvonalának kijelzése.

A blokkban kiolvasásra kerül az Express nevű logikai változó „Local Variable” node-jából az értékét, mely vezérel egy eset (azaz „case” struktúrát). Amennyiben igaz („true”) az érték (azaz express módban futtatjuk a programot), egy fix igaz érték lesz beolvasva, mely a következő blokkban a hangfájl „wav” formátumúvá történő átkonvertálását fogja eredményezni. Amennyiben nem express módban futtatunk (igaz, vagyis „true” értékű az Express változó), felugrik egy párbeszédpanel, melynek megfelelő gombjára kattintva kiválaszthatjuk, hogy megtörténjen-e a „wav” formátumúvá konvertálás, avagy ne történjen meg. A konvertálásra vonatkozó logikai változó, valamint az Átmeneti hangfájl, a Kiírandó WAV fájl útvonala fog átadódni a következő blokknak.

12. blokk

A blokkban található eset („case”) struktúra vezérlésére befut az előző blokkban meghatározott igaz-hamis érték. Amennyiben az érték hamis, nem kerül sor az Átmeneti hangfájl „wav” formátumúvá átkonvertálása, s az útvonalához tartozó kijelző is üresen

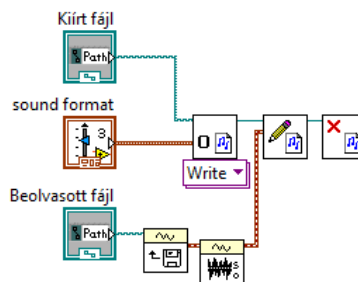
marad. Amennyiben a konvertálást választottuk a Fájlfelkezelés módja beállítástól függően a belső eset struktúra Automatikus vagy Manuális lapja kerül futtatásra. Manuális esetben a már ismert módon próbálja meg beolvasni a kezelő felület megfelelő kontroll mezejéből az útvonalat, s amennyiben üres, feldob egy párbeszédpanelt. Automatikus esetben az átmeneti hangfájl nevéből eltávolítja a „HANG-„ előtagot, s a kiterjesztést lecseréli „.wav”-ra. A wav fájl útvonalat fut tovább egy subVI-ba, melynek neve Hangfajlkonvertalas.vi. Ez a subVI fogja a konvertálást végrehajtani.

Hangfajlkonvertalas.vi (57. ábra):



57. ÁBRA HANGFAJL KONVERTALO.VI

Ebben a subVI-ban kívülről nem módosítható módon szerepel a fájlformátum. Miután beolvasta az átmeneti hangfájlt normalizálja azt, s a formátumnak megfelelően kiírja a korábban megadott vagy generált útvonalra (58.ábra). Ezen VI-t külön is használhatjuk átkonvertálásra, az átmeneti hangfájl bemeneti fájlként megadva.



58. ÁBRA HANG ÁTKONVERTÁLÁSA

13. blokk

Felül ismét frissül a Program Állapota mező, most a „Program Lefutott!” szöveggel, valamint ugyanezzel a felirattal egy gombos párbeszédpanelt is. Ezen felül minden gombot láthatóvá tesz, mégpedig úgy, hogy az egyes gombok láthatóságáért felelős property node-ba igaz értékű logikai konstanst írunk be.

4. Mérések

A szakítóvizsgálatot Zwick Z005-ös számítógép vezérlésű terhelő géppel végeztük szobahőmérsékleten, 2 mm/perc terhelési sebességgel, 60 mm-es befogási hosszal, 2 N előterheléssel.

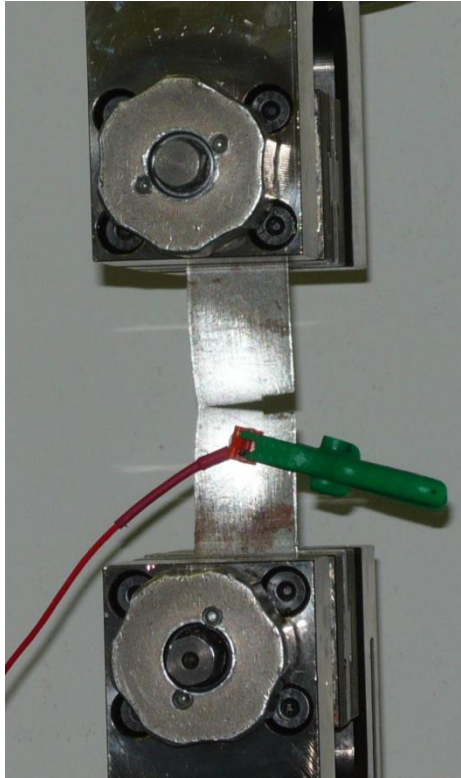
Célunk a terhelés során keletkező hanghullámok detektálása és kiértékelés volt. A mérési elrendezés az 59. ábrán látható.



59. ÁBRA EGY BEHELYEZETT PRÓBATEST KÖZVETLENÜL A MÉRÉS MEGKEZDÉSE ELŐTT, AKUSZTIKUS EMISSZIÓS SZENZORRAL

A mérés első lépésében a mérendő anyagból létrehozott próbatestet befogjuk a gép két erre szolgáló befogó pofába. Miután meggyőződünk, hogy a minta a mérés folyamán biztosan nem fog a rögzítésből kicsúszni megkezdjük a mérést. Program által vezérelt módon a két fél közül egyiket adott sebességgel kezdi a gép mozgatni, megnyújtva ezzel a próbatestet egészen a minta elszakadásáig. A számítógép rögzíti az erőt, s az elmozdulás, s ezek segítségével kirajzolja az erő-elmozdulás diagramot. Ezen diagramból tudjuk az anyagra vonatkozó mechanikai információkat meghatározni.

A próbatesteken szokás bevágás ejteni, mellyel „kontrolláljuk” a repedés helyét, amelyet a nem kívánt inhomogenitások fognak meghatározni, melyeket előre nem ismerünk. Amennyiben van bevágás, bemetszés a szakadás ott fog bekövetkezni (60. ábra).



60. ÁBRA PRÓBATEST MÉRÉS KÖZBEN

Az akusztikus emissziós szenzort a próbatestre rögzítettük a befogása után, s a mérés folyamán rögzítjük az eseményeket. A szenzorhoz vazelin szilikátsírt alkalmaztunk csatoló folyadékként.

Felhasznált anyagok: farost lemez
hullámkarton,
alumínium,
bazalt szöveterősítésű epoxi,
szénszövet erősítésű epoxi,
üvegszál erősítésű epoxi,
lemezhengerelt acél,
önerősítésű PET,
üvegszál erősítésű PP.

Minden próbatesten látható a bemetszés, mellyel megszabtuk a szakadás helyét (61.ábra).



61. ÁBRA VIZSGÁLT PRÓBATESTEK (VIZSGÁLAT ELŐTT)

5. Futási eredmények

Ebben a fejezetben lefuttatjuk a programot a mérési eredményeken. Néhány példával illusztráljuk a különböző beállítások hatását, majd egyforma beállítást használva összehasonlítjuk egymással a különböző anyagok hangjait.

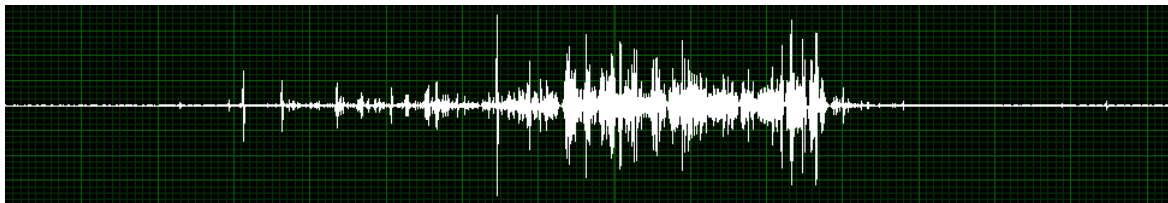
5.1. Beállítások hatásának bemutatása

A következőkben néhány példán keresztül bemutatjuk a különböző beállítási lehetőségeket és azok hatását az futás eredményeként kapott hangfájlokra. A példákhoz a farostlemez mérési adatait használtuk fel. A hangfájlok megtalálhatóak a CD mellékleten. A fájlok „.wav” formátumúak, és bármilyen hanglejátszó programmal futtathatók. Mi VLC Media Player-t használtunk a célra. Hangszerkesztő szoftverrel (pl. Cool Edit Pro 2.1) megnyitva a hanghullám grafikonja is megtekinthető.

Az összehasonlítás alapjául legtöbbször az alapértelmezett beállításokkal lefuttatott fájl szolgál, mely *farost(alap).wav* néven található meg. Ennek beállításai:

- Frekvenciaszorzó: 0,01
- Ablakfüggvény: Hanning
- U_0 : 1 μ V
- Időegység: 10 ms
- Amplitúdó számítás: maximum amplitúdóból
- Amplitúdó módosítás: arányos

Az 62. ábrán a *farost(alap)* generálásakor a program által kirajzolt hanghullám látható.

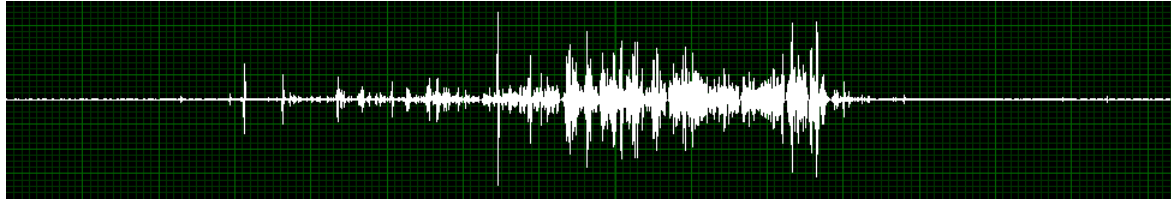


62. ÁBRA FAROST HANGHULLÁM (ALAP BEÁLLÍTÁSSAL)

Frekvenciaszorzó

Az alapértelmezett beállítás 0,01. Futtatáskor látható, hogy a maximális frekvencia 500 kHz, amit a program így 5 kHz-re transzformál. Ez még bőven a hallható tartományban található. A szorzót 0,005-re beállítva egy oktávval mélyebb hangot kellene tapasztalnunk. A

futtatás eredménye a *farost(0.005).wav* fájlban hallható. A hang a várakozásoknak megfelelően mélyebb, de ugyanolyan jellegű. Míg az alapbeállításokkal futtatott fájl hangja leginkább csipogásra és madárcsiripelésre emlékeztet, addig a második futtatási eredmény leginkább vízcsobogásra hasonlít. A grafikonjuk alapján szinte megkülönböztethetetlen a kép hang (63. ábra).

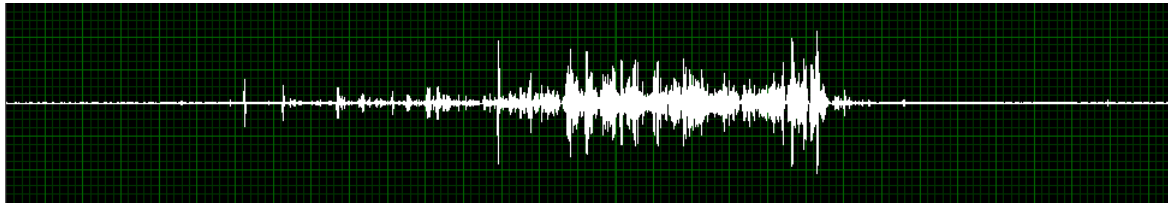


63. ÁBRA FAROST 0,005-ÖS FREKVENCISZORZÓVAL

Nagyobb szorzót használva a hang magasabbá válik. 0,02-t beállítva a hang „cincogássá” válik (*farost(0.02).wav*), grafikonján azonban még mindig látszik, hogy alapvető jellege ugyanolyan.

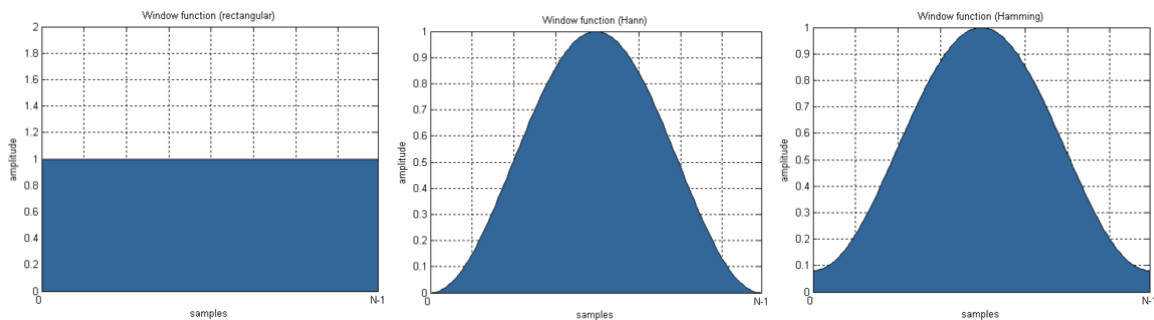
Ablakfüggvény

Az ablakozás funkciója, hogy az elemi hullámok csatlakozásánál fellépő folytonossági hiányt és az ebből fakadó zajt megszüntesse. Hogy bemutassuk ezt, a programban "Rectangle" azaz téglalap típusú ablakfüggvényt állítottunk be. Ennek értéke az intervallumon végig 1, azaz effektíve úgy működik, mintha nem lenne ablakozva a jel. Az eredményül kapott *farost(Rectangle).wav*-ot meghallgatva végig erős zajt tapasztalhatunk mindenhol, kivéve, ahol nincs egyáltalán esemény. Ehhez képest az alapbeállítás eredményét meghallgatva ugyanazt hallhatjuk, csupán a zaj nélkül. A grafikonokon is látszik, hogy nincs jelentős különbség, azaz a zajt sikerült megszüntetni anélkül, hogy a hang jellegét megváltoztattuk volna. Csupán annyi látható, hogy a hullámok kicsit "szögletesebbek" (64. ábra).

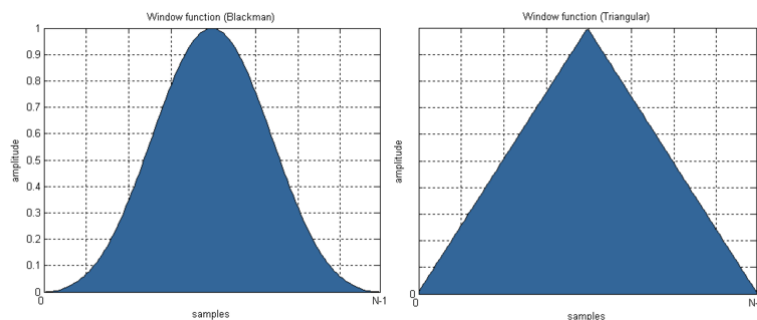


64. ÁBRA FAROST HANGHULLÁM "RECTANGLE" ABLAKZÁSSAL

Az alapbeállítás Hanning (vagy más néven Hann) függvényt használ, de mi néhány másik ablakfüggvényt is kipróbáltunk összehasonlítás céljából: Hamming (65. ábra), Blackman és „Triangle”, azaz háromszög függvény (66 ábra).



65. ÁBRA ABLAKOZÁSOK (BALRÓL JOBBRA): „RECTANGLE”, „HANN”, „HAMMING”



66. ÁBRA ABLAKOZÁSOK(BALRÓL JOBBRA): „BLACKMAN”, „TRIANGLE”

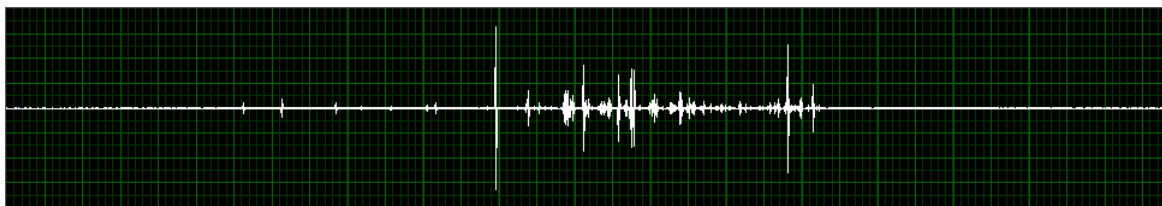
Ezek futtatási eredményei megtalálhatóak a mellékletben, értelemszerű elnevezéssel. Hamming függvény esetén hallható némi zaj, mivel ez nem nullából nullába tart, így nem szünteti meg, csupán csökkenti a folytonossági hiányt. A háromszögfüggvény a szögletes jelalak miatt kissé recsegős hangot produkált, a Blackman függvényesetében azonban nincs hallható különbség az eredetihez képest.

Időegység

Az időegység alapértelmezett értéke 10 ms, azaz 0,01 s. Ezt annak alapján állítottuk be, hogy az egy időegységbe eső hangokat már a fül ne tudja időben különválasztani. Ez azt jelenti, hogy a hangokat úgy halljuk, mintha egyszerre lennének, így nem változtat a hang jellegén, ha tényleg egyszerre játsszuk le őket. Ha az időegységet feljebb vesszük a 10-szeresére, azaz 100 ms-ra, akkor viszont az elemi hullámok már elkülöníthetővé válnak az emberi fül számára (*farost(100)*). Ha csak 30 ms-os beállítást használunk, akkor már kevésbé hallatszanak külön a hangok, viszont az elemi hullámok hossza miatt mégis hallható különbség az alapértelmezett beállításhoz képest (*farost(30)*). Ha az időegységet lecsökkentjük 3 ms-ra, ugyan jobb "felbontást" kapunk, azonban a rövid elemi hullámok miatt nem igazán lehet egyértelmű hangmagasságokat érzékelni, az emberi fül számára inkább csak sistergésnek tűnik (*farost(3)*). Ez alapján a 10 ms-os alapbeállítás nagyjából optimálisnak tekinthető.

Amplitúdó számítás

Az amplitúdó számítás alapértelmezetten a maximális amplitúdókkal történik. Logikusabb megfontolás lehet, hogy az egyes események energiájával legyen arányos az amplitúdó. Futtatás után már a grafikonban is jelentős különbséget láthatunk (67. ábra):

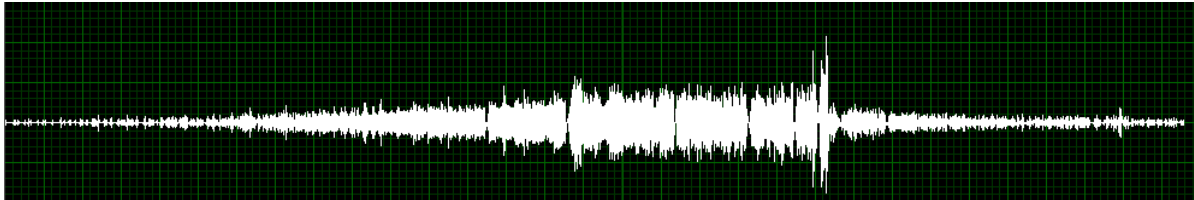


67. ÁBRA FAROST HANGHULLÁM ENERGIA ALAPÚ AMPLITÚDÓ SZÁMÍTÁSSAL

Ilyen beállítással az eredményt a *farost(energia)* néven találhatjuk. Azt tapasztalhatjuk, hogy ezzel a beállítással sokkal kevesebb nagy amplitúdójú hangot hallunk, illetve látunk. Ez arra utal, hogy kevés olyan esemény van, melynek amplitúdója és hossza is jelentős. Ennek ellenére a jellemzőbb "vonások" beazonosíthatóak itt is az alapbeállítású hang grafikonjához képest.

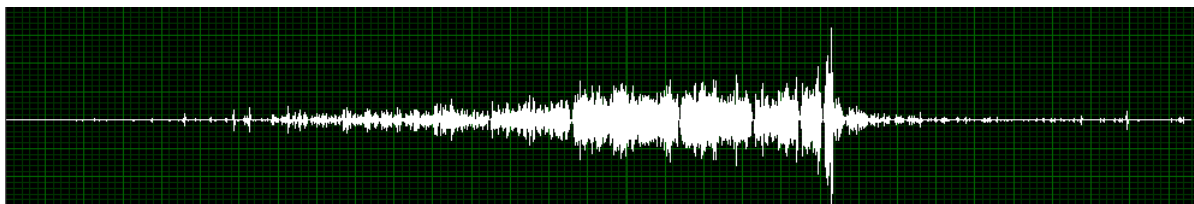
Amplitúdó módosítás

Ez a beállítás szintén jelentős változásokat okoz a hullámban. Az alapértelmezett arányos helyett logaritmikust használva a gyengébb események is felerősödnek, mely jól hallatszik a fájlban (*farost(log)*), és jól látszódik a grafikonon is (68. ábra):



68. ÁBRA FAROST HANGHULLÁM LOGARITMIKUS AMPLITÚDÓ MÓDOSÍTÁSSAL

Ha megnöveljük az U_0 értéket, azt tapasztaljuk, hogy a jelleg megmarad, de az amplitúdó csökken. Ezt a *farost(log100)* fájlban hallgathatjuk meg, $U_0=100 \mu V$ beállítással. Itt látható a hozzá tartozó diagram (69. ábra):



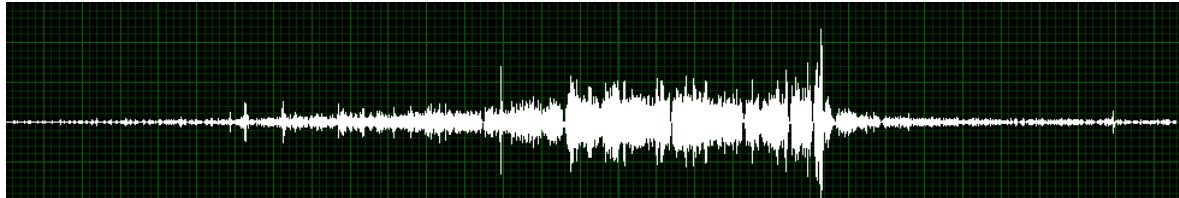
69. ÁBRA FAROST HANGHULLÁM $100 \mu V$ -OS BEÁLLÍTÁSSAL

A módosításhoz használt képlet átalakítható a logaritmikus azonosságok felhasználásával (6):

$$A^* = \lg \frac{U}{U_0} = \lg U - \lg U_0 \quad (6)$$

ahol A^* a módosított amplitúdó, U az (eredeti) maximum amplitúdó, U_0 pedig az amplitúdó alapértéke. Ennek alapján látszik, hogy U_0 növelése konstans értékkel csökkenti az össze amplitúdót. A kisebb események amplitúdója így 0 alá csökkenhet, de ezeket a program kiszűri, így nem jelennek meg a hangban.

Ez a beállítás nagyon "elkeni" az amplitúdóbeli különbségeket, így átmeneti megoldásként született meg a négyzetgyökös módosítás lehetősége. A hangfájl (*farost(sqrt)*) és a grafikon alapján ez az átmenet meg is valósul: a kisebb események itt is megjelennek, ennek ellenére látható rajta az arányos számításnál megjelenő lefutás jellege (70. ábra):



70. ÁBRA FAROST NÉGYZETGYÖKÖS AMPLITÚDÓ MÓDOSÍTÁSSAL

5.2. Mérési eredmények kiértékelése

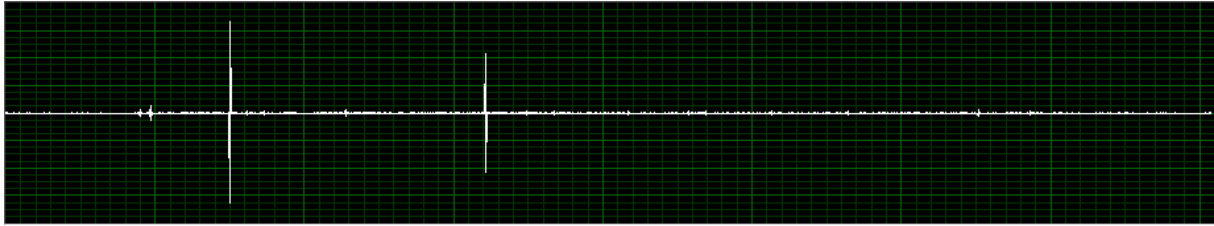
A különböző minták összehasonlítása csak egységes beállításokkal lehetséges. Ezek a beállítások:

- Frekvenciaszorzó: 0,01
- Ablakfüggvény: Hanning
- U_0 : 1 μ V
- Időegység: 10 ms
- Amplitúdó számítás: energia
- Amplitúdó módosítás: négyzetgyökös

Olyan beállításokat igyekeztünk találni, amely minden mérési adat esetén optimális. A fenti beállítások nagyjából megfelelnek a célra, de néhány esetben máshogy is lefuttattuk a programot. Ezeket külön jelezzük.

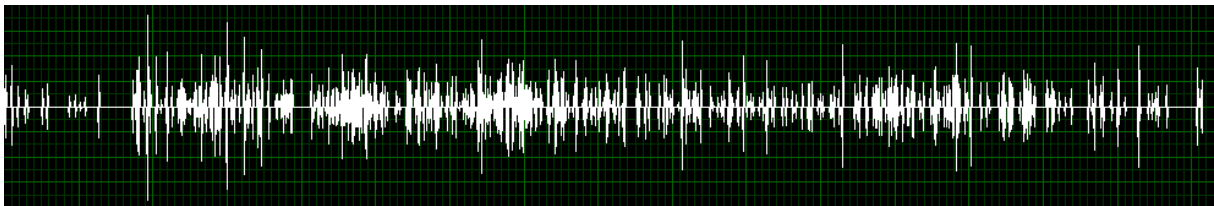
A hangokat Cool Edit Pro 2.1 programmal játszottuk le. Ezzel előtte „reverb” effektel lecsengést adtunk a hangoknak, hogy jobban hallhatóak és kellemesebbek legyenek. "Medium auditorium" típusú zenetűt használtunk hozzá.

Lemezhengerelt acél (71. ábra):



71. ÁBRA LEMEZHENGERT ACÉL NÉGYZETGYÖKÖS MÓDOSÍTÁSSAL

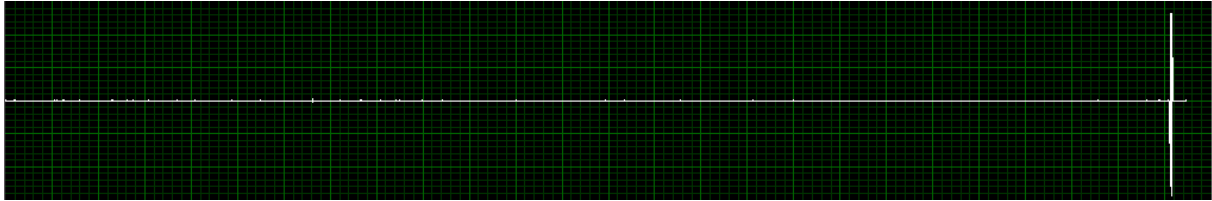
Az ábrából látható, hogy ezesetben két nagy esemény mellett szinte jelentéktelen a többi, a négyzetgyökös skálázás ellenére. Hallani se lehet többet, mint amit látunk, ezért ezt a fájlt logaritmikus módosítással, 1 μ V-os alapértékkel is lefuttattuk: (72. ábra)



72. ÁBRA LEMEZHENGERT ACÉL LOGARITMIKUS MÓDOSÍTÁSSAL

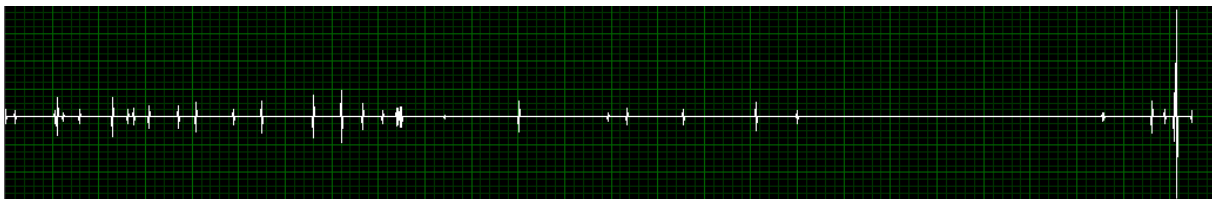
Ekkor a nagy események vesznek szinte el a többi közt, viszont így már némiképp érdekesebb a hang. Az események elég ritkásak, kevés az összhangzat. Emiatt ez adja a legtisztább "dallamot", és a hangok közti szünetek miatt ritmus is kialakul. A hangmagasságok viszonylag hasonlóak, nem túl nagy intervallumban mozognak. Mivel a próbadarabot nem sikerült elszakítani, így a végén nem található az ehhez kapcsolódó hangkitörés és az azt követő csönd, hanem végig találhatóak események.

Alumínium (73. ábra):



73. ÁBRA ALUMINIUM HANGJÁNAK DIAGRAMJA

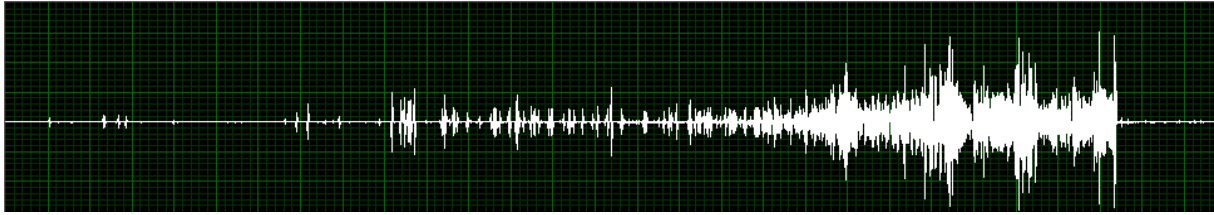
Az alumínium még az acélnál is kevesebb eseményt produkál. Egyedül az utolsó, feltehetően a szakadáshoz kapcsolódó hang jelentős, valószínűleg több esemény összege lehetett ez a nagy jel. Hogy több információhoz jussunk, ezt is lefuttattuk logaritmikusan is (74. ábra):



74. ÁBRA ALUMINIUM HANGJA LOGARITMIKUS MÓDOSÍTÁSSAL

Ebből már látható, hogy most még az acélnál is sokkal kevesebb eseménnyel van dolgunk. Annak ellenére, hogy nagyon ritkás, mégis hallható néhány két-három hangból álló összhangzat. A hangmagasságok keskeny sávban mozognak, kivéve a legvégét, ahol zárásként egy sokkal mélyebb, erősebb, többszörös összhangzatot hallhatunk.

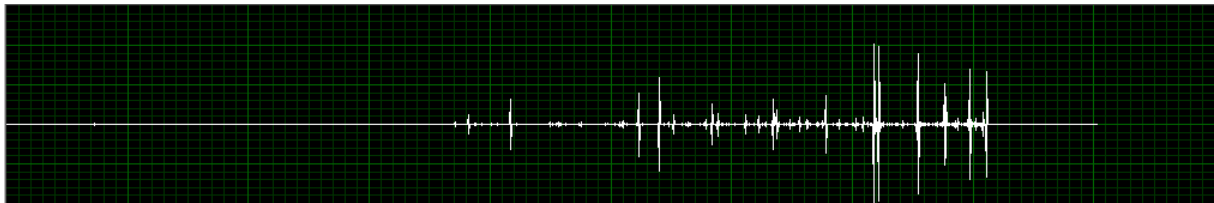
Bazaltszövet erősítésű epoxi (75. ábra):



75. ÁBRA BAZALTSZÖVET ERŐSÍTÉSŰ EPOXI HANGJÁNAK DIAGRAMJA

Ez már négyzetgyökös módosítással is elég sok információt tartalmaz. Láthatóan sokkal több esemény történt, mint az előzőekben, és ezek amplitúdója közelebb is van egymáshoz. A jól hallható hangok mellett végig található egy folyamatos, halk "csobogás". A hangmagasságok az előzőekhez hasonlóak a minta első részében. Az utolsó harmadban elkezdnek sűrűsödni az események, és itt már előfordul több mélyebb hang is. Ezek összhangzatban hallhatóak, kitörésszerűen. Valószínűleg valami nagy energiájú folyamatot jelez, ami akár eredetileg is hallható frekvenciájú volt, pl. szálszakadás. A legnagyobb ilyen kitörés a végén található, valószínűleg a szakadást jelzi.

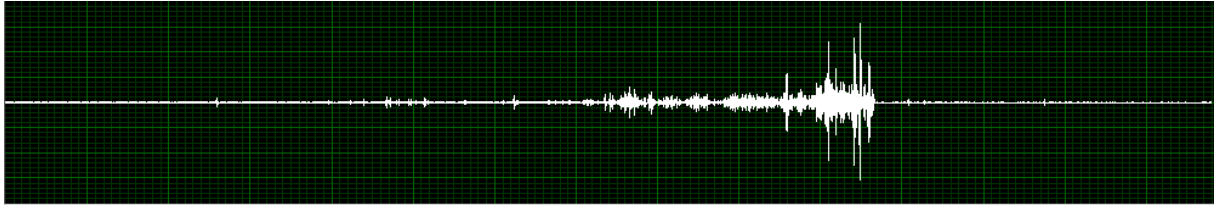
Szénszövet erősítésű epoxi (76. ábra):



76. ÁBRA SZÉNSZÖVET ERŐSÍTÉSŰ EPOXI HANGJÁNAK DIAGRAMJA

Sokkal ritkásabb, mint az előző, annak ellenére, hogy a mátrix anyaga ugyanaz. Az első egyharmadban szinte semmi nem hallható. Valószínűleg ez a Kaiser-effektus miatt van, mivel a mérés csak harmadszorra sikerült; de lehetséges, hogy az erősebb adhézió miatt kevesebb az esemény. Utána is jóval kevesebb jelentősebb esemény van, viszont megjelenik az előzőben is hallott halk "csobogás". Kevesebb az összhangzat is. A hangok jellemzően jelentősen mélyebbek, mint a korábbiaknál, de a végén kissé emelkedik. Érdekes módon nem a szakadáshoz kötődik a legnagyobb hangintenzitás.

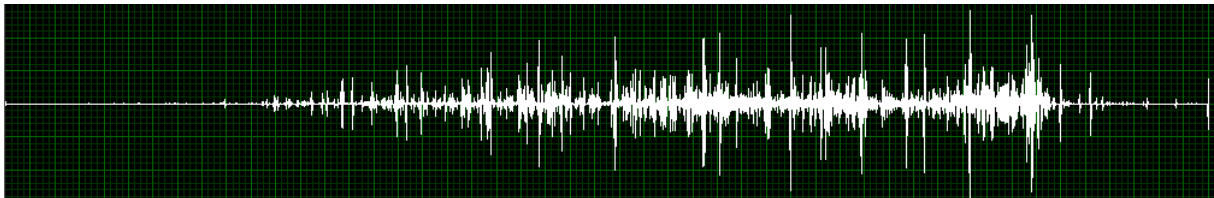
Üvegszál erősítésű epoxi (77. ábra):



77. ÁBRA ÜVEGSZÁL ERŐSÍTÉSŰ EPOXI HANGJÁNAK DIAGRAMJA

Ez a minta átmenetet képez a másik két epoxi között. Fontos különbség azonban, hogy bár elég "sűrűek" a hangok, itt csak közvetlenül a szakadás előtt találhatók nagyobb amplitúdójú kitörések, előtte sokkal egyenletesebbek az amplitúdók. A hosszú, lapos rész a végén már feltehetően a szakadás után van. Hangmagasságban szintén az előző kettő közé esik.

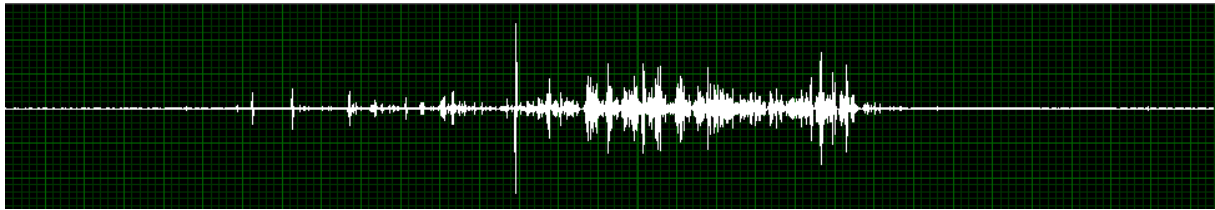
Hullámkarton (78. ábra):



78. ÁBRA HULLÁMKARTON HANGJÁNAK DIAGRAMJA

Igen sok eseményt tartalmaz ez a minta. Igen változatos amplitúdójú, bár megfigyelhető rajta egy növekedő tendencia. Az erősebb hangok eléggé kitűnnek a többi közül, az acélhoz hasonló dallamot hozva létre, de ezt kísérik a folyamatosan erősödő és sűrűsödő gyengébb hangok. A frekvenciasávja is szélesebb, bár a mélyebb hangok inkább az utolsó harmadra jellemzőek. Érdekes, hogy nincs egyértelmű vége, bár van a végénél van egy erős összhangzat, ami feltehetően a szakadásra utal, de még ezután is vannak jelentősebb események.

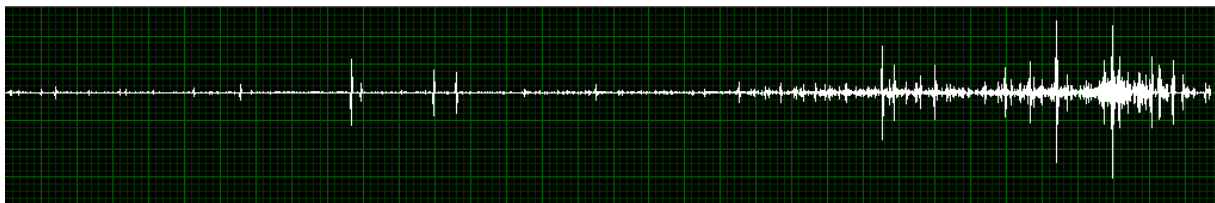
Farostlemez (79. ábra):



79. ÁBRA FAROSTLEMEZ HANGJÁNAK DIAGRAMJA

Elsőre feltűnik, hogy a legnagyobb amplitúdó nem a szakadáshoz kapcsolódik, hanem középen található. Ez előtt viszonylag ritkásabbak az események, míg utána sűrűbbek, és jellemzően nagyobb amplitúdójúak is. A hangból kihallatszik néhány erősebb esemény hangja, melyek a kartonhoz hasonló, de kevésbé jellegzetes dallamot adnak ki. A hangskála szintén viszonylag széles, de ennél inkább a magasabb hangok dominálnak. A szakadás jól felismerhető a végén hallható összhangzattól, és itt nem találhatóak már jelentős események ezután.

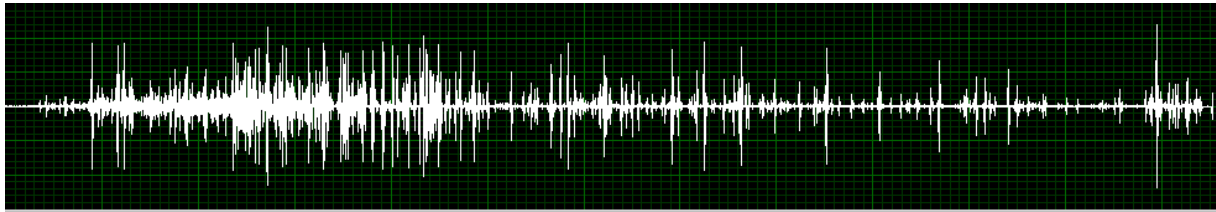
Üvegszál erősítésű polipropilén (80. ábra):



80. ÁBRA ÜVEGSZÁL ERŐSÍTÉSŰ POLIPROPILÉN HANGJÁNAK DIAGRAMJA

Ennél több mint az első felében csak néhány nagyobb eseményt hallhatunk, azon kívül csak a nagyon halk "pityergést". Ezután kezdenek csak fokozatosan sűrűsödni az események, és emellett erősödni is. A szakadás valószínűleg a legnagyobb amplitúdónál következett be, mivel ez egyben egy összhangzat is volt. Itt azonban nem ér véget a hang, még elég sok minden hallható, mire teljesen lecseng. A hangmagasság inkább magas, de a vége felé néhány mélyebb hang is van, jellemzően a nagy amplitúdójúak.

Önerősítésű polietilén tereftalát (81. ábra):



81. ÁBRA ÖNERŐSÍTÉSŰ POLIETILÉN TEREFTALÁT (PET) HANGJÁNAK DIAGRAMJA

Ez a leghosszabb hangminta az összes közül, hossza több mint öt perc. Feltűnő még, hogy ennek az elején található több esemény, míg a vége fele ritkásabb, és az amplitúdók is láthatóan csökkennek. Ez pont ellentétes, mint amit a többi anyagnál tapasztaltunk. A végén a szakadást feltehetően a legnagyobb amplitúdó jelzi, de nem annyira jellegzetes a hang, hogy biztos legyen, és utána csökkenés helyett még nő is az események intenzitása, így nehéz megmondani, mikor történt pontosan a szakadás. Az amplitúdókban nagy eltérések tapasztalhatóak, de időben viszonylag egyenletes, nincs olyan jellegzetes tendencia benne, mint a legtöbb másik anyagnál. A hangskála itt főleg magasabb hangokból tevődött össze, bár az utolsó harmadban található néhány mélyebb hang is.

5.3. Konklúzió

Az egyes anyagok közt a legnagyobb különbséget az jelentette, hogy fémről, vagy kompozitról van szó. Míg a fémek esetében kevés, és néhány kiugró kivételtől eltekintve igen gyenge és kevés esemény volt, addig a kompozit anyagok hangja sokkal "eseménydúsabb", nem csak darabszámot tekintve, de a relatív nagyobb amplitúdók alapján is. Ez valószínűleg abból fakad, hogy az összetett szerkezetű kompozit anyagokban sokkal több lehetőség van nagy energia felszabadulásokra (szálkihúzóadás, szálszakadás stb.), míg fémek esetén nagyrészt csak mikro repedések kialakulásából és terjedéséből, illetve kristályszerkezeti átalakulásokból származnak a hangok, melyek intenzitása jóval csekélyebb. Ezenkívül fémeknél a homogén szerkezet homogénebb feszültségeloszlást eredményez, míg a kompozitok szerkezete miatt azokban feszültségtorlódások tudnak kialakulni, amik nagyobb energiájú hangkitöréseket okoznak.

A fémeknél az alumínium sokkal kevesebb eseményt produkált, mint az acél. Ennek oka valószínűleg abban áll, hogy az alumínium sokkal képlékenyebben viselkedik az acélnál.

Mivel nem lágy, hanem lemezhengerelt acélról van szó, ezért ennek viselkedése viszonylag rideg, így jellemzőbb rá a repedés, mint a képlékeny alumíniumra.

Érdekes megfigyelni az azonos mátrixú kompozitok viselkedését. A háromféle epoxi kompozit igen eltérő hangokat adott, bár egy hasonló tendencia megfigyelhető volt mindegyiknél. A bazalt erősítésű hangja volt a legmagasabb, a szénszövetesé pedig a legalacsonyabb, az üvegszálás pedig a kettő közt volt található. Megfigyelhető, hogy ugyanez a sorrend az eseményszám tekintetében, a legtöbb eseményt a bazalterősítésű produkálta, utána az üvegszálás, végül pedig a szénszövetes kompozit. Ennek oka a szál/mátrix határfelület minőségével magyarázható. Bazaltszálás kompozit esetében a szál/mátrix együtt dolgozás gyengébb volt, mint üveg vagy szén esetében.

A kétféle üvegszálás anyag hasonló tendenciát mutat a grafikonjaik alapján, pár másodperccel a szakadás előtt sok, hangos esemény történik, az azt megelőző szakaszon azonban csak ritka, és halk hangok hallhatóak. Ezek alapján úgy tűnik, hogy az anyag akusztikus emissziós viselkedését kompozitok esetén elsősorban az erősítőanyag szabja meg, és nem a mátrix.

Minden anyagnál megfigyelhető, hogy a jelentősebb események hasonló frekvenciatartományba esnek, egy 2-3 oktávus sávba. Lehetséges, hogy ennek oka a szenzor érzékenységében is keresendő. Szinte minden anyagnál egy nagy amplitúdójú, sok hangból álló összhangzat jelezte a szakadást, amely általában viszonylag mély is volt. Ennek oka az lehet, hogy a szakadás eseménye valójában egy sor, egymást igen gyorsan követő nagy energiájú elemi eseményből tevődik össze, melyeket a program egyetlen időegységbe tömörít.

6. Összefoglalás

A program teljesíti a kitűzött célt, azaz az akusztikus emissziós mérési eredményeket képes hallhatóvá tenni, miközben minél többet megőriz az eredeti információból. A sokféle beállítás lehetőséget ad további kutatásokban való felhasználásra, a strukturált programszerkezet pedig megkönnyíti a továbbfejlesztést.

A program egy fejlettebb változata képes lehet megkönnyíteni, felgyorsítani az AE vizsgálatokat. Valamint anyag szerkezzettani kutatásokban is használható lehet: a hallás útján új megközelítést kapunk, amellyel az anyagok belső folyamatait feltérképezhetjük. Továbbá oktatási, demonstrációs célokra is alkalmas.

Továbbfejlesztési lehetőségek

A dolgozat témája elsősorban a program létrehozása, a mérési adatok feldolgozása és elemzése csak kisebb súllyal szerepel benne. Logikus lenne tehát újabb méréseket végezni, és a program segítségével kiértékelni azokat. Ismételt mérésekkel megismerhető lenne, hogy mennyire reprodukálhatóak az egyes hangminták, továbbá érdekes lehetne összehasonlítani ugyanolyan anyagok hangját különböző típusú terhelés hatására. Ezen módszerrel megismerhető lenne, mennyire az anyag sajátja a produkált „hang”, és mennyire függ a "gerjesztéstől". Roncsolás mentes vizsgálatoknál is érdekes információkhoz juthatunk, például ütés, vagy rugalmas deformáció hatására létrejövő rezgés elemzésével.

A mérési összeállításon is lehetne javítani, például érzékenyebb műszerrel, valamint jobb akusztikus és elektromos zajmentesítéssel, valamint jelszűréssel is lehetne csökkenteni az esetleges zavaró hatásokat. Több szenzor használatával lehetne a méréseket pontosítani.

A program többféleképpen továbbfejleszthető. Jelenleg a felfutási időt nem használtuk fel a számítások során, az egyenletekbe való „beépítését” is lehetne vizsgálni. A memóriahasználát és a futásidő optimalizálásával javítani lehetne a program működését. Továbbá a jobb futási feltételek lehetővé tennék a valós idejű adatfeldolgozást.

7. Irodalomjegyzék

- [1] Pellionisz P.: Akusztikus emissziós anyag- és szerkezetvizsgálatok. GTE, Budapest (1992)
- [2] Asztalos B.: Akusztikai ismeretek. Kohó- és Gépipari Továbbképző és módszertani Intézet, Budapest (1981)
- [3] Öveges J.: Az élő fizika. Aranyhal Könyvkiadó, Budapest (2000)
- [4] Holroyd J. T.: The acoustic emission and ultrasonic monitoring handbook. Coxmoor Publ., Oxford (2000)
- [5] Rahman J.: Labview. Addison-Wesley, München (2004)
- [6] Kehtarnavaz N.: Digital Signal Processing System Design. Focal Press, Massachusetts (2008)
- [7] Wood A.: The physics of music. Methuen & Co LTD, 36 Essex Street, Strand (1944)
- [8] Atchinson D. A., Smith G.: Optics of the Human Eye. Elsevier Health Sciences, Edinburgh (2000)
- [9] Bishop R. H.: Labview 8 Student Edition. Prentice Hall, New Jersey (2007)
- [10] Hongwei W.: Evaluation of Various Window Functions using Multi Instrument. Virtins Technology, (2009)
- [11] Klimaj G.: Anyagok hangja. TDK dolgozat, BME, Budapest (2001)
- [12] Czigány T.: Az anyag hallhatatlan hangja. Élet és Tudomány, 55, 461-463 (2000)
- [13] Pellionisz P.: Akusztikus emissziós vizsgálat. I. rész: Méréstechnikai alapok. Anyagvizsgálók Lapja, 1, 47-50 (1991)
- [14] Pellionisz P.: Akusztikus emissziós vizsgálat. II. rész: Alkalmazási területek. Anyagvizsgálók Lapja 1, 79-81 (1991)
- [15] Pellionisz P.: Akusztikus emissziós vizsgálat. III. rész: Az AE technika eszközei. Anyagvizsgálók Lapja 2, 19-20 (1992)
- [16] Elliott C., Vijayakumar V., Zink W., Hansen R.: National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement. Journal of the Association for Laboratory Automation, 12, 17-24 (2007)
- [17] Czigány T., Karger-Kocsis J., Ostgathe M.: Damage development in GF/PET composite sheets with different fabric architecture commingled yarn. Journal of Reinforced Plastics and Composites, 17, 250-268 (1998)

- [18] <http://www.muzsika.eoldal.hu/cikkek/zeneelmelet> (2011.08.09.)
- [19] http://fib.bme.hu/fib/cikk/v04_1_teljes/cikk04-1-4.php3 (2011.08.21.)
- [20] <http://zone.ni.com/devzone/cda/tut/p/id/4844> (2011.07.10.)
- [21] http://www.cis.rit.edu/resources/software/sig_manual/windows.html (2011.09.14.)