

Petrovszki András

Konvolúció és teremszimuláció Matlab program segítségével

Szakdolgozat

2008

Széchenyi István Egyetem

Műszaki Tudományi Kar

Jedlik Ányos Gépész-, Informatikai és Villamosmérnöki Intézet

Távközlési Tanszék

Rádiórendszerek szakirány

Győr, 2008.április

Tartalomjegyzék

1. Teremszimuláció – hogyan?	4
1.1. Célkitűzés	5
2. Tapasztalatgyűjtés	6
2.1. A hang terjedése	6
2.2. Valós hangtér, reflektált terjedések	7
2.2.1. A visszhang jelensége	7
2.2.2. Az utózenge idő	8
2.3. Irányhallás, irányinformáció	10
2.4. A teremszimuláció alapgondolata	11
2.5. Valós RIR és spektruma	13
2.6. Tükrérfrekvenciás redundancia a diszkrét mintavétel spektrumában	14
2.7. RIR függvények alkalmazása saját hangforrásunkra	15
2.7.1. A konvolúció	15
2.7.1.1. A konvolúció matematikai alakja	15
2.7.1.2. Adathosszváltozás a konvolúcióban	17
2.7.2. A tartományhatárok átlépése (Fourier)	17
2.7.2.1. Fourier-sor matematikai alakja	17
2.7.2.2. Jelablakolási probléma a spektrumban	19
2.7.2.3. Cirkulációs probléma az FFT-s konvolúciónál	21
2.7.2.4. A mintavételezett jelsorozat kiterjesztésének hatása	22
2.7.2.5. Spektrumvonal-illesztés a gerjesztés és a RIR között	23
2.7.2.6. Gyors Fourier transzformáció (FFT)	23
2.8. RIR függvény felvétele és mérési módszerei	24
2.9. A MATLAB alapjai	25
2.9.1. A programról	25
2.9.2. Grafikus felület létrehozása a MATLAB-ban	27
2.9.3. Alapvető parancsok és írásjelek	28
2.9.4. Komplex számok kezelése	32
2.9.5. Statisztikai adatok a Matlab-ban	32
2.9.6. Időkezelés, időmérés	32
2.9.7. Változók érvényessége - adatforgalom a grafikus felületen	32
2.10. Microsoft „WAV” formátum	34
3. ROOMER funkcióterve, grafikai megjelenése	36
3.1. Fájlkezelés	36
3.1.1. Megnyitás, létrehozásra kijelölés	36
3.1.2. Típusazonosítás, információolvasás	36
3.1.3. Törlés	37
3.2. Hangfájl lejátszása	37
3.3. Hangfájl adatok ábrázolása	37
3.3.1. Hullámforma ábrázolása	37
3.3.2. Spektrumábrázolás	37
3.4. Terem-válaszjel (RIR) felvételi funkció	38
3.4.1. Hangfelvétel készítés	38
3.4.2. Kirajzolás	38
3.4.3. Dirac-generátor	38
3.4.4. Aluláteresztő szűrő modul (DC-10Hz szűrés)	38
3.4.5. Impulzuskereső modul (kezdeti szünet levágása a válaszjelről)	38
3.4.6. RIR normalizálása	39

3.4.7.RIR visszajátszása.....	39
3.5.Teremszimulációs számítási folyamat.....	39
3.5.1.Diszkrét mátrixműveletes konvolúció.....	39
3.5.2.Belső „conv” függvényes konvolúció.....	39
3.5.3.Komplex frekvenciatartománybeli szorzás.....	40
3.5.4.Számításközi százalékos nyomonkövetés.....	40
3.5.5.Diagrammszintű nyomonkövetés.....	41
3.5.6.Megszakítási művelet kezelése.....	41
3.5.7.Amplitúdókövetés, normalizációs szorzó frissítése.....	41
3.5.8.Ideiglenes fájlok kezelése.....	42
3.5.9.Normalizált fájlósszefűzés.....	42
3.5.10.Összesített kijelzés számításakor.....	42
3.5.11.Wav adatkivonási számítás.....	42
4.ROOMER programozási megoldása.....	43
4.1.Fájlkezelés.....	43
4.1.1.Megnyitás, létrehozásra kijelölés.....	43
4.1.2.Típusazonosítás, információolvasás.....	43
4.1.3.Törlés.....	43
4.2.Hangfájl lejátszása.....	43
4.3. Hangfájl adatok ábrázolása.....	44
4.3.1.Hullámforma ábrázolása.....	44
4.3.2.Spektrumábrázolás.....	44
4.3.3.Terem-válaszjel (RIR) felvételi funkció.....	44
4.3.3.1.Hangfelvétel készítés.....	44
4.3.3.2.Kirajzolás.....	44
4.3.3.3.Dirac-generátor.....	44
4.3.3.4.Aluláteresztő szűrő modul (DC-10Hz szűrés).....	44
4.3.3.5.Impulzuskereső modul (kezdeti szünet levágása a válaszjelről).....	45
4.3.3.6.RIR normalizálása.....	45
4.3.3.7.RIR visszajátszása.....	45
4.3.4.Teremszimulációs számítási folyamat.....	45
4.3.4.1.Diszkrét mátrixműveletes konvolúció.....	45
4.3.4.2.Belső „conv” függvényes konvolúció.....	46
4.3.4.3.Komplex frekvenciatartománybeli szorzás.....	46
4.3.4.4.Számításközi százalékos nyomonkövetés.....	47
4.3.4.5.Diagrammszintű nyomonkövetés.....	47
4.3.4.6.Megszakítási művelet kezelése.....	47
4.3.4.7.Amplitúdókövetés, normalizációs szorzó frissítése.....	47
4.3.4.8.Ideiglenes fájlok kezelése.....	47
4.3.4.9.Normalizált fájlósszefűzés.....	48
4.3.4.10.Wav adatkivonási számítás.....	48
5.Program tesztelése, eredmények.....	49
5.1.Mért jellemzők.....	49
5.2.Értékelés.....	51
6.Összefoglalás.....	52
7.Irodalomjegyzék.....	53
8.Ábrák jegyzéke.....	54
9.Táblázatok jegyzéke.....	55
10.Függelék.....	56

1. Teremszimuláció – hogyan?

Talán kezdetnének a téma fejtegetését egy korábban felmerülő kérdéssel is: Miért?

Ezzel kapcsolatban bennem egy néhány évvel ezelőtti tapasztalat hagyott igen mély benyomást. Mindig is szerettem a zenét, annak sokféle formáját. Az elektronika eleinte számomra egy „kényszerű” csináld-magad folyamat mellékterméke, egyfajta „szegény ember zenedoboz” volt, ami minden mozdítható alkatrészt képes volt zenegéppé varázsolni számomra, és amibe végül is beleszerettem. Később egy kis énekkar alakult, amiben mint egyedüli férfi kerültem szólamhoz. Jó baráti kapcsolat alakult ki közöttünk. Egy alkalommal a csapat egy kis részével Ausztriában töltöttünk egy verőfényes nyári hetet, a hazaúton pedig megálltunk a Melki Apátságánál pihenni és nézelődni egy kicsit. Négyen voltunk, pont kitettünk egy kis énekkart, és az egyik múzeumteremben nézelődve egyedül maradtunk. A terem egyenlő szárú kereszt alakban volt kialakítva, középen és a szárakban pedig igen magas üvegfalak mögé zárt apró kiállítási tárgyak vettek körül minket. A helyzet spontaneitása odáig vezetett, hogy egy négyszólamú éneket kezdtünk énekelni, és közben mindegyikünk a terem különböző pontjain sétálgatva mindenféle tárgyakat nézett meg magának, de a figyelmünk hamar az ének hangzására terelődött. Ahogyan az üvegfalak és a tér beszínezte és megzengette az egyébként nem túl erőteljes hangokat, és ahogyan a négy szólam különböző interferenciákba került az énekesek változó helyzetétől függően, mindezek egy különleges varázssá kovácsolódtak a helyszín egyébként is felemelő hangulatával. Sajnos az újonnan belépő nézelődők hamar véget vetettek ennek a fajta élvezetnek, de a benyomás mindenesetre meghatározó lett számomra. Bizonyára érdekes lett volna ugyanezt a dolgot egy nagy barlanghelységben is megfigyelni, illetve nagyobb templomok akusztikája képes az emberi hangot ilyen teltséggel megtölteni.

Persze nem én vagyok az első akit egy hely varázsa, akusztikája arra indít, hogy a fentiekhez hasonló hangzásvilágot reprodukálhassa újra, a nyolcvanas évek digitális forradalma nagy lendületet adott a hasonló irányú fejlesztéseknek.

Ha reprodukálni szeretnénk egy tér hangzásvilágát, több lehetőség is adott a számunkra. A legélethűbb eredményt persze az eredeti hangtér fizikai megvalósítása, vagy a zenekar beköltöztetése jelentheti, ami anyagilag vagy sok esetben fizikailag nem kivitelezhető. Vannak a valós hangtérben olyan folyton változó és jelen lévő paraméterek, amelyeket nagyon nehéz pontosan utánozni vagy szimulálni. Vegyük például a hőmérséklet változásából adódó eltéréseket, vagy a zenészek, énekesek pozíciójának apró változásaiból eredő térinformáció-változásokat, amik egy élő előadás sajátjai maradnak. Egy gyanútlan néző aki épp a mellékhelységbe siet, a teste által elnyelt vagy visszavert hullámok útján szintén más módon befolyásolja a hangteret, mint egy szimulált hangtérben. Még egy nehezen utánozható tényező az infrahangok tartományának visszaadása, ami hangfalon történő kisugárzás szempontjából igen kellemetlen, „teljesítményzabáló” hatású, a dinamikus hangszóró fizikai felépítése miatt

pedig a nagy kitérés kellemetlen modulációkat okozhat a hasznos tartományban, mégis élettanilag ezeknek a hangoknak a hiánya csökkenti a valóságérzetet. Sok esetben mégis kénytelenek vagyunk egy alternatív megoldást alkalmazni.

Másik lehetőségként tehát a hangtér szimulációja áll rendelkezésünkre, vagyis annak fizikai, akusztikai tulajdonságainak meghatározása, és az ezekből az adatokból származtatott szűrő segítségével annak kisugárzása a hangfalakon keresztül. A teremszimuláció megvalósításában két fontos irányvonalat, illetve ezeknek bizonyos keveredését figyelhetjük meg. Az első irányvonal a kor technikai fejlettségére jellemzően az 1960-as évek környékén a szalagos magnók hőskorában alakult ki. A paraméteres reprodukció ezen időben a tér különböző sajátosságainak megfigyelésével, és azoknak bizonyos határok között szabályozható, paraméterezhető alkalmazásával érte el a valósághű hanghatást. Példaként a térben megfigyelhető korai reflexiót egy magnószalagon egymás mellett elhelyezett felvevő és lejátszófej segítségével valósították meg, a szalag sebességével és a fejek távolságával paraméterezhető késleltetéssel. Különböző, főleg mechanikai segédeszközökkel megvalósított megoldások után analóg elektronikus vödörlánc-tárolós megoldások következtek analóg frekvenciaszűrőkkel fűszerezve, majd a digitális technika a 80-as években újabb lendületet adott az ilyen irányú fejlesztéseknek. A jel analóg szűrős feldolgozása helyett digitális hangfeldolgozó DSP egységek kezdtek különböző digitális szűréseket, késleltetéseket és átalakításokat végezni a később analóggá visszaalakított jelen. Ezek a filteres és effektus zengetők még mindig konkrét matematikai függvényekkel leírható szűréseket végeztek, ahol a változó csak a paraméter és a bemenő jel. Itt már sok esetben nem az élethűsége volt a hangsúly, hanem a különböző extrém hangzások elérésében. A matematikai formák igen jó közelítést adhatnak a valós hangtér legfontosabb és legmarkánsabb jelenségeit utánozva, a finomítások lehetősége mégis életre keltette a másik irányvonalat, a konvolúciós zengetőket.

1.1. Célkitűzés

Jelen dolgozatomban célja és témája a konvolúció megvalósításának különböző lehetőségeit összehasonlítani Matlab programkörnyezetben egy konkrét program megírásának segítségével (Roomer), ezáltal áttekintést adni a programkörnyezet működéséről, és megvilágítani az egyes megvalósítási módszerek közötti különbségeket.

A dolgozatban a témák tárgyalása sok esetben nem a megszokott módon történik, vagyis nem követem a pontos alapok ismertetése, majd a végkövetkeztetések definiálása sorrendet, saját tapasztalatomban ugyanis, hogy az olvasót ezzel lehet a leggyorsabban és a legmesszebbmenőkig érdektelenné tenni a témával kapcsolatban. Számomra sokkal szemléletesebb egy bizonyos problémakör megjelölése, és az utólagos problémamegoldás, majd a feltétlenül szükséges matematikai háttér leírása.

2. Tapasztalatgyűjtés

2.1. A hang terjedése

Ha a valós tér hangzásának hű visszaadására készülök, meg kell látnom, hogy egy komplex mechanikai rendszerrel állok szemben, és meg kell vizsgálnom annak tulajdonságait, hogy megérthessem a működését. A hangok a légtér apró nyomásrezgéseiként írhatók le, amelyek a normál légnyomáshoz képest elenyésznek ugyan, mégis életünk egyik legfontosabb tájékoztatói lehetőségét adják. A levegőnek mint anyagnak tömege ($1,184\text{kg/m}^3, 25^\circ\text{C}$) és rugalmas tulajdonsága van, így kiszámítható a benne haladó rezgés terjedési sebessége is.

$$c = 331,5 + 0,6 \cdot T [m/s]$$

Mivel a levegő sűrűsége jócskán hőmérsékletfüggő, ez a hangsebesség képletében is megjelenik. A „T” a hőmérséklet értéke $^\circ\text{C}$ -ban.

Ha a tér egy pontjában megállunk, és az ott fellépő nyomásváltozásokat az időben jellemezni szeretnénk, a hanghullám alapképletét kell elővennünk.

$$P(t) = P_0 + A \cdot \sin(2\pi f t + \varphi)$$

Ahol $P(t)$ a pillanatnyi nyomás, P_0 a légköri nyomás, A a nyomásváltozás csúcsértéke, f a frekvencia, φ pedig a fázishelyzet értéke. Emberi fül számára közvetlenül a 20Hz és 16kHz közötti frekvenciaértékek hallhatóak, bár egyes kutatások szerint a 20 Hz alatti infrahangok is jócskán befolyásolják a környezetünk érzékelését.

A hangnyomás effektív értékéből ($P_{eff} = \sqrt{2} \cdot A$) származtathatjuk az egyik legfontosabb jellemzőt, amellyel egy hangjel erősségét mérhetjük. A hangnyomásszint számításánál sok megfigyelő hallásküszöbét vették alapul, és megállapították, hogy a még észlelhető legkisebb hangnyomás 1 kHz-es frekvencia esetén $P_0 = 20\mu\text{Pa}$. Ebből a hangnyomásszint a következőképpen adódik:

$$P = 20 \log\left(\frac{P_{eff}}{P_0}\right)$$

Ha a tér egy bizonyos pontjában hangot keltünk, és egy másik ponton figyeljük az eredményt, a hangsebességtől és távolságtól függő késleltetést fogunk tapasztalni. Egyutas terjedés esetén a fázishelyzet számunkra még nem bír nagy jelentőséggel.

Másik jellemző mennyiség a hangintenzitás. Ez a mennyiség a nyomásváltozás helyett az egységnyi felületen egységnyi idő alatt átáramlott energiával fejezi ki a hang teljesítménytartalmát. Ez a mennyiség nehezebben mérhető, a részecskesebességből és hangnyomásból számítható.

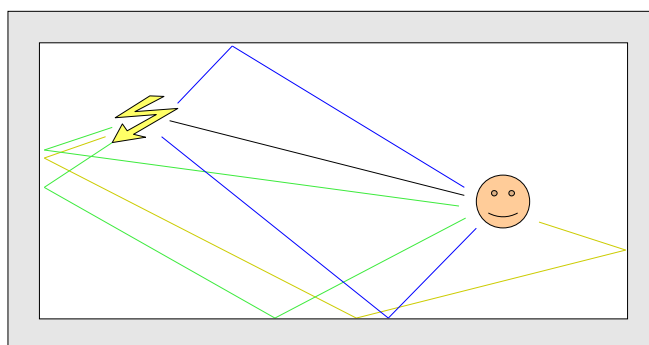
$$I = P_{eff} \cdot v_{eff} [W/m^2]$$

$$I(dB) = 10 \log(I/I_0)$$

$I_0=10^{-2}\text{W/m}^2$ szintén a hallásküszöbnek megfelelő érték. Érdekes megfigyelni az analógiát a részecskesebesség és az áramerősség, a nyomásváltozás és a feszültség, valamint az intenzitás és a teljesítmény között.

2.2. Valós hangtér, reflektált terjedések.

Egy valós hangtérben állva a hang előbb-utóbb akadályba ütközik, és irányt változtat. Egy része az új anyagban tovább terjed, egy része elnyelődik, és ami minket legjobban érdekel, egy része a hullámhossztól függően a fényhez hasonló módon visszaverődik. Ezek a visszavert jelek különböző késleltetéssel szintén megérkeznek a figyelőpontra, és a megtett távolság függvényében különböző fázishelyzetekben összegződnek. Az 1. ábrán fekete vonal jelöli a közvetlen jel útját, kék vonal a korai reflexiók útvonalát. A zöld és sárga vonalak egyre több visszaverődésen átesett jelet mutatnak, és ha megfigyeljük, észrevehetjük, hogy a visszaverődések számát növelve egyre több különböző útvonalat találhatunk egyre nagyobb megtett hullámtávolságokkal. Vegyük figyelembe azt is, hogy a valóságos visszaverődések háromdimenziós térben mennek végbe, ezzel egyre több beérkező jelet szuperponálva a hallható hangra. Mivel minden egyes visszaverődési pont és minden megtett méter adott csillapítást eszközöl a hanghullámon, egyre kisebb amplitúdóval érkeznek ezek a jelek a hallgatóhoz, ugyanakkor a beesések növekvő száma az időben elkeni az eredeti hang alakját.



1. ábra: Többutas terjedés

2.2.1. A visszhang jelensége

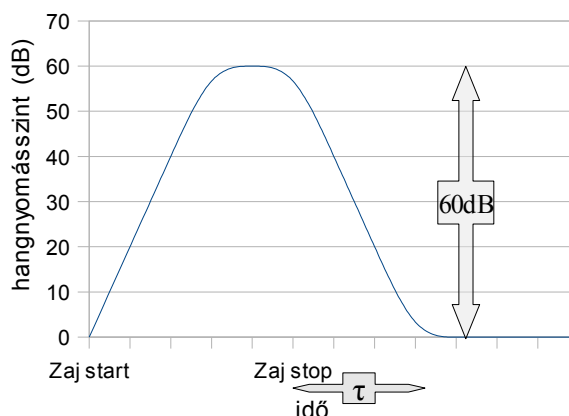
Ha egy hangteret szubjektív vizsgálatnak vetünk alá, vagyis meghallgatjuk a tér hangzását, tehetünk néhány észrevételt. Az első ilyen észrevétel a korai visszaverődésekből származik. Ha a hangterünk kellően nagy, és a visszaverő felületek síknak tekinthetők, jól el tudjuk különíteni az elsődleges terjedést és a viszonylag ép jelalakú visszaverődéseket. Ehhez legalább 50ms késleltetés szükséges két jel között, ilyen esetben visszhangról beszélünk. Sok esetben zavaró a visszhang jelenléte, rontja a beszédhang érthetőségét ahelyett, hogy kellemes dúsulást érezhetnénk a hangzásban. Hosszabb hangoknál, például énekhangnál ez a jelenség nem okoz különösebb problémát. Azokat a visszavert hullámokat, amelyek nem haladják meg az 50ms késleltetést, nem tudjuk a közvetlen hullámtól szabad füllel megkülönböztetni. A fül időbeli elfedése miatt szinte csak hangerőnövekedésnek érzékeljük azt. Érdekes megfigyelni, mikor a hordozható lejátszónkat szabad térbe visszük, annak hangereje sokkal csekélyebbnek tűnik azonos beállítások mellett, mint a szobában volt megfigyelhető.

2.2.2. Az utózengési idő

A másik szubjektíven is megfigyelhető tulajdonság a tér utózengési ideje. A legegyszerűbb szubjektív megfigyelés egy hangtér hangosítási rendszerét hívja segítségül. Ha egy mikrofont kézzel megkocogtunk, a hang hatása több másodpercig hallható maradhat.

Ha egy hangrendszer teljesen zárt lenne, ideális hangvisszaverő felületekkel teljesen körülhatárolt, és a levegő mozgása és sűrűsödése sem vonna el energiát a rendszerből, akkor az ott keletkezett rezgés, hang végtelen ideig megmaradhatna különböző fázishelyzeteket és eloszlásokat, módusokat kialakítva a hangtérben. A megjelenő hang teljesítménye azonban mozgásba hozza a tárgyakat, és azok disszipatív módon túladsnak a beeső energián. Az egyenetlen felületek szétszórják a hullámfrontot és növelik az ütközési felület nagyságát, így nyelve el az energiát.

Az utórezgési idő definíciószerűen is meghatározott. Azt az időtartamot értjük alatta, ami elteltével a gerjesztés nélkül magára hagyott térben a hangnyomásszint 60dB-lel csökken a gerjesztett állapot hangnyomásszintjéhez képest. Az utózengési idő jele a τ (tau).



2. ábra: Utózengési idő mérése zajgenerátorral

Az utózengési idő mérésel történő meghatározására többféle lehetőség adott. Az egyszerűsített eljárásban egy impulzusszerű hangforrást alkalmazunk, és mérjük azt az időt, ami alatt a csúcsértékhez képest a jelszint 60dB alá esik (ezred rész). Ennek a módszernek hátránya, hogy nehéz jó minőségű impulzusjelet előállítani, és a teremhangzás tehetetlensége miatt igen nagy teljesítményt kell bevinni a rendszerbe. Alternatív mérési lehetőség a tér zajgenerátorral történő meghajtása. Ilyenkor hosszabb gerjesztési idő alatt megvárjuk a hangtér telítődését, és a zajgenerátor lekapcsolása után mérhetjük a zengési időt. Az utóbbi megoldás ellenőrzöttebb frekvenciaátvitelt és minőséget biztosít a mérőjel szempontjából is.

Az utózengési idő becslésére különböző számítási módok adottak. Mindegyik módszer kiindulási adatként a térhatároló felületek hangelnyelési tényezőjéből és a hantér térfoga-

tából indul ki. A hangelnyelési tényező az elnyelt energia és a beeső energia hányadosa, jele az α (alfa), mely anyag típusonként változó mennyiség. A hangelnyelési tényező frekvenciafüggő mennyiség, általában a frekvencia növekedésével nő az értéke. A formulák tapasztalati megfigyeléseken alapszanak, ezért igazán pontos eredményt egyiktől sem várhatunk.

Egyik ilyen módszer a Sabine-formula:

$$\tau = 0.161 \frac{V}{\alpha \cdot S}$$

Ahol az S a felület, V a térfogat, α a hangelnyelési tényező. Ez a képlet akkor érvényes, ha a térhatároló felület homogén. Ha ez nem teljesül, akkor a tört nevezőjében lévő kifejezést részoszegekre kell bontani.

$$\tau = 0.161 \frac{V}{\sum \alpha_i \cdot S_i}$$

Jelen számítási mód viszonylag kis utózungési időkben ad jó közelítést.

Amennyiben nagyobb négyzetű illetve párhuzamos falakkal határolt terek hangzását vizsgáljuk, a Norris-Eyring formula használata előnyösebb.

$$\tau = 0.161 \frac{V}{-S \ln(1 - \alpha)}$$

Amennyiben a térhatároló felületek nem azonos anyagból készültek, a nevező itt is felbontható részfelületek összegére. Ekkor kapjuk a Millington-Settle formulát:

$$\tau = 0.161 \frac{V}{-\sum S_i \ln(1 - \alpha_i)}$$

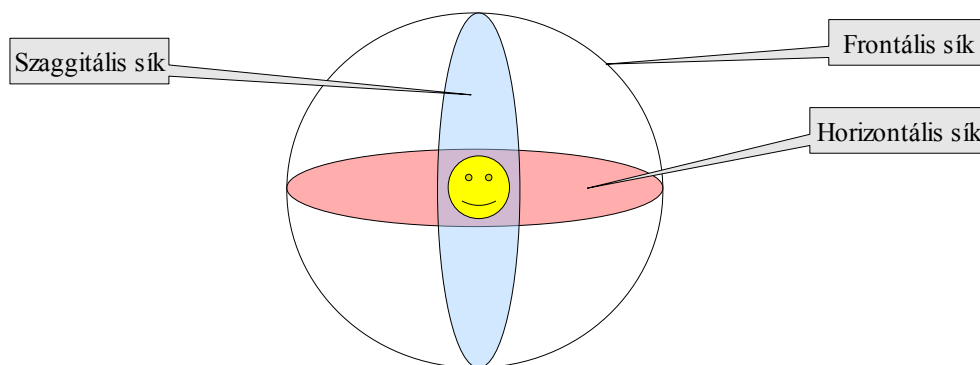
Az utózungési időre jellemző, hogy minél nagyobb utakat tesz meg a hang visszaverődés nélkül, vagyis minél nagyobb a tér, annál nagyobb az értéke. Párhuzamos falakkal határolt térben jellemzően állóhullámok alakulhatnak ki erős leszívásokkal és erősítésekkel, itt az utózungés sokkal inkább pozíciófüggő mint más esetben. A fal egyenetlenségei illetve berendezési tárgyak szintén szétszórják a jelet, ezzel növelve a felületet és a visszaverődések számát.

Néhány tipikus utózungési idő:

- ➔ Ideális süketszoba: 0s
- ➔ Normál lakószoba: 0,3-0,5s
- ➔ Tárgyalóterem: 0,6-1s
- ➔ Mozi : 1-1,3s
- ➔ Operaház : 1,4-2,1s
- ➔ Hangár : 2-6s

2.3. Irányhallás, irányinformáció

Az emberi fül képes a hangforrás önmagához képest vett irányának a közelítőleges meghatározására. Persze az iránymeghatározás az agyban áll össze, és mint ilyen folyamat, tanulható és nem teljes mértékben velünk született tulajdonság. Az agy azonosítja a két fülből érkező jeleket, és a közöttük lévő késleltetés és amplitúdókülönbség segítségével előzetes iránybecslést végez. Az ismert hangok irányának beazonosítása valamivel könnyebb, mert az emlékeztünkben lévő hangok spektrumához képest is tudunk különbségeket észlelni, ez is segít az irány beazonosításában. Ha a hangforrás irányában bizonytalanok vagyunk, egy kis fejmozdítással még pontosabb információt kaphatunk, hiszen a fázishelyzet-változás is sok információt tartalmaz, és nem utolsó sorban a hallásunk nem minden irányban egyformán pontos az iránybecslésben, így ha szembe fordulunk a hangforrással, sokkal jobban be tudjuk határolni a pontos irányt.

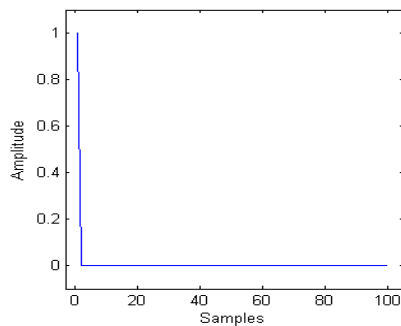


1. rajz: A lokalizáció síkjai

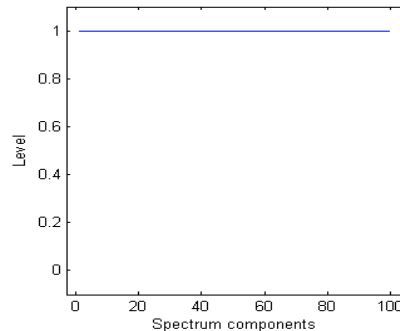
Az emberi irányérzékelés, lokalizáció sajnos igen összetett folyamat, amit nagyon nehéz teljes élethűséggel szimulálni, különböző hangreprodukciós eljárásokkal más és más lokalizációs funkciókat lehet egyre jobban kiszolgálni. Egyik lehetőség például a fejhallgatós rendszerű hangzásvisszaadás. Számítási szempontból ez a legideálisabb megoldás, hiszen két csatorna-adat segítségével egy hangforrás gyakorlatilag bárhova elhelyezhető a térben, viszont a fejmozgásokat nem követik a fázisváltozások, így az agy hamar „rájön a turpisságra”. A másik lehetőség a hangfalas technológia, ami viszont egy pontszerű hangforrás, és csak igen nagy számú hangforrás segítségével lehet élethűen és elég szélesen szimulálni egy virtuálisan más irányból hallható hang hullámfrontját. A hangfalas sugárzási rendszerben a fő információt az amplitúdóviszonyok hordozzák, hiszen a hangfalelrendezés nagyon ritka esetben egyezhet meg pontosan a felvételi pozíciókkal.

2.4. A teremszimuláció alapgondolata

Ahhoz, hogy a terem valódi tulajdonságait szimulálhassam, szükségem van egy olyan csak az adott teremre jellemző információra, amely egyértelműen jellemzi annak az akusztikáját, és könnyen alkalmazhatom tetszőleges hangforráson ezt az információt. Ezt nevezzük Terem-válaszjelnek (Room Impulse Response, továbbiakban: RIR). Az elektronikában hasonló modell megalkotására törekedtek a Bode-diagramm kitalálásakor, és lényegében ezt az analógiát alkalmazhatjuk a valós terek vizsgálatának megértésénél is azzal a különbséggel, hogy a mi esetünkben a „diagramm” felrajzolását nem matematikai műveletekkel végezzük. Esetünkben a matematikai modell leírása a reflexiók igen nagy összetettsége miatt nehézkes, komplex fizikai modellek alkalmazását igényelnék. Ehelyett mintavételt alkalmazunk alkalmas mérőjel és mikrofon segítségével, és a saját diagrammunkat ez alapján „rajzoljuk meg”. A mérőjel ideális esetben az az impulzus lenne, aminek a frekvencia-összetevői azonos teljesítménnyel vannak jelen az egész spektrumban, vagyis a Dirac-impulzus adja a legkönnyebb megoldást a mérésre. A gyakorlatban ennek a jelnek a használata nem minden esetben kivitelezhető, magának a Dirac-jelnek az előállítás is közelítőleges eredménnyel sikerülhet csupán.



4. ábra: Dirac-delta jelalakja 100 mintavétel hosszúságú diszkrét jelben



3. ábra: Dirac-delta frekvencia-összetevői (vektorhossz)

A fenti ábrákon az ideális Dirac-jel spektrumát és jelalakját láthatjuk, és ez egyben szemlélteti is számunkra az időtartomány és a frekvenciatartomány kapcsolatát. A frekvenciatartománybeli átvitelnek fontos részét képező képzetes részt itt most nem ábrázoltam, mert annak értéke jelen esetben nulla minden frekvenciaértéken, de fontos információval szolgál a jel késleltetésében, semmiképp nem elhanyagolható.

Ha egy valós fizikai rendszert egy Dirac-jellel meghajtunk, megkapjuk annak válaszfüggvényét az időtartományban, majd ennek a válaszfüggvénynek a spektrumát meghatározva megkapjuk a frekvenciatartománybeli átviteli függvényt is.

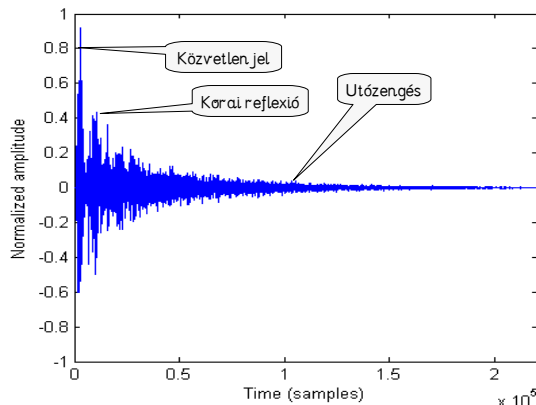
Mindezt akkor tehetjük meg, ha fizikai rendszerünk (hangterünk) passzív elemekből áll, vagyis a válaszjelünk újabb gerjesztés hiányában az időtartományban tekintve konvergál a nulla felé, valamint a tér homogén, lineáris, vagyis két különböző gerjesztőjel együttes elhangzása esetén a válaszjel megegyezik a gerjesztőjelek egyenkénti betáplálása esetén mért válaszjelek összegével. Tudnunk kell továbbá, hogy a kapott eredmény szorosan függ a mérőjel betáplálási pontjától és iránykarakteristikájától is, valamint a mintavételezés pontos

helyétől, és a mintavevő mikrofon iránykarakterisztikájától is. Ez azt jelenti számunkra, hogy ha a hangforrásunk például egy légyzümmögés, nem tudjuk szimulálni annak mozgásával járó hangzásváltozásokat, így a mintavétel még mindig nem képes teljes élethűséggel visszaadni a tér tulajdonágait, mégis jó nagyságrendi közelítést ad az utózengési idők és frekvenciaválasz tekintetében.

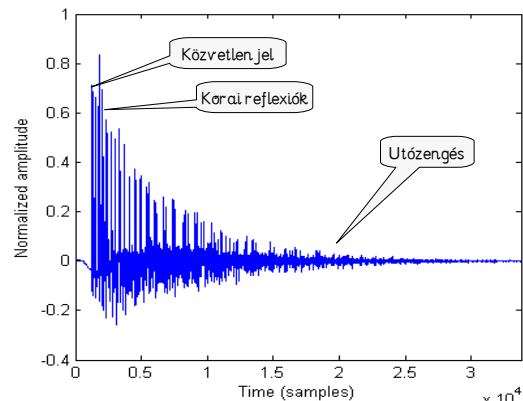
A RIR értelmezéséhez egy másik megközelítést is használhatunk. Mivel digitális jelekkel dolgozunk, amelyek konkrét időpillanatokban konkrét mintaértékeket tartalmaznak, vehetjük az egységimpulzust a lehető legrövidebb ideig tartó maximális értékű mintának. Vehetünk ezután minden soron következő bemenő jelet úgy, mintha ennek az egységimpulzusnak az időben eltoltsága és egy konstanssal megszorított értéke lenne. Ha megvizsgáljuk, hogy az egységimpulzusra a hangtér hogyan reagál, akkor meghatározhatom, hogy annak felére, negyedére hogyan reagál, és időben eltolva egyszerűen egymásra szuperponálhatom az egyes számított válaszjeleket. Ez a művelet lesz az időtartománybeli konvolúció.

2.5. Valós RIR és spektruma

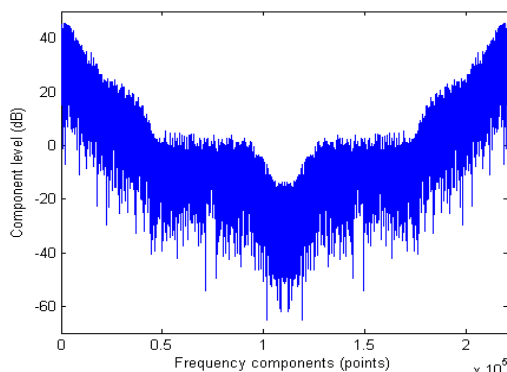
A következő képek két jellemző terem-válaszjel függvényt mutatnak be. Az egyik egy barlangteremben felvett mintát mutat, a másik egy kis templom akusztikájáról készült pillanatfelvétel.



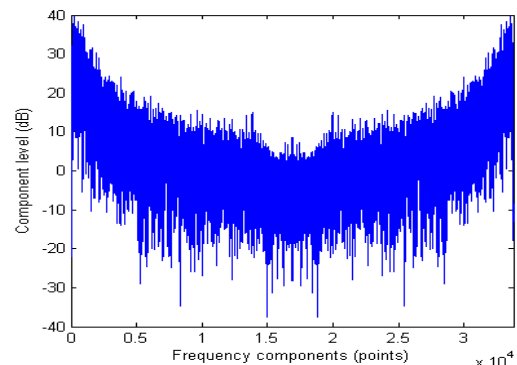
5. ábra: Barlangban felvett RIR függvény



6. ábra: Kisméretű templomban felvett RIR függvény



7. ábra: Barlangi RIR spektruma (vektorhossz)

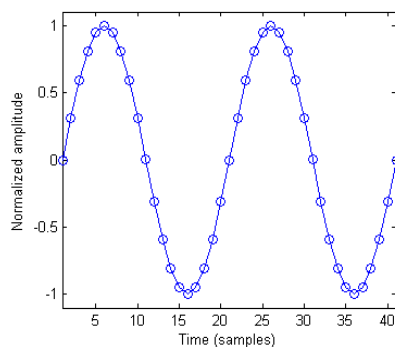


8. ábra: Templomi RIR spektruma (vekh.)

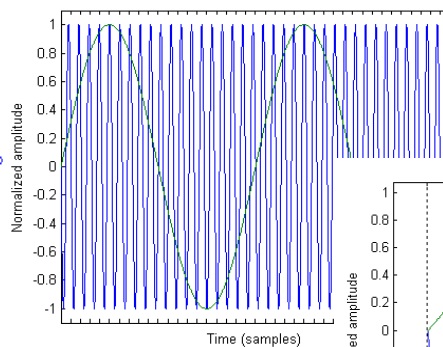
A barlangban készült felvételen jól kivehető a primer hangforrás jele, bár igen erős színezetet kap, valamint jól meghatározható a korai visszaverődés az ezredik mintavétel környékén. A többi visszaverődés az egyenetlen falfelületek miatt nem rajzolódik ki egyértelműen, míg a templomi felvétel Dirac-jele közel tiszta marad, és az egyes visszaverődések élesen elhatárolódva megkülönböztethetők egymástól. A spektrumábrák Matlab környezetben készültek FFT függvény segítségével. Az ábrák a diszkrét (és gyors) fourier transzformáció egy érdekes jelenségét is mutatják, a tükrözött redundanciát...

2.6. Tükörfrekvenciás redundancia a diszkrét mintavétel spektrumában

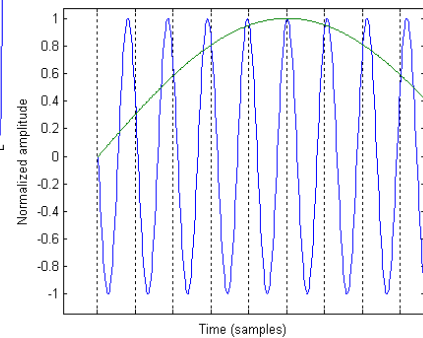
A gyakorlatban használt mintavételezés során gondosan ügyelünk a Shannon-tétel teljesülésére, vagyis a mintavételi frekvencia felénél nagyobb frekvenciájú jeleket nem engedünk a mintavételező bemenetére jutni. A Fourier-transzformáció jól láthatóan nem hallott a Shannon-tételtől, a frekvencia-adatokat egészen a mintavételi frekvencia értékéig szolgáltatja. A mintavételi frekvencia felénél tehát megszűnik a számunkra hasznos információknak a száma, az ezen felüli információk tükörképei (komplex konjugáltjai) az alatta lévő adatoknak. A jelenség egy megközelítését az alábbi ábrákon kísérhetjük figyelemmel.



9. ábra: 2Hz-es jel mintavételi pontokkal



10. ábra: 38Hz és a fedés



11. ábra: Kinagyított részlet

A nyomon követhetőség kedvéért egy 2 Hz frekvenciájú jelet láthatunk 40 Hz mintavételi sebességgel kvantálva. Jól látszik, hogy a 2 Hz frekvenciájú jel mintavételezett értékei a mintavételező jel és a Shannon-tételt megkerülő 38 Hz frekvenciájú jel interferenciájából adódóan megegyezik a 38 Hz frekvenciájú jel mintavételezett értékeivel. A kinagyított ábrán a kvantálási időpillanatot szaggatott függőleges vonal jelzi. Jól látszik, hogy a 2 Hz és az invertált 38 Hz azonos pontokban metszi a kvantálási vonalakat. Ha további felharmonikus jeleket keresnénk, amelyek illenek a fenti kvantálási vonalakra, felfedezhetjük, hogy a diszkrét fourier transzformáció eredménye a frekvenciatartományban periodikus. Amikor a folytonos jeleket kvantálás segítségével diszkrét jelfolyamra cseréltük, a folytonos jelet tulajdonképpen megszoroztuk egy impulzusjellel, így annak spektrumában is megjelenik ez a művelet.

Tudjuk, hogy ezek a „nagyfrekvenciás” összetevők a gyakorlatban nincsenek a bemenőjelben, ezért nem vétünk hibát, ha csak a mintavételi frekvencia fele alatti frekvenciakomponensek értékét rajzoljuk fel a spektrumábránkban, nem utolsó sorban szemléletesebbé téve a valódi spektrumot. Ezután következő spektrumábrákon már használjuk ezt a korrekciót.

2.7. RIR függvények alkalmazása saját hangforrásunkra

A tényleges teremszimulációs számítás tehát két fő tartományban oldható meg, idő és frekvenciatartományban. A megírt programmal a szándékom ennek a két számítási tartománynak a vizsgálata és összehasonlítása. Van néhány tapasztalati információnk előfeltevésünk, és szeretném ezeket megvizsgálni, milyen mértékben állnak meg, milyen minőségi feltételek mellett.

Egyik ilyen feltevés az időtartománnyal kapcsolatos, azt várjuk, hogy egyszerűbb összefüggésekkel jutunk el a végeredményhez, valamint hogy nagyobb pontossággal kapjuk meg az eredményt. Másrészt feltesszük, hogy a frekvenciatartomány a konverziókkal együtt is kisebb számítási időt igényel, valamint a spektrumábrán más szűrők számára is könnyebben feldolgozható adathalmazhoz jutunk.

2.7.1. A konvolúció

Az időtartományban tehát adott egy bemeneti hangjelünk tetszőleges átalakítandó hanganyaggal, és adott egy RIR függvény, amit korábban egy megfelelő mérőjellel mértünk és számítottunk. A bemeneti nyers hangjel tetszőleges hosszúságú lehet, a RIR függvény a gyakorlatban néhány másodperc hosszú, közel exponenciálisan csökkenő amplitúdójú jel. Mindkét jelünk diszkrét mintavételezett értékekből áll, a kettő közötti kapcsolatot a konvolúció fogja számunkra megteremteni.

A konvolúció a szuperpozíciós tételből indul ki, ami szerint a bemenetre adott elemi gerjesztések esetén a kimeneten mért elemi gerjesztésre adott válaszok összege megegyezik a bemenetre összegzetten adott gerjesztés hatásával a kimeneten. Mivel a birtokunkban van egy elemi jelre adott válaszjel, az egymást követő gerjesztésértékek hatásai a kimeneten összeadva, és persze időben eltolva kiszámíthatóak. A 12. ábra szemlélteti a konvolúció működését. Ebből kiolvasható, hogy a RIR függvényünket meg kell szorozni az elemi gerjesztőjel értékével, és hozzá kell adnunk időben eltolva a kimeneti jelünkhöz. Ez elvileg megmutatja a konvolúció számítási igényét is. Minden bemeneti nyers mintára jut tehát N darab szorzás és ugyanennyi összeadás, ahol N a RIR függvény mintáinak a száma. Ha figyelembe vesszük, hogy egy becsületos hangjel másodpercenként 44100 mintát tartalmaz másodpercenként, egy 1 másodperces RIR függvény esetén egy másodperc bemenőjel kiszámításához közel kétmilliárd összeadás és szorzás szükséges. Ez a mai számítástechnika szintjén is elég tetemes számítási időt jelent.

2.7.1.1. A konvolúció matematikai alakja

A konvolúció folyamatos jelek esetében a következő képlettel írható le:

$$c(x) = \int_{-\infty}^{\infty} f(t)w(x-t)dt$$

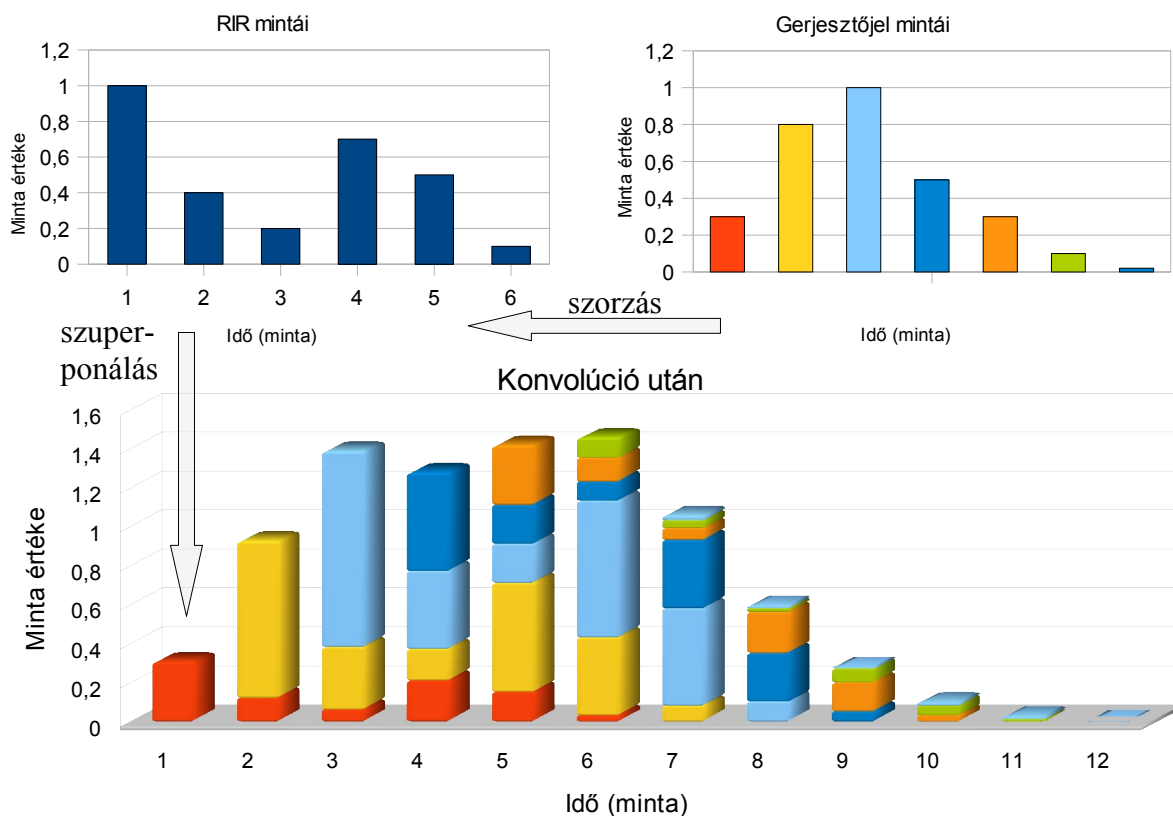
Ebben a képletben $c(x)$ a konvolúció (x) -edik értéke, f esetünkben a súlyozandó függvény, w

pedig a súlyozó függvény. Az f és a w függvények matematikailag felcserélhetőek, a program szempontjából viszont az egyik függvény méretét korlátoznunk kell, ez pedig determinálja a két függvény pozícióját a képletben. A $c(x)$ értékét úgy kapjuk, hogy az $f(t)$ függvényt a $w(x-t)$ súlyokkal megszorozva integráljuk úgy, hogy a t változó szerint „végigtoljuk” a súlyfüggvényt az f függvényen. Mivel mi diszkrét jelekkel dolgozunk, dt számunkra egy konstans mennyiség, $f(t)$ pedig nem folytonos, az integrálás számítása területszámítássá alakulhat számunkra.

A konvolúció elemeinek egy lehetséges kiszámítása a következő képletben látható, ahol $C(x)$ az eredmény x -edik eleme, N_w a súlyfüggvény (RIR) elemeinek száma, $f(x)$ a bemenő adat eleme, $w(x)$ pedig a RIR függvény eleme.

$$C(x) = \frac{1}{N} \sum_{i=0}^{N_w-1} f(x - N_w + i) \cdot w(N_w - i)$$

$1/N$ szorzat az integráció elhagyása miatt kerül a rendszerbe, mi ezt a konstans elhagyjuk, és a számításvégi normalizációval hozzuk a függvényünket amplitúdóban ideális értékre. A Roomer programban nem ezt a folyamatot használok, hanem a bemeneti adatmintát tekintem számítási alapegységnek, és annak a hatását alkalmazom a kimeneti tömb egészére. Így a folyamat egyszerűbben megszakítható a lecsengések megtartása mellett. A Roomer számítási folyamata tehát nem a kimeneti elemeket tekinti egy blokkban kiszámítandó egyégeknek, hanem veszi a bemeneti jel egy értékét, és képezi annak hatását a kimenetre. Ehhez szükséges a kimeneti adatblokk méretének előzetes lefoglalása is. A programban használt folyamat tehát az alábbi ábrán követhető nyomon:



12. ábra: A konvolúció folyamata

Ha megfigyeljük a 12. ábrát, láthatjuk, hogy a jelszintünk jelentősen megnövekedett. A bemeneti jelek amplitúdója a legnagyobb értéken 1 volt, a konvolúció után a legmagasabb jelérték 1,4 felett van. Ez persze részben az $1/N$ konstans elhagyásából adódik, mégis a fizikai jelenség leírásához ez áll közelebb. Több mintából álló jelek esetén az átlapolások száma is megnő, a jelszint 40dB feletti értékkel is nőhet. A digitálisan tárolt adatoknál figyelni kell az eltárolható maximális jelértéket, és megfelelő módszerrel biztosítani kell a megnövekedett amplitúdójú jel torzításmentes tárolását. A Matlab program a megnyitott adatokat 32 biten tárolja akkor is, ha a bemeneti adatok 16 bites kvantálásúak, ezért a számítás közben a jel nem vezérlődhet túl, viszont a kész adat elmentéséhez a jel normalizálására lesz szükség.

2.7.1.2. Adathosszváltozás a konvolúcióban

További megállapítást is tehetünk az ábra alapján az adathosszal kapcsolatban. Egy bemenő minta a RIR függvényen keresztül több kimeneti mintára is hatással van, és hatása saját megjelenésekor fejt ki először a hatását, érdemes megvizsgálni az utolsó bemenő jel hatását. A konvolvált jelben ennek a mintának a hatása még N_w-1 hosszan jelentkezik, ez megnöveli a kimeneti adatsort, egyfajta lecsengést produkál a rendszer. A kimenő adat hossza tehát $M+N_w-1$, ahol M a bemenő adatminták száma, N a RIR függvény hossza.

A konvolúció jele a következő:

$$C = F \times W$$

2.7.2. A tartományhatárok átlépése (Fourier)

Az élet a gyorsabb számítás miatt jóval egyszerűbbnek tűnik frekvenciatartományban, mégis időtartományban létezik. A méréseink csak az időtartományban rögzíthetők, ezért van szükségünk azokra a függvényekre, amelyek az átlépést biztosítják vica-versa a tartományok között.

2.7.2.1. Fourier-sor matematikai alakja

A tartományhatár-átlépést a fourier sorbafejtés, illetve az inverz alakja biztosítja számunkra. Folyamatos jelekre értelmezett sorbafejtés alakja a következő:

$$f(x) \sim \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx)$$

A fenti képlet szerint $f(x)$ függvény felírható egy egyenkomponens, és az alapharmonikus többszöröseinek szinuszos és koszinuszos összetevőiből, ahol az egyes összetevők $a(k)$ és $b(k)$ együtthatókkal lesznek felírva.

Az együtthatók a következő módon számolhatók:

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos kx \, dx \quad (k=0,1,2,3,\dots)$$

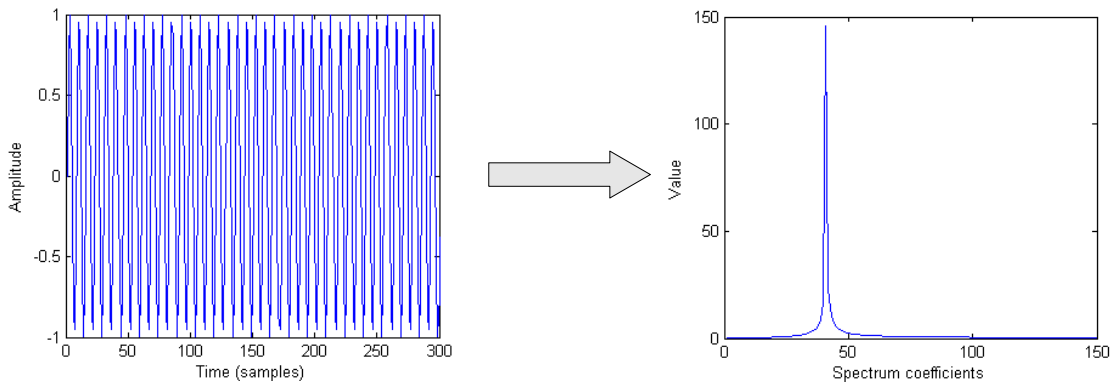
$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin kx \, dx \quad (k=0,1,2,3,\dots)$$

A fourier-transzformációt értelmezhetjük folytonos jelekre, illetve diszkrét jelekre is, számunkra az utóbbi lesz a fontos. Mint ahogyan már fentebb szóba került, a transzformáció időben ismétlődő jelekre értelmezett, ezt pedig ablakolással, kvázi ismétlődő jelekké alakítjuk. Számunkra nem fontos az ablakolási tranziensek csökkentése, ezért négyyszögös ablakot használunk. A diszkrét esetű fourier-transzformáció matematikai formája a következő:

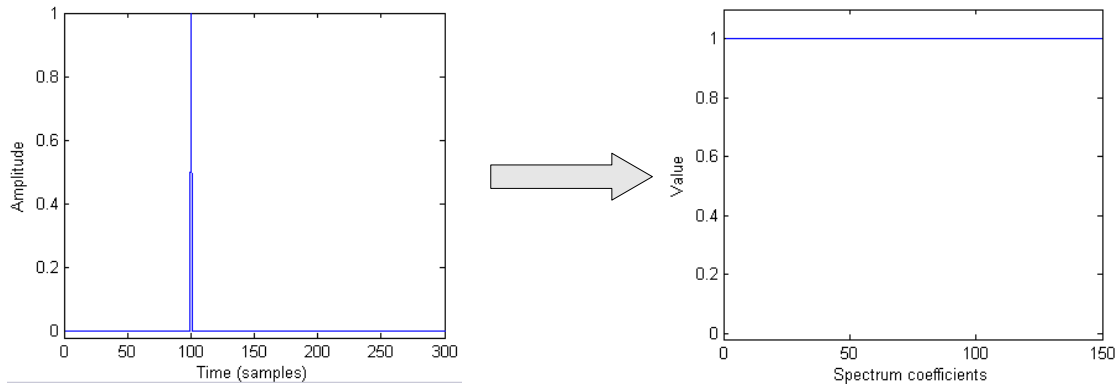
$$F'(x) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) \cdot e^{-j \cdot 2\pi x n / N}$$

Itt $F'(x)$ az x -edik komplex spektrumérték, N a mintavételek száma, az exponenciális szorzó a szinuszos és koszinuszos értéket képviseli komplex trigonometrikus alakban, $f(n)$ a bemenő jel. Az átalakítás redundáns módon jön létre, a mintavételi frekvencia fele fölötti értékeket elhagyva pontosan annyi adatpontot kapunk, mint a transzformáció előtt birtokunkban volt, viszont az inverz függvény kiszámításához, az időtartományba való visszatéréshez a teljes redundáns spektrumra szükség van. A spektrumfelbontásra jellemző, hogy az első frekvenciaösszetevő a diszkrét esetben a DC szintet jellemző érték, a második érték az ablakolt időhossz reciprokából adódó alaphfrekvencia, a többi spektrumérték az alaphfrekvencia egész számú többszöröse. Adódik ebből, hogy a spektrum felbontása, pontossága egyszerre függ a mintavételek számától, ez határozza meg a spektrumvonalak számát, és függ az ablakolási időintervallumtól, ez határozza meg a spektrumvonalak közötti frekvencialéptéket.

A fourier-transzformáció egy érdekes összefüggése, hogy transzformált és visszatranszformált jelek jellegükben egymás inverzei. A vonalas jellegű impulzus időfüggvény a frekvenciatartományban egyenletes, széles spektrumot ad, és fordítva, a vonalas, diszkrét értékű spektrum az időtartományban egyenletes amplitúdójú szinuszos jelet ad.

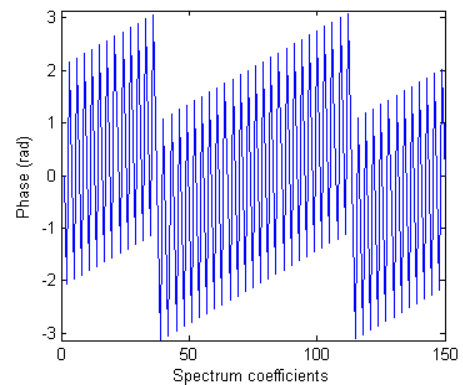


13. ábra: Egyenletes amplitúdó vonalas spektruma



14. ábra: Impulzusjel spektruma

Érdekes továbbá megfigyelni az impulzus késleltetésének hatását a spektrumra. A komplex vektorképben a vektorhossz állandó marad, a fázisinformáció hordozza a késleltetés információját.

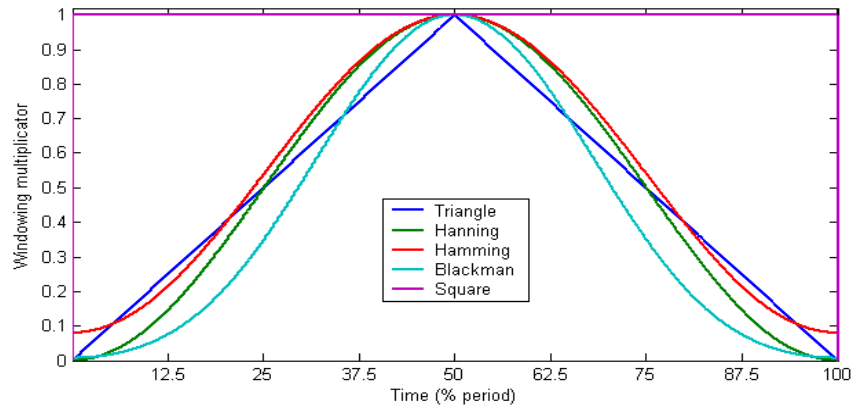


15. ábra: Késleltetés hatása a spektrumfázisra

2.7.2.2. Jelablakolási probléma a spektrumban

Beszélnünk kell a Fourier-transzformáció egy másik jellemző problémájáról, ami a jelanalízis folytán kellemetlenséget okoz a helyes spektrum megrajzolásakor. A Fourier-transzformáció egyik alapfeltétele, hogy a vizsgálandó jel periodikus legyen, és egy egész periódus, vagy annak többszöröse kerüljön vizsgálat alá. Nos, a gyakorlatban ez szinte soha nincs így, ehelyett önkényesen „megerőszakoljuk” a jelfolyamot, azt állítva egy bizonyos kiválasztott darabjáról, hogy ez bizony periodikusan ismétlődik. A transzformáció hibátlanul visszaadja számunkra ezt a jelfolyamot, viszont az ablakolás peremeinél nemkívánt tranzienseket viszünk a rendszerbe. Vegyünk például egy 1 Hz-es szinuszjelet, és gondolatban ablakoljunk belőle másfél másodpercnyi hosszúságú szeletet. Ha ennek a jelnek a spektrumára lennénk kíváncsiak, az FFT függvényre bocsájtva igen érdekes eredményt kapnánk. Ha az általunk ablakolt jelet periodikusnak tekintjük, és egymás után több darabot elhelyezünk ebből, jól látható tranziensek kerülnek a jelformába. Ha a célunk a jel spektrumösszetevőinek pontos ábrázolása, ezek az ablakolási anomáliák igen fals eredményeket okoznak, torzítják a spektrumábrát. Sok tehetséges ember vált híressé a különféle jó és kevésbé jó amplitúdó-ablakolási megoldások és abalkformák feltalálásával, ismertebbek a Hanning, Hamming, Blackman, háromszög ablakolási módok. Ezek az eljárások arra törekszenek, hogy az ablakolt jel be

és kiléptetési pontjaiban az amplitúdót olyan módszerrel csökkentsek a nullára, amely a legkevesebb spektrumtorzítást okozza a rendszerben a jel valódi spektrumához képest. Az ezen ablakolási módokkal kezelt jelek ideális-közeli spektrumábrát adnak, viszont a jel visszaalakítása szempontjából igen károsak. A következő ábra néhány tipikus esetet mutat az ablakolási megoldásokra.



16. ábra: Ablakolási módok időtartományban

Számítás szempontjából a legegyszerűbb a háromszöges számítási mód, a felsoroltak között a Blackmann súlyozás adja a legjobb paramétereket. A súlyozási számításkor N -el jelöljük az ablak összes mintájának a számát, n -el pedig a számolandó súlyozás pozícióját. A számolásnál $n=0$ amikor a mintavételek felénél járunk, vagyis n értéke $-N/2$ és $N/2$ között változik esetünkben. Ilyen feltételek mellett a következő táblázat szerint számolhatjuk a súlyozást.

Ablak-függvény	Kiszámítása	Legnagyobb melléknyaláb dB	Melléknyaláb csökkenési sebessége dB/oktáv	Ekvivalens zaj-sávszélesség	-3dB-es határ	Maximális mintavételi hiba dB
Square	$w_{(n)} = 1$	-13	-6	$1\Delta f$	$0,89\Delta f$	3,92
Háromszög	$w_{(n)} = 1 - \frac{ n }{N/2}$	-27	-12	$1,33\Delta f$	$1,28\Delta f$	1,82
Hanning	$w_{(n)} = \frac{1}{2} - \frac{1}{2} \cos X$	-32	-18	$1,5\Delta f$	$1,44\Delta f$	1,42
Hamming	$w_{(n)} = \frac{27}{50} - \frac{23}{50} \cos X$	-43	-6	$1,36\Delta f$	$1,3\Delta f$	1,78
Blackman	$w_{(n)} = a_0 - a_1 \cos(X) + a_2 \cos(2X)$	-58	-18	$1,73\Delta f$	$1,68\Delta f$	1,1

1. táblázat: Néhány ablakolási függvény számítása és tulajdonságai

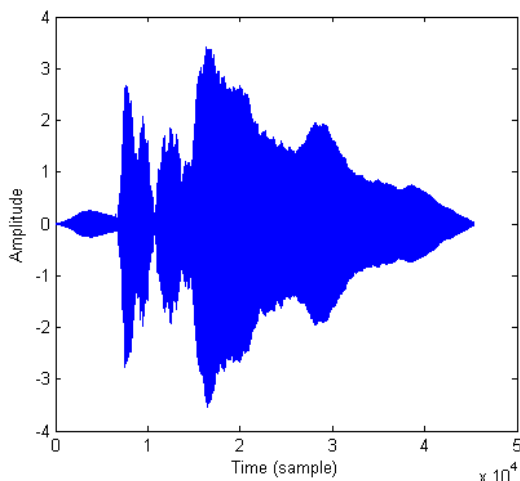
$$X = \frac{2\pi n}{N} \quad a_0 = \frac{3969}{9304} \quad a_1 = \frac{2310}{4652} \quad a_2 = \frac{1530}{18608} \quad -\frac{N}{2} \leq n \leq \frac{N}{2}$$

A táblázat adatai azt az esetet jellemzik, amikor egy adott frekvenciájú jelet táplálunk adott ablakolási mintaszámmal, és változó ablakolási görbéket alkalmazunk. Ideális esetben a spektrumábránk egyetlen diszkrét vonalból kellene hogy álljon, ehelyett azonban különböző anomáliákat látunk a spektrumábrán. A melléknyalábok a valós frekvencia-összetevő mellett megjelenő egyéb összetevők nyalábszerű megjelenése, és jellemző érték, hogy ezek a nyalábok milyen sebességgel csökkennek a hasznos összetevőtől távolodva.

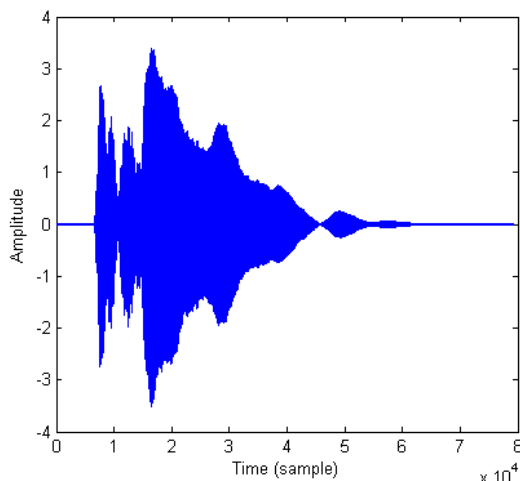
A jelen dolgozatban nagyobb hangsúlyt fektetnek a jó inverzfunkciós visszaalakíthatóságra, ezért az ablakolási módok közül a négyszög-ablakolást választjuk, vagyis nem avatkozunk az amplitúdóba.

2.7.2.3. Cirkulációs probléma az FFT-s konvolúciónál

Amikor a teremszimulációs számítást időtartományban oldottuk meg, találkoztunk azzal a jelenséggel, amikor a konvolvált jel hossza megváltozott a lecsengési folyamatok miatt. Amikor áttérünk időtartományból frekvenciatartományba, teszünk egy feltevést, vagyis hogy a fourier-sorba fejtett jelünk periodikus jel. Amikor a frekvenciatartományban elvégezzük a szorzást, a lecsengési jelek szintén periodikusnak hatnak, a sorbafejtés feltételezi, hogy az aktuális jelre egy korábban lezajlott, azonos periódus lecsengési periódusa hat, és amikor a számított komplex szorzatot inverz transzformációnak vetjük alá, a lecsengési periódust a jel kezdetéhez hozzáadva találjuk meg, mintha azt egy előző periódus okozta volna, a jelen lecsengési periódusunk pedig nem jelentkezik.



17. ábra: Kiterjesztés nélküli FFT-konvolúció



18. ábra: Kiterjesztett FFT konvolúció

A probléma orvosolására a bemeneti ablakolásokat úgy kell kialakítanunk, hogy a visszaalakításkor a periódusnak ne legyen a mintákon kívüli lecsengése, így az algoritmus feltételezheti, hogy a jelen ablakra sem hat az előző periodikus blokk. Tudjuk, hogy a jelenlegi időpillanatban megjelenő bemenő jel még $N-1$ hosszan hat a kimeneti jelre, ahol N a súlyozó függvényünk (RIR) hossza. Kézenfekvő megoldás, hogy az ablakolással kapott időtartománybeli

részfüggvényünket terjesszük ki annyi zero értékű mintával, hogy a blokkon kívülre ne kerülhessen nullától különböző érték. Így az ablakolt jelünket $N-1$ értékkel megtoldjuk az eredeti ablakmérethez képest, és az így keletkező részt nullákkal töltjük fel, ezzel együtt helyet készítünk a lecsengési periódusnak a visszatranszformált jelben.

A fenti ábrákon megfigyelhetjük a kiterjesztés eredményét. A gerjesztőjel első 5000 mintáját nullákkal töltöttem fel, így a konvolúciót követően az első 5000 mintának zérusnak kellene lennie. Láthatjuk a második ábrán a lecsengési folyamatot, ami az első ábrán nem a jel végén, hanem a jel elején jelenik meg, ezzel felborítva a jelünk időrendiségét. Láthatjuk, hogy a kiterjesztett jelünk hosszabb, de a lecsengés jó helyre kerül.

Amikor az ablakolt eredményeket újra összeilleszjük, figyelniünk kell a helyes átlapolásokra. Az eredeti ablakméretnak megfelelően kell összeilleszteni az eredményadatokat, és az egymást fedő adatokat össze kell szuperponálnunk. Így tudjuk kivédeni a cirkulációs ablakolási problémát.

2.7.2.4. A mintavételezett jelsorozat kiterjesztésének hatása

A mintavételezett jelet tehát helykitöltő nullákkal megtoldottuk, és $N-1$ -el több pontot kaptunk az eredeti minták számához képest. Vajon hogyan jelentkezik ez a kapott spektrumképben?

Tudjuk, hogy a transzformáció spektrumvonalai egymástól a jelismétlődés alaphétfrekvenciájának megfelelő távolságban vannak, vagyis $f_0=1/T$. Most tekinthetjük T -t az eredeti mintavételek, az eredeti ablak méretének is, amit $N-1$ -el megtoldottunk, tehát $f_1=1/(T+N-1)$. Ebből adódik, hogy

$$f_1 = f_0 \frac{T}{T+N-1}$$

vagyis az alaphétfrekvencia csökkent, a spektrumvonal-sűrűség pedig nőtt, így a spektrum felbontása nőtt. Ezt tapasztalhatjuk bármely mintabővítés esetén tehát általánosan igaz az összefüggés.

A spektrumvonalak száma a fourier-transzformáció után megegyezik a vett minták számával. Az alapesetben T , utána $T+N-1$ mintánk volt, ezért a spektrumvonalak száma is ezzel megegyező. A növekedés mértéke:

$$M_1 = M_0 \frac{T+N-1}{T}$$

Ezzel együtt persze nőni fog a számítási mennyiség is, tehát a funkció akkor hatásos, ha $T \gg N$, a gyakorlatban ez azt jelenti, hogy az ablakolási méret legalább ötszöröse kell hogy legyen a lecsengési időnek ahhoz, hogy a jelkiterjesztés ne rontsa jelentősen a számítási határfokot. Tudjuk azt is, hogy a lecsengési idő csak a RIR hosszától függ, a legegyszerűbb tehát az ablakolást úgy beállítani, hogy annak mérete kb ötszöröse legyen a RIR hosszának. Nagy méretű RIR függvények esetén ez több másodperc is lehet, és ez megnöveli a program reakcióidejét, tehát az ötszörös szorzót nem teljesítem minden esetben.

2.7.2.5. Spektrumvonal-illesztés a gerjesztés és a RIR között

Amikor időtartományból frekvenciatartományba alakítunk jeleket, fontos figyelni arra, hogy a frekvenciatartományban csak egyforma frekvenciájú komponenseket tudunk egymással összeszorozni, ellenkező esetben torz eredményt kapunk visszatranszformálás után. Ahhoz viszont, hogy a jeleink fourier-sorának felbontása megegyezzen, azonos méretűnek kell lennie a két mintavételezett ablakunknak. Ezt hasonlóan az előző problémához úgy érhetjük el, hogy a kisebbik (arra számítunk, hogy a RIR kisebb mint az ablakolt gerjesztés) adatunkat kell kitoldanunk nullákkal akkora méretre, hogy egyezzen a mintavételi számunk a másik méretével. Érdekes tehát a RIR méretét az ablakolási mérethez igazítani, hiszen a RIR függvény nem változik, egész számítás alatt csak egy alkalommal kell a transzformáltját képezni, vagyis ez nem fogja a számítást jelentősen lassítani. Mivel pedig a RIR-t csupán nullákkal egészítjük ki, annak nem lesz a blokkon túlnyúló hatása, vagyis nem kell további mérettel az ablakokat növelni.

2.7.2.6. Gyors Fourier transzformáció (FFT)

A Fourier sorbafejtés számításigénye csökkenthető, ha a bemeneti adathalmazt részhalmazokra bontjuk. Mivel a DTF N^2 nagyságrendű számítást igényel, két résztömb számítása $2(N/2)^2$ nagyságrendre csökken, melyhez a részzadatok összeillesztése miatt $N/2$ komplex összeadás is tartozik. A felbontást további részekre is ki lehet terjeszteni, az eredmény visszanyerésére lépcsőzetesen van lehetőség. A felbontás matematikai alakja a következő:

$$X_{(k)} = \sum_{n=0}^{N-1} x(n) W_N^{kn}, k=0,1, \dots, N-1$$

$$X_{(k)} = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{mk} + \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{mk}$$

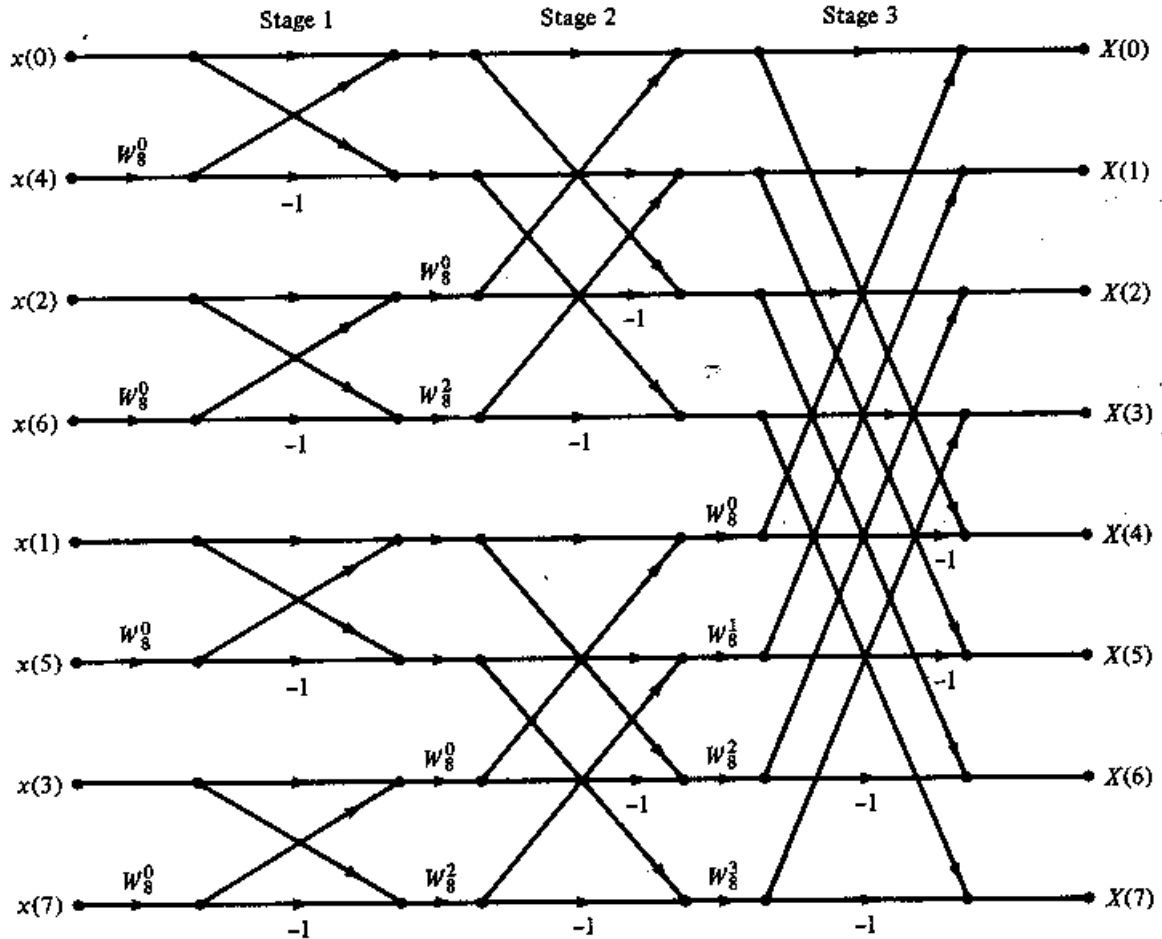
$$X_{(k)} = F_1(k) + W_N^k F_2(k), k=0,1, \dots, N-1$$

$$X_{(k+\frac{N}{2})} = F_1(k) - W_N^k F_2(k), k=0,1, \dots, N-1$$

$F_1(k)$ és $F_2(k)$ az $f_1(m)$ és $f_2(m)$ részhalmazok $N/2$ pontos DFT spektrumai. A számítás kihasználja a DTF redundáns részét a számításra. A számítási mód 4 pontos felbontásra is hasonló módon adott.

A nyolc pontos felbontás esetére hoz számítási folyamatot a 19.ábra. A folyamat 4 egyenlő részre bontott adatfolyam DFT értékeivel dolgozik, rekurzív összeillesztés után kiadja a 8 pontos DFT eredményét. A számítás elemi összeillesztési ábrája miatt pillangó módszernek, azaz butterfly módszernek hívják az eljárást. Az elemi butterfly számítás a következő egyszerűsített módon számolható.

$$x_0 = \frac{1}{2}(y_0 + y_1) \quad x_1 = \frac{\omega^{-1}}{2}(y_0 - y_1)$$



19. ábra: 8-pontos FFT számítási folyamata

2.8. RIR függvény felvétele és mérési módszerei.

A függvény felvételekor fontos a megfelelő eszközök, megfelelő módszerek meghatározása. Bármilyen mérésről is legyen szó, szempont, hogy a hangforrás megfelelően jó átviteli karakterisztikával rendelkezzen, illetve minél egyenletesebb iránykarakterisztikával. A legegyszerűbb mérések legtöbbször mechanikai hangforrásokat használnak, mint például egy kidurrantott luftballon. Ezeknek a módszereknek szépséghibája, hogy nagyon nehezen reprodukálhatóak a mérés pontos körülményei, és a hangforrás minősége nehezen ellenőrizhető. Sok esetben használunk egy ismert hanghullámot ismert átviteli karakterisztikájú hangfalakkal és mikrofonnal, így több mérési eredményt is összehasonlíthatunk a hozzáadódott zajok kiszűrésére. Többfajta mérőjel elterjedt ilyen mérésekre, mint például a szinuszos sweepjel, illetve természetes zene megfelelő spektrumtelítettséggel.

Amennyiben a forráshangunk nem diracjel, szükséges a helyes eredmény elérésének érdekében a mért jel visszaalakítása a mérőjel segítségével. Ez lehetséges időtartományban dekonvolúció, illetve frekvenciatartományban spektrális osztás segítségével. Ezek a műveletek éppen ellentétesek azokkal, ami a teremszimulációhoz szükséges.

2.9. A MATLAB alapjai

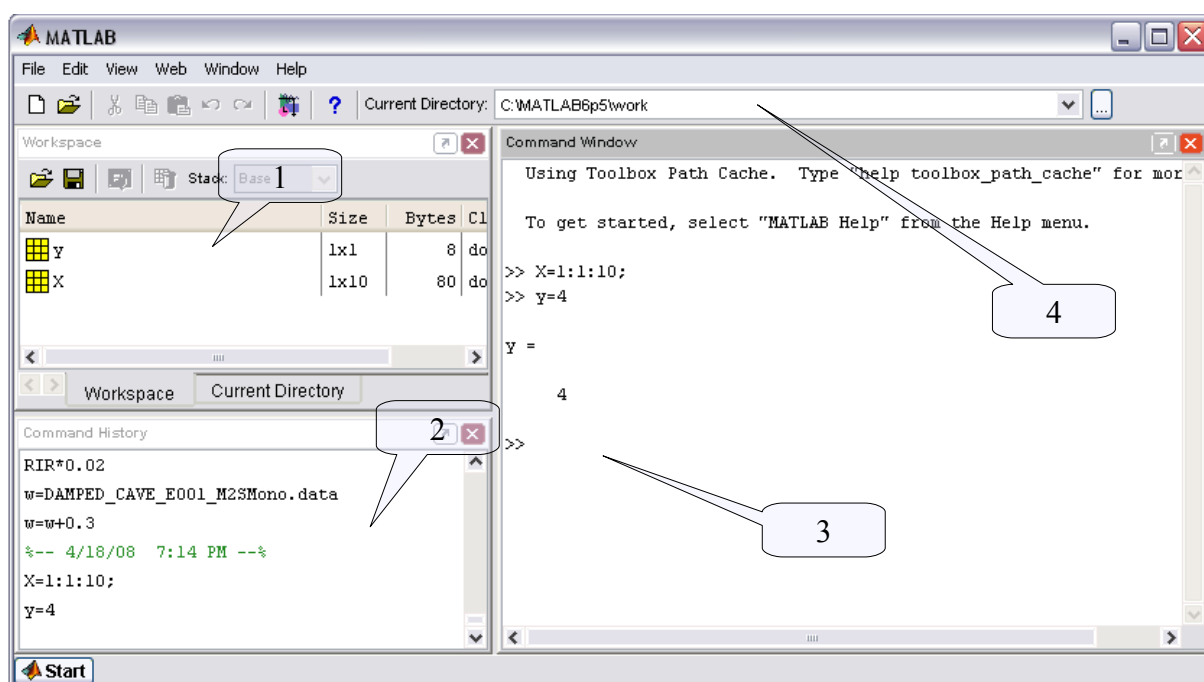
2.9.1. A programról...

A MATLAB programkörnyezet egy önálló, magasszintű programozási nyelv, amely működését tekintve kvázi valósídejű, és programsoros vezérlésű. A nevét a MATrix LABoratory kifejezés mozaikszavaként kapta, és nevéhez hűen kiválóan alkalmas komplex matematikai-mérnöki számítások elvégzésére nagyobb adatmennyiségen is. Fejlesztője az amerikai The MathWorks Inc. A programnyelv viszonylag sok parancsszóból áll, jó struktúráltsága és helyi help-rendszere miatt mégis könnyen kezelhető és átlátható. A Matlab elérhető windows és linux rendszereken is, a megírt funkcióknál szükség esetén érdemes figyelni a hordozhatóság miatti kompatibilitási kritériumokra. A programkörnyezet tartalmaz jó néhány alapfunkciót, és kiegészítő funkciókat. A kiegészítő funkciók az alapfunkciók meghívásával képezhetők, melyeket akár szerkeszthetünk, vagy használatukkal új kiegészítő funkciókat építhetünk a programnyelvbe, így a Matlab kvázi nyitott rendszer, tetszőleges funkciókkal bővíthető. Saját fájltypusai a *.m, illetve *.mex kiterjesztést kapják. Fontos tulajdonsága a programnak, hogy más programnyelvekkel ellentétben alapértelmezetten nem készít függetlenül futtatható fájlokat, mint pl *.exe, a megírt programok forráskódjukban kerülnek mentésre, és újbóli futtatásuk a környezet futtatását igényli. A hagyományos szövegformátumban tárolt parancssorok mentésére a *.m fájl használatos, míg a C vagy C++ formában megadott parancsok a *.mex fájlokban tárolódnak. A programkörnyezet C nyelven íródott, talán ezért is, sok hasonlóság van a MATLAB és a C szintaktikája között.

A MATLAB célja, hogy a változódeklarációkat, illetve tömb és struktúráképzést könnyítve a programozó közvetlen matematikai formulákkal élve érhesse el a kívánt eredményt. A MATLAB-on belül sok hivatkozás, funkció egy az egyben átkerült a C programnyelvből, némi írásmód- és jelölésváltoztatással, néhány funkció pedig szemléletesebb formát öltött, vagy speciális esetekre optimalizálva lett. Ilyen módon van lehetőség a C-ben megszokott fájlmegnyitási műveletet követni, illetve ha egy Microsoft WAV fájl szerelnénk megnyitni, vagy bizonyos képfájl, és rábízunk a formátum kezelését a Matlabra, egy paranccsal rögtön egy adott változótömbbe helyezhetjük az adatsort.

A programnyelv alapértelmezetten valós idejű, vagyis a parancsokat soronként begépelve azok az Enter billentyű lenyomására azonnal végrehajtódnak. Ha komplexebb matematikai műveletet kívánunk kipróbálni, amelynek egymásra épülő felépítése van, a munkakönyvtárban elhelyezett *.m fájl szerkesztésével érdemes ezt megtenni. Erre a célra beépített szerkesztő áll rendelkezésünkre nagyszerű hibakeresési lehetőségeket kínálva számunkra. Az elementett „m” fájl a Matlab parancssorba begépelve rögtön futtatható. A következő ábrán a Matlab főablaka látható a létrehozott változók listájával (1), a kiadott parancsok emlékeztető-

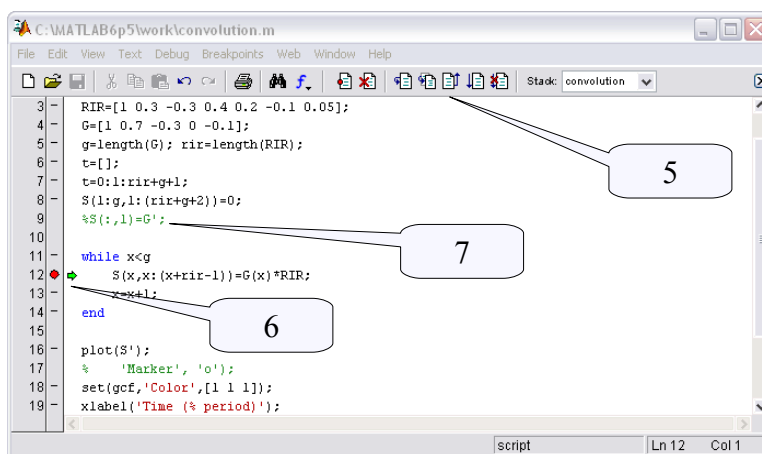
jével (2), illetve a konkrét parancssorral (3), amely egyben a programüzeneteket, változók tartalmát, hibaüzeneteket is megjeleníti szükség esetén. A parancsok tartalmazhatnak egyszerű értékadó műveleteket, egyenleteket, relációkat, függvényeket. A kifejezésben legtöbbször az új értéket felvevő változó áll, majd egyenlőségjel után az értékadó művelet. A művelet végrehajtását az Enter gomb leütése kezdeményezi. A sor végén használatos írásjel a pontosvessző (;) amely után egy sorban akár új értékadó kifejezés is állhat. Amennyiben ezt a jelet használjuk a sor (vagy művelet) végén, a parancssor nem ad visszajelzést az adott művelet végrehajtásáról.



20. ábra: Matlab megjelenése

A MATLAB regisztrált könyvtárszerkezettel dolgozik, amelynek fontos része az aktuális munkakönyvtár is. Az itt található támogatott fájlokat automatikusan felismeri, és alapbeállítás szerint használja a program. A „Current Directory” mezőben (4) a munkakönyvtár beállítható.

Programfejlesztés folyamán fontos funkció a hibakeresés az m-file editorban. A sorok elején beállítható megszakítási pontokat találunk (6), ahol ellenőrizhetjük a változók értékeit, esetlegesen parancssorból műveleteket is végezhetünk a



21. ábra: M-file szerkesztő megjelenése

változókon, majd választhatunk a program megállítása, soronkénti léptetése vagy további megszakítási pontig való futtatása között (5). Lehetőség van ideiglenesen kihagyni bizonyos sorokat a futtatásból, illetve kommentáló szövegeket fűzni a programkódhoz a „%” karakter használatával, az ilyen inaktív sort a szerkesztő zölddel jelöli (7).

A program igen fontos és hasznos funkciója a diagrammos adatmegjelenítés, széles típuskáláról választhatunk, interaktívvá tehetjük illetve tetszőlegesen elforgatott irányból, tetszőlegesen kinagyítva jeleníthetjük meg a kívánt adatokat. A számítási adatokat könnyen elmenthetjük a „save”, vagy más specializált funkció segítségével, illetve a diagrammok nyomtatására is van lehetőség. Képen végzett számítások szintén megjeleníthetők, hangadatok lejátszhatók beépített függvények segítségével.

A változók megadására rugalmas és felhasználóbarát módot találtak a Matlab-ban, ugyanis a változó első használatakor az abba kerülő adatnak megfelelő memóriefoglalás történik meg az értékadáskor, amely foglalás később új adat hozzáadásával bővíthető, illetve parancsok segítségével változtatható, így legtöbbször szükségtelen a változót előre definiálni. A Matlab alapértelmezett változója a mátrix. Gyakorlatilag tetszőleges dimenziójú mátrix létrehozható, és számításokkor tetszőleges részhalmaza kiválasztható. A változók neveire megkötés, hogy betűvel kell kezdődniük, illetve néhány foglalt nevet vagy fix változó nevét (pl.: pi) nem illendő felhasználni. A programsorok érzékenyek a kis és nagybetű különbségére, ezért „a” és „A” két független változó lehet. Lehetőség van struktúrált változók létrehozására is, ahol az egyes szintek ponttal választhatók el. Példa erre a következő

2.9.2. Grafikus felület létrehozása a MATLAB-ban.

A grafikus elemek létrehozására többfajta lehetőség adódik a Matlab-ban. A 10+1 előre definiált formájú grafikai elem mindegyike előcsalható függvények segítségével egy egyszerű „m” kiterjesztésű fájl segítségével, illetve parancssorból is, de lehetőség van a felületet interaktívan, grafikus formában is megtervezni, ekkor viszont az elmentett felület nem csak „m” formátumban, hanem egy „fig” kiterjesztés alatt is megjelenik, amely új kiterjesztés a kezdeti grafikai elemek tulajdonságait tárolja. A grafikus fejlesztőfelület a „guide” parancs kiadásával indítható. Az egyes grafikai elemekhez speciális függvénytársítások tartoznak a mellékelt „m” fájlban, amelyek bizonyos esetekben automatikusan lefutnak. Minden grafikai elemhez tartozik egy ún. „callback” függvény, amely a felhasználó beavatkozása esetén automatikusan lefut, esetleg megvizsgálva egy beírt szám értékét, vagy elindítva bizonyos számításokat. Minden grafikai elemhez tartozik egy függvény, ami az elem keletkezése előtt fut le („createfcn”) egy alkalommal, illetve az ablakkerethez tartozik néhány speciális függvény, amely csak rá jellemző. Az ablakkeret egyedi grafikai elem, a neve „figure”. Több ilyen keret is nyitható, és a további grafikai elemek (10db) hozzá vannak rendelve egy-egy kerethez, amelyiken megjelennek. Az egyes grafikai elemek tulajdonságai (színe, felirat, láthatóság, stb.) bármelyik függvény számára elérhetők és módosíthatók.

2.9.3. Alapvető parancsok és írásjelek

A következő táblázatokban néhány alapvető kifejezést láthatunk, amelyek segítségével a számítások végrehajthatók.

<i>demo</i>	Bemutató programok	<i>help</i>	Súgó meghívása
<i>what</i>	Támogatott fájlok listázása a munkakönyvtárban	<i>type</i>	M-fájl tartalmának (funkció kódjának) kiírása
<i>diary</i>	A munkaszöveg mentése	<i>delete</i>	Munkaállomány törlése
<i>who</i>	Betöltött változók listázása	<i>clear</i>	Munkaterület (vagy változó) törlése
<i>whos</i>	Részletes információ változó(k)ról	<i>save</i>	Munkaterület változóinak mentése
<i>load</i>	Változók betöltése a munkaterületre	<i>length</i>	Vektor (vagy mátrix) hossza
<i>size</i>	Mátrix méretei minden dimenzióra	<i>more</i>	Ablakos lapozás
<i>clc</i>	Parancsképernyő törlése	<i>quit</i>	Program befejezése
<i>startup</i>	M-fájl végrehajtása indításkor	<i>guide</i>	Grafikai fejlesztő indítása

2. táblázat: Fontosabb utasítások

Tömbműveletek		Mátrixműveletek	
+	Összeadás	+	Összeadás
-	Kivonás	-	Kivonás
.*	Elemenkénti szorzás	*	Szorzás
./	Elemenkénti osztás	/	Osztás
.^	Elemenkénti hatványozás	^	Hatványozás
.\	Elemenkénti osztás balról	\	Osztás balról
.'	transzponálás	'	Konjugált transzponálás (illetve dimenziócsere)

3. táblázat: Algebrai alpműveletek

Az egyes utasítások állhatnak önmagukban, illetve némelyek után speciális kapcsolók is állhatnak, amelyek a hatókört szűkítik (2. táblázat). Az algebrai műveletek előtt és után számok, változók vagy függvények állhatnak (3. táblázat). A relációs műveletek igaz-hamis jellegű eredményt adnak, és legtöbb esetben ciklusok vagy feltételes műveletek kimenetelét határozzák meg (4. táblázat).

<	Kisebb	>	Nagyobb
<=	Kisebb vagy egyenlő	>=	Nagyobb vagy egyenlő
==	Egyenlő	~=	Nem egyenlő
&	Logikai ÉS		Logikai VAGY
~	Logikai NEGÁCIÓ	xor	Logikai KIZÁRÓ VAGY

4. táblázat: Relációs műveletek

=	Értékkadás vagy hozzárendelés	.	Tizedes pont
%	Megjegyzés kezdete, sor nem végrehajtható	;	Sor vagy műveletvég-jel
,	Választójel	:	Tömbmanipuláció, műveleti tartomány megadása
[]	Értékmegadás több érték esetén (mátrix sor)	()	Argumentumok, értelmezési tartományok megadása, műveleti zárójel
{ }	Értékmegadás több oszlop esetén (mátrix oszlop)	...	Művelet folytatása a köv. sorban
!	DOS parancs beszúrása	! !	Szövegadat határainak jelölése

5. táblázat: Speciális karakterek

Néhány speciális karakter segít bizonyos műveletek elvégzésében, ezeket a 5. táblázat mutatja. Használatos néhány kitüntetett változó, amelyeknek az értéke adott és bárhol felhasz-

pi	3.14159265	NaN	Nem szám (Not a Number)
ans	Utolsó számítás eredménye	realmin	Legkisebb valós szám
i,j	Imaginárius egység $\sqrt{-1}$	realmax	Legnagyobb valós szám

6. táblázat: Nevezetes változók

Egy speciális karakterről érdemes bővebben is beszélni, mivel nagyon sokoldalú és hasznos része a programnak. A kettőspont több funkcióval is bír. Előállíthatók vele szabályos lépésközü mátrixsorok, és kijelölhető segítségével egy mátrix bizonyos területe számításra. A következő kép tipikus lehetőségeket mutat a kettőspont használatára.

Az első sorban egy egysoros mátrixot képezünk az elemek egyenkénti beírásával, négy oszlopban elhelyezve a négy számot. A program visszajelzi X új tartalmát. Ugyanezt megtehetjük a következő kifejezéssel is, ahol az első szám a kezdőértéket, a második a lépésközt, a harmadik az utolsó elemet jelöli. Látványosabb a módszer előnye, ha az elemeket ötezerig kell megadni. A harmadik sorban kiírjuk X elemeit az elsőtől a másodikig. Mivel nem írtunk értékadási műveletet, az eredményt az „ans” változóban tárolja a program.

```

Command Window
K>> X=[-1,0,1,2]

X =

    -1     0     1     2

K>> X=-1:1:2

X =

    -1     0     1     2

K>> X(1:2)

ans =

    -1     0

K>>

```

22. ábra: Kettőspont funkciói

feval	Függvény végrehajtás	function	Új függvény létrehozása
if	Feltételes végrehajtás (ha...)	elseif	If esetén továbbfűzés (vagy ha...)
else	If használata esetén egyéb eset (egyébként...)	while	Önálló ciklus, feltételes ismétlés (amíg igaz:...)
for	Határozott számú ismétlés (x= tól:ig)	switch	Kapcsoló utasítás kezdete
case	Kapcsoló utasítás esete (ha switch ez, akkor...)	end	If, while, for, switch, try-catch ciklusok vége
pause	Végrehajtás felfüggesztése	break	hurokművelet befejezése
return	Visszatérés a meghívó függvényre	dbstatus	Hibakeresési pontok listázása
try	Első hibajelzésig futtatás	catch	Try testvére, hiba esetén lefutó blokk

7. táblázat: Különleges utasítások, ciklusok

nálható. A fontosabbak az 6.táblázatban megtalálhatók. Viszonylag nagy számú függvény tartozik a különböző szögfüggvények, diagrammtípusok használatához, ezek mind megtalálhatók a Matlab Help-rendszerében pontos szintaktikai leírással és mintapéldákkal, illetve összefoglaló táblázat formájában jelen dolgozat mellékletében is.

ciklikus futású függvényeket, a MATLAB-ban megtalálhatóak ezek a funkciók is (7.táblázat). A „while”, „for” jól kezelhető ciklusfüggvények, és az „if”, „switch” feltételkezelő funkciók a C programnyelvhez hasonló módon működnek.

A ROOMER program megírásához fontos eszközök voltak a különböző fájlkezelő és multimédiás funkciók is. A Microsoft wave fájlt két egyszerű funkció képes írni és olvasni nagyon jól paraméterezhető módon, melyek a fájlban található adatot normalizált értékkel egy megadott változóban tárolják. A beépített funkciók az általam vizsgált 6.5

verzióban sajnos nem működtek minden esetben hibátlanul, bizonyos kiterjesztett információk megléte esetén a beépített funkciók nem működnek. Később a wav-fájl felépítésénél erre még kitérek. A beépített funkciók másik hiányossága, hogy bár képesek az adatot beolvasni vagy kiírni, meglévő adat bővítése sajnos nem lehetséges, ezért a Roomer megírása során egy módosított funkciót kellett készítenem, amely a „wavappend” névre hallgat, és lehetőségessé teszi a darabokban mentett ideiglenes wav

wavread	Microsoft wav fájl beolvasása	wavwrite	Microsoft wav fájl kiírása lemezre
wavplay	Változó mátrix lejátszása alap hangkimeneten	wavrecord	Alap hangbemenetről hangadatot rögzít

8. táblázat: ROOMER-ben használt médiakezelő utasítások

fopen	Tetszőleges fájl megnyitása olvasásra, írásra	fwrite	Írás a megnyitott fájlban
fseek	Pozicionálás a megnyitott fájlban	fread	Olvasás a megnyitott fájlból
cwd	Munkamappa váltása	pwd	Munkamappa kiírása
delete	törlés	fileparts	Fájlnev részeinek szétválasztása
dir	Könyvtárlistázás		

9. táblázat: ROOMER-ben használt fájlkezelő utasítások

(unsigned), ahol a 127. felel meg a nulla értéknek, a negatív csúcs illetve pozitív csúcs a 0 és a 255 értékek. A 16 bites előjeles módban tárolt (signed int), vagyis értéke ± 32767 közötti értékeket vehet fel. A wavread tetszőleges bitfelbontást képes beolvasni, és az adatokat minden esetben -1 és +1 közé normalizálja, ahol a mindenkor kvantálási határ jelképezi a -1 illetve a +1 értékeket. A konvolúció során nagyon fontos tulajdonság, hogy a beolvasott értékek ezután nagyobb kvantálási pontossággal tárolódnak, meghaladhatják a korábbi korlátokat. Újabb kiírás előtt viszont gondoskodni kell arról, hogy az adatok ± 1 között legyenek.

A Matlab számomra legfontosabb három függvénye a „conv”, „fft” és „ifft”. Az első függvény konvolúciót valósít meg a „filter” függvény meghívásával. A „filter” algoritmusát tekintve időtartománybeli differenciálegyenletek z-transzformációs átiratát használja a

számítás gyorsítása érdekében, jelentősen gyorsítva a folyamatot. Az „fft” függvény szabad forráskódú algoritmust használ, amely elérhető a <http://www.fftw.org> internetes címen. Az algoritmus abban az esetben a leghatékonyabb, amikor a transzformálandó pontok száma kettő hatványa, a legrosszabb eredmény pedig primszámok, illetve nagy primszámok szorzataiként adódó számoknál várható. A Roomer program a RIR függvény méretét tekinti majd alapnak, és az algoritmusra bízva a megfelelő fft kombináció megtalálását. Az „ifft” algoritmus megegyezik az fft algoritmusával, különbség csupán előjelben és szorzó értékében van, ezért a sebessége megegyezik az fft sebességével adott számításon belüli adatokkal.

A program grafikai, diagrammok rajzolásával kapcsolatos fontosabb függvényei a következő táblázatban láthatóak. Az esetek nagy részében a „plot” és a „semilogx” függvények elegendőek a ROOMER programban használt ábrák megjelenítésére, az Y tengely logaritmizálását a bemeneti adatok átalakításával célszerűbb megoldani. A két dimenziós ábrázoló függvények az egyes oszlopokat ábrázolják függvényszerűen, amennyiben a kirajzolandó változó több sort is tartalmaz, azokat azonos koordináta-rendszerben, más színjelöléssel rajzolja ki a program.

Külön szót érdemel a felsorolásban a „figure” függvény, amely új ablak nyitására, vagy az aktív ablak kiválasztására is használható attól függően, hogy a megadott pointer létező vagy nem létező ablakot takar.

plot	Lineáris, kétdimenziós adatábrázolás	loglog	X,y tengelyeken logaritmikus léptékű plot
semilogx	X tengelyen logaritmikus plot	semilogy	Y tengelyen logaritmikus plot
bar	oszlopgrafikon	polar	Poláris függvényábrázolás
fplot	függvényábrázolás	stairs	Lépcsőzetes ábrázolás
grid	Méretáló a legutolsó grafikonra	title	Grafikon címkézése
text	Szöveg elhelyezése a grafikonon	gtext	Egérrel pozicionált „text”
xlabel	X-tengely felirata	ylabel	Y-tengely felirata
zlabel	Z-tengely felirata	zoom	Nagyítás-kicsinyítés engedélyezése
plot3	3D rajzolás	contourc	Kontúrozott rajz számítás
contour	2D kontúr rajzolás	contour3	3D kontúr rajzolás
mesh	3D metszetek rajzolása	surf	3D árnyékolt felületek rajzolása
axis	Tengelyadatok kirajzoltatásának határai	axes	Tengelyek kirajzoltatása
cla	Tengely törlése	clf	Rajz törlése
figure	Grafikus ablak létrehozása	colormap	Színtérkép
newplot	Új tengelyek és grafikus képernyő	subplot	Több grafikon egy oldalon

10. táblázat: Gyakoribb diagramm és grafikai ábrázoló függvények

2.9.4. Komplex számok kezelése

A Matlab a komplex számok valós és képzetes részét egyazon változónév alatt tárolja el. A felírás módja:

$C=[1.2+3j];$

A j és i karakterek itt egyenrangúak. Ha két komplex számot szorzok, vektoriális szorzatot kapok, a program figyeli a változó tartalmát. Mégis előfordulhat, hogy egy függvény képzetes részét szeretném ábrázolni, ilyen szelektív esetekben a táblázatban felsorolt függvényeket használhatom.

abs	Komplex érték esetén vektorhossz, egyébként abszolút értéként működik	angle	Vektor fázisszöge radiánban
real	Komplex valós része	imag	Komplex képzetes része

11. táblázat: Komplex adatokat kezelő függvények

2.9.5. Statisztikai adatok a Matlab-ban

A statisztikai függvények közül csak az általam használt függvényeket jelölöm, a lista nem teljes. A max és min értékek egy mátrix elemeinek bizonyos irányban vett maximum-értékeit és minimumértékeit adják plusz és mínusz végtelen irányába.

max	Plusz végtelenhez legközelebbi érték	min	Mínusz végtelenhez a legközelebbi érték
-----	--------------------------------------	-----	---

12. táblázat: Statisztikai függvények

2.9.6. Időkezelés, időmérés

Az idő mérésére egy nagyszerű ötletet alkalmaznak a Matlab-ban. Lehetőség van egy számláló háttérben való futtatására, amelyet a „tic” függvény indít el. A „toc” függvény a számláló aktuális értékét kérdezi le másodpercben, így nyomon követhető a számítási folyamat.

tic	Másodpercalapú stopper indítása	toc	Stopperidő lekérdezése
-----	---------------------------------	-----	------------------------

13. táblázat: Statisztikai függvények

2.9.7. Változók érvényessége - adatforgalom a grafikus felületen

A grafikus felület használatának van egy nagyon érdekes problémája. A változók élettartama sajnos csak abban a funkcióban érvényes, ahol azt létrehozták, minden további funkciómeghívás új és teljesen független munkaterületet nyit. Alapesetben az egyes funkciók számára szükséges adatok bemeneti adatként zárójelben továbbítódnak, az eredmények pedig a funkció visszatérésekor értékadásként jelentkezik olyan módon, mintha a funkció

neve maga a változó lett volna. Egy példa erre:

$D=X/2;$

$S=\sin(D);$

A funkció megkapja a „D” változót, és a számítás elvégzésekor értéket ad „S” számára. Abban az esetben, ha a „sin” funkcióban valamiért kíváncsiak lennénk X értékére, nem tudnánk meg, mivel az nem került átadásra számára. Amikor grafikus felületen dolgozunk, az egyes funkciók közötti adatforgalom elég nehézkessé válhat, ilyen például a megnyitott fájlok nevei. Ezek egyrészt nagy mennyiségű változó átadását igényelnék, másrészt a grafikai felület passzív állapotban nem futtat semmilyen funkciót, így hagyományos értelemben véve nincsenek deklarált változók sem, ahol ezek az információk tárolódnak. A probléma megoldására a Matlab egy különleges tömböt használ, a „handles” struktúrát. Ez a struktúra a grafikai felület megnyitásakor generálódik, és minden grafikai elemről tartalmaz egy ún. „pointer”, aminek segítségével lekérhetjük az elemek állapotát, és meg is változtathatjuk azokat. A pointer nem egy valós adat, nem változó, hanem egy hivatkozási szám a memória egy bizonyos területén tárolt információkra. Az objektumok adatait segédfüggvények segítségével tudjuk elérni, amelyek képesek kezelni és elérni a pointerok által mutatott memóriaterületeket.

A 14. táblázat ilyen funkciókat hoz. A „handles” struktúra tehát tartalmazza a program indításakor létrehozott grafikai elemek pointerait, de lehetőség van felhasználói információk elmentésére is a struktúrában, amelyek elérhetőek minden olyan függvény számára, amelyeket a grafikai objektumok indítanak. Az ilyen függvények számára tehát adott egyrészt a handles struktúra, másrészt pedig a grafikai objektumok információi. Egyik lehetőségünk tehát az adatmentésre a handles struktúra, a másik nagyon érdekes lehetőség pedig az egyes objektumokhoz tartozó felhasználói információ. Ez egy tulajdonság típusú mező, amit szabadon megtölthetünk adatokkal. Néhány helyen nehézkes a handles struktúra használata, itt kézenfekvő megoldás ez az alternatíva. Persze ha az objektum törlésre kerül, a benne tárolt információ is elvész, erre figyelni kell a programozás során.

figure	Új keretet nyit, vagy aktiválja a meglévőt	findobj	Megfelelő grafikai objektum keresése
set	Objektum tulajdonságának módosítása	get	Objektum tulajdonságának lekérdezése
gcf	Aktív keret pointer	gca	Aktív grafikon pointer

14. táblázat: Grafikai elemek kezelésével kapcsolatos függvények

2.10. Microsoft „WAV” formátum

A ROOMER program a hangadatokat Microsoft „WAV” formátumban olvassa és tárolja, ezért érdemes néhány sort fordítani a pontosabb megismerésére.

A Microsoft WAV hangformátum egy igazi matuzsálem a számítástechnikában, bár igen változatos formában tartalmazhat adatot akár tömörített formában is. A Microsoft által kifejlesztett és elterjesztett formátum a kezdetektől fogva kíséri a cég operációs rendszereit mint alapformátum. A hagyományos típus PCM kódolású tömörítetlen hanganyagot tárol hasonlóan az AUDIO CD-hez. A wav lehetőséget nyújt emellett egyéb információk tárolására, mint az előadó, számcím, és hasonló szöveges kísérődokumentumok, bizonyos kódolási módok esetén pedig váltakozó bitrátájú adatblokkok következhetnek egymás után amennyiben a jel jelentős része alacsony amplitúdójú. Az egyes adatblokkokat ún. „chunk”-ban tárolja a formátum, míg a chunk-ok összefoglaló mérete a „header” nevű táblázatban (fejlécben) foglal helyet. Az eredeti wav 32 bites pozícionáló információt tárol, amely 4Gb-ra maximalizálja a fájl méretét, reméljük hogy ezt nem kell átlépnünk a tesztek során. A WAV felépítése egyszerű esetben a következő:

A helyzet és méretjelzés bájtban értendő. Az első három mező a fejlécet takarja, ahol a *RIFF* szöveg ASCII kódja kezdi a sort, majd a *ChunkSize* mezőben az utána következő összes bájtok száma található, a *Format* pedig a *WAVE* szó ASCII kódja. A sárga mező az első adatblokk (SubChunk), amely a következő „data SubChunk” formátumát írja le. Az 'fmt' ASCII kódja jelöli ezt a subchunk-típust. A SubChunk Size mező az utána következő bájtok számát mutatja a Chunk-on belül, jelen esetben értéke 16. Az Audio Format mező PCM esetén 1, más esetben tömörített vagy nemlineáris kvantálású adatformátum következik. Channels Num. Monó esetén 1, Sztereó esetén 2, stb. A Sample Rate a mintavételi frekvencia, pl.:44100. ByteRate a következő szorzatból adódik:

helyzet	név	méret
0	ChunkID (RIFF)	4
4	ChunkSize	4
8	Format (WAVE)	4
12	SubChunkID (fmt)	4
16	SubChunk Size	4
20	Audio Format	2
22	Channels Num.	2
24	SampleRate	4
28	ByteRate	4
32	BlockAlign	2
34	BitsPerSample	2
36	SubChunk2ID (data)	4
40	SubChunk2Size	4
	DATA	SubChunk 2 Size

23. ábra: WAV alapformátuma

$$ByteRate = SampleRate \cdot ChannelsNum \cdot (BitsPerSample / 8)$$

A BlockAlign mező az egy mintához tartozó bájtok számát adja meg az összes csatornán együtt.

$$BlockAlign = NumChannels \cdot (BitsPerSample / 8)$$

A BitPerSample mező az egy csatornához tartozó mintabitek számát adja meg, tipikusan 8 vagy 16.

A Harmadik blokk, amely a konkrét hangadatot tartalmazza, a „data” szó ASCII kódjával kezdődik, majd az adatblokk soron következő hosszát tartalmazó ChunkSize2 mező következik. Ezután már csak a tényleges hangadat áll mintavételi blokkonként, blokkon belül először a bal majd a jobb csatorna adataival.

A WAV kiterjeszthető egyéb információs „Chunk”-okkal, ezeknek a megjelenítését a MATLAB beépített funkciója nem támogatja, és a módosított „wavappend” funkció számára is nehézkes kezelni, hiszen a plusz információ eltárolásáról és újbóli hozzáfűzéséről gondoskodni kellene, ezért a bővített funkció egyáltalán nem támogatja az ilyen fájlokat. A ROOMER-en belül ez nem okoz gondot, hiszen a saját maga által írt fájlokat kell csak összefűzni.

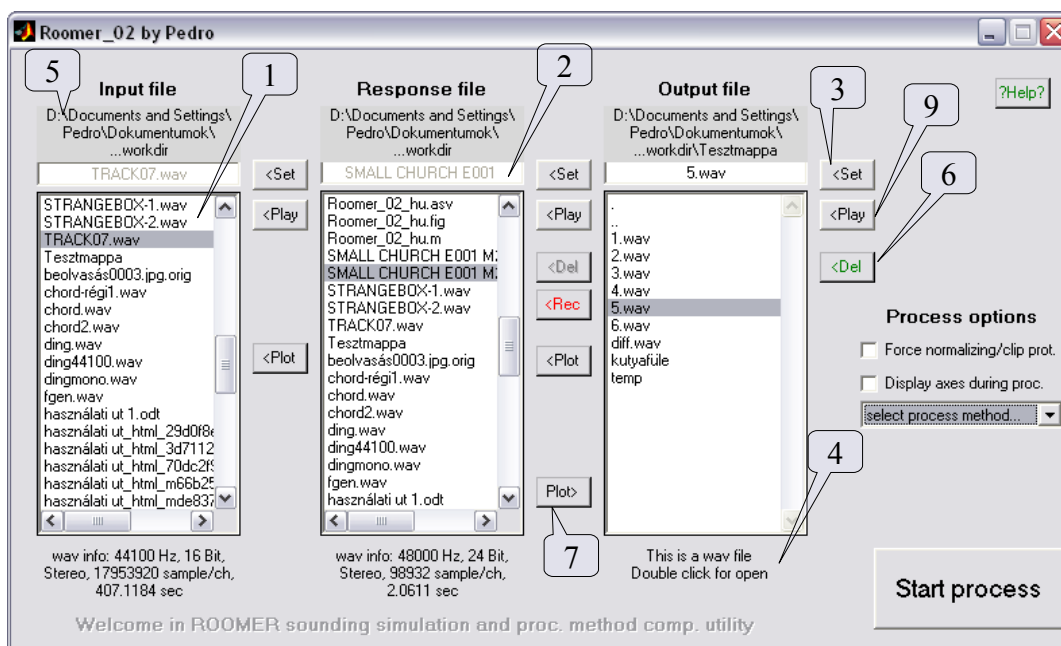
3. ROOMER funkcióterve, grafikai megjelenése

Első lépés a tervezésben a szükséges funkciók meghatározása és a kezelőfelület elrendezése. A fő funkciók és a későbbi grafikai megjelenésük a következő:

3.1. Fájlkezelés

3.1.1. Megnyitás, létrehozásra kijelölés

Ez a funkció ellenőrzi a fájlok létezését, útvonalát, és a megadott adatok alapján eldönti hogy létrehozandó vagy létező adatról van szó, illetve megerősítést kér adat felülírás esetén. Ehhez a funkcióhoz tartozik három fájllista-ablak (1), három gépelőmező (2) és három windows-alapú böngészőablak (3).



24. ábra: ROOMER főablaki megjelenése

3.1.2. Típusazonosítás, információolvasás

Minden megnyitott fájlt vagy könyvtárt azonosítani kell, a nyitott könyvtárakat a fájllista-ablakok felett kijelezni, a megnyitott fájlok típusát azonosítani, illetve megfelelőség esetén a WAV fájlról további tájékoztató információkat kiírni. Ehhez tartozik három könyvtár-felirat (5) és három fájlinfo a lista-ablakok alatt (4), valamint a lejátszás, törlés gombok aktiválása is ide tartozik.

3.1.3. Törlés

Wav fájlok törlése, 2 db törlés gomb tartozik hozzá azokban az ablakokban, ahol a létrehozás is lehetséges (6). A funkció aktív, amikor egy „wav” fájl ki van jelölve a listában, de nincs megnyitva feldolgozásra.

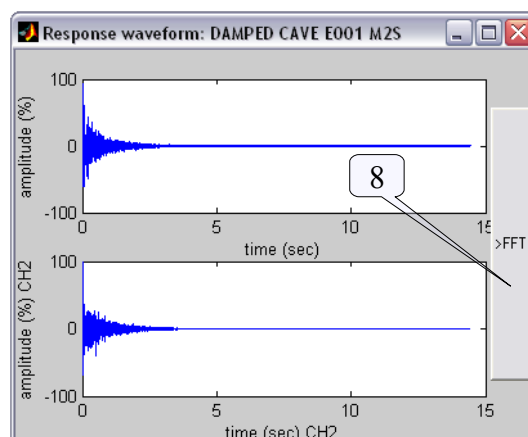
3.2. Hangfájl lejátszása

A lejátszáshoz szükséges a hangadat ablakolása, illetve a használt funkció kezelése. Ehhez a funkcióhoz három „play” gomb kapcsolódik (9). A gomb aktív, ha van kijelölve vagy megnyitva hangfájl a listában.

3.3. Hangfájl adatok ábrázolása

3.3.1. Hullámforma ábrázolása

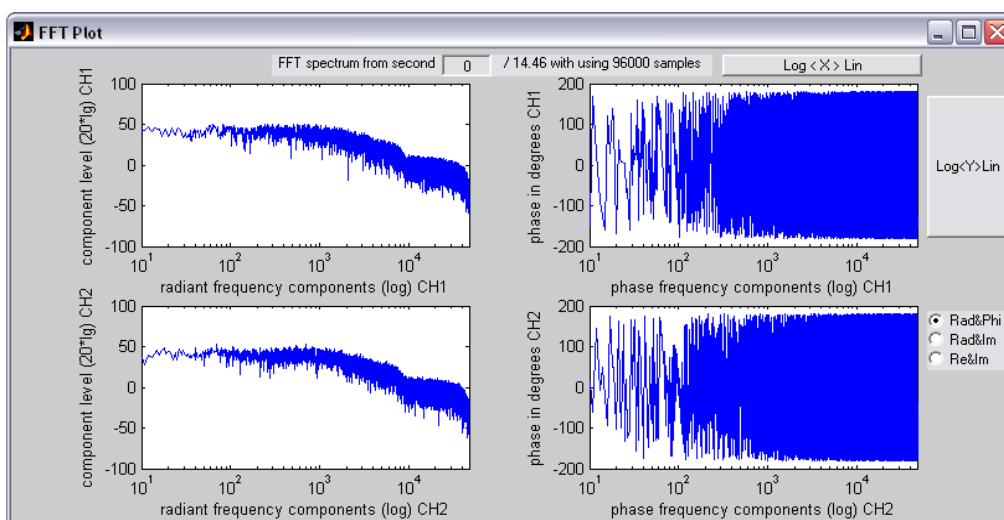
Megnyitott fájlok ábrázolása tartozik hozzá, ahol szükséges az adat ablakolt beolvasása mellett a memóriatarékosság miatt egy egyszerűsített adattömb létrehozása, amely könnyen ábrázolható. 3 db „plot” kapcsoló tartozik hozzá (7).



25. ábra: "plot" funkció megjelenése

3.3.2. Spektrumábrázolás

Ez a plot funkció kiterjesztése. Az adat egy bizonyos részéről, alapesetben egy 1



26. ábra: FFT plot megjelenése

másodperces szeletről készítünk spektrumábrát, és szükséges léptékekkel kijelz-tetjük. A funkcióhoz egy-egy kapcsoló tartozik a plot-ábrák mellett (8). Lehet-séges a spektrumábrákat logaritmikus és lineáris léptékben kirajzoltatni, és a komplex értékek ábrázolhatók vektorhossz és fázisszög, valós és képzetes, vagy kevert módon.

3.4. Terem-válaszjel (RIR) felvételi funkció

A funkció eléréséhez a válaszjelhez tartozó fájllista-ablakot választottam, ahol a létező fájl felülírásával, vagy új fájlnev begépelésével aktiválható „Rec” kapcsoló nyitja meg a hangfelvevő modult.

3.4.1. Hangfelvétel készítés

Időzített felvételek készítése memóri-ába a beállított paraméterek szerint (10). A rögzítési hossz 0,1 és 30 másodperc között lehet.

3.4.2. Kirajzolás

A feldolgozott felvételeadatokat rajzolja ki (11).

3.4.3. Dirac-generátor

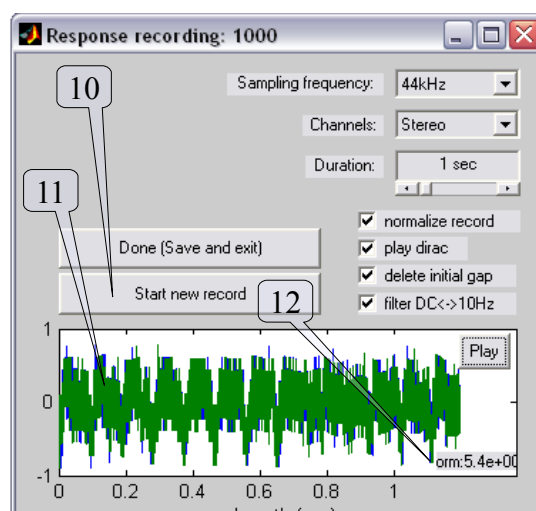
A felvételhez készít és játszik le diracjelet megfelelő időpillanatban. (✓ play dirac)

3.4.4. Aluláteresztő szűrő modul (DC-10Hz szűrés)

Ez a funkció egy fft-iff t kör alkalmazásával kiszűri a 10Hz alatti jelkomponen-seket, ami hasznos egy hibás hangdigitalizáló egység által hozzáadott egyen-szint kiszűrésére is. (✓ filter DC->10Hz)

3.4.5. Impulzuskereső modul (kezdeti szünet levágása a válaszjelről)

A RIR rögzítésekor a felvételt a dirakjel megszólalása előtt kell elindítani, hogy az esetleges indítási zajok, tranziensek a hasznos jel előtt jelenjenek meg. Utólag ezt a biztonsági időszávot érdemes levágni a fájl elejéről. Az első terjedés hangját keresi meg az impulzuskereső modul. (✓ delete initial gap)



27. ábra: Felvételi modul megjelenése

3.4.6. RIR normalizálása

Nem könnyű előre megjósolni a válaszjel amplitúdóját, érdemes lehet normalizálni a felvételt. (√ normalize record) A szorzó megjelenik a diagrammfelületen is (12).

3.4.7. RIR visszajátszása

Gyors ellenőrzési lehetőség a felvétel visszajátszása, és jó eredmény esetén menthető a hang. (Play)

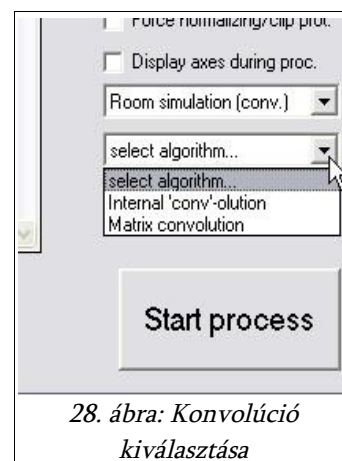
3.5. Teremszimulációs számítási folyamat

3.5.1. Diszkrét mátrixműveletes konvolúció

Elemi matematikai összefüggéseket használ a konvolúció kiszámítására. Algoritmusszintű ablakolást nem hajt végre, néhány műveletenként megszakítás segítségével lehetőséget ad az ábrázoló funkciónak. A funkció kiválasztása után a „Start process” indítja a számítást.

3.5.2. Belső „conv” függvényes konvolúció

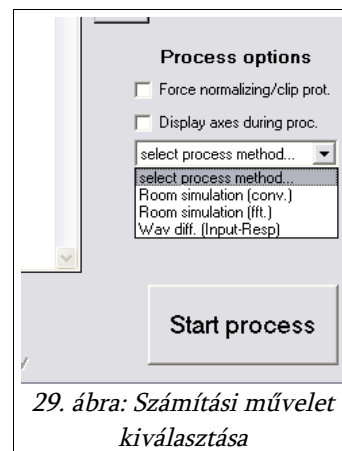
A MATLAB beépített conv függvényét használja a konvolúció kiszámítására. A Z-transzformáció némi sebességnövekedést eredményez, viszont az adatok kiszámítását másodlagos ablakolással végzi. Ablakonként lehetőséget nyújt az ábrázolásra, megszakításra, illetve összefűzi az ablakolt eredmények átlapolt részeit. A funkció kiválasztása után a „Start process” indítja a számítást.



28. ábra: Konvolúció kiválasztása

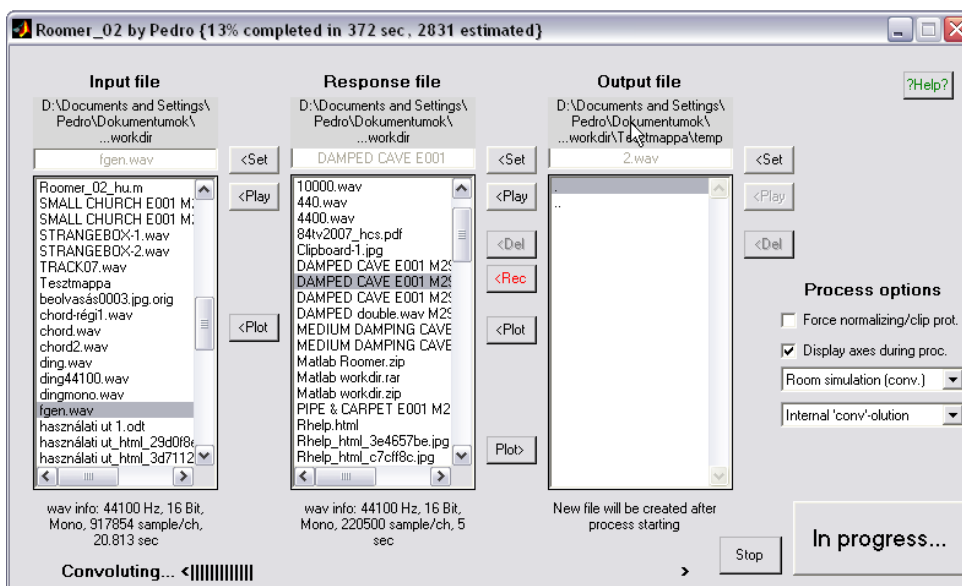
3.5.3. Komplex frekvenciatartománybeli szorzás

FFT és inverz FFT függvényekkel lépünk a frekvenciatartományba és vissza, ott pedig egy komplex számos szorzással redukálódik a művelet. A folyamat ablakolással zajlik, ablakonként ábrázolással, megszakítással, összefűzéssel. A funkció kiválasztása után a „Start process” indítja a számítást.



29. ábra: Számítási művelet kiválasztása

3.5.4. Számításközi százalékos nyomonkövetés

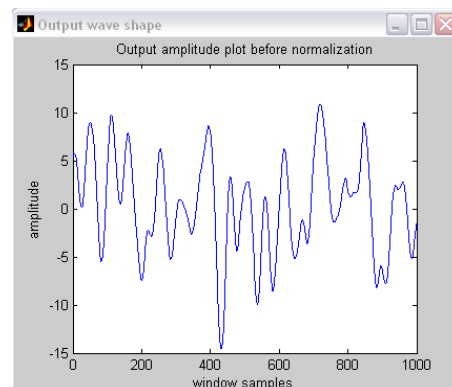


30. ábra: Százalékos nyomonkövetés a fejlécben és grafikusán

A számítás folyamán a befejezett adatmennyiségből kiírja a fejlécben a százalékos előrehaladást, és a becsült várható számítási időt, valamint a program bal alsó részében a futó funkció kiírása mellett jelzi az aktuális funkció készülségi szintjét

3.5.5. Diagrammszintű nyomonkövetés

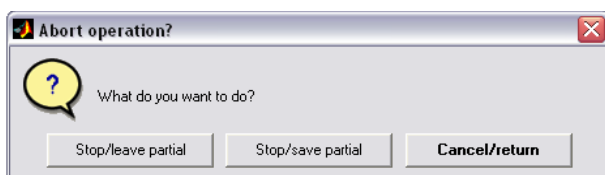
A diagrammos nyomonkövetés betekintést ad a pillanatnyi adatfolyam tulajdonságaiba, kirajzolja az aktuálisan számolt adatokat. A konvolúciós nyomonkövetés egy ábrában hozza a kimeneti jelalakot, míg az FFT nyomonkövetési ablaka a pillanatnyi bemenő és kimenő spektrum-ábrákat is hozza.



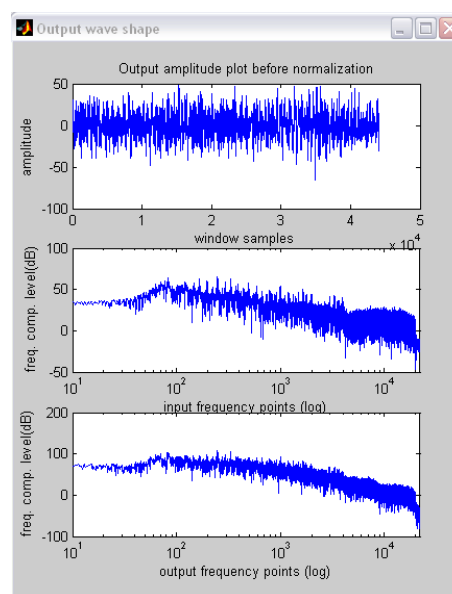
31. ábra: Konvolúció kimeneti jelfolyama

3.5.6. Megszakítási művelet kezelése

A Stop kapcsolón keresztül lehetséges a folyamat megszakítása, amely alkalmas egyszerű szüneteltetésre, megszakításra adattörléssel, és adatmegtartással is. A funkció az adattörléskor felszabadítja a meghajtón elfoglalt területet, megtartáskor pedig átadja a meglévő adatokat a normalizáló funkciónak, ami összefűzi a meglévő ideiglenes fájlokat.



33. ábra: Folyamatmegszakítási ablak



32. ábra: FFT kimeneti jelfolyama és spektrumábrája

3.5.7. Amplitúdókövetés, normalizációs szorzó frissítése

Ez a funkció adatot gyűjt a maximális adatértékekről, és az aktuális ideiglenes fájlt normalizálja az eddigi lehangosabb érték szerinti 100%-ra. Második feladata, hogy regisztrálja az egyes ideiglenes fájlok normalizációs szintjét, hogy összefűzéskor egységes szintre normalizálható legyen a végleges adat. Ez a funkció nem jelenik meg látható módon a grafikus felületen.

3.5.8. Ideiglenes fájlok kezelése

Ez a funkció maximalizálja az egyszerre memóriába olvasott adatok mennyiségét, elsődleges ablakolást hajt végre, a blokkokat lemezre menti ideiglenes wav fájlokban, valamint elvégzi az adatblokkok közötti átlapolt adatok hozzáadását a következő blokk elejéhez. Ez a funkció nem jelenik meg a grafikus felületen.

3.5.9. Normalizált fájlösszefűzés

Az ideiglenes fájlok összefűzését végzi azonos amplitúdóra hozással, amennyiben szükséges a normalizálási funkció is. Néhány másodpercet vesz igénybe a számítási folyamat végén a tároló sebességétől függően. A funkció befejeztével kiírja a végső korrekciós szorzót. A funkció a főablakon bekapcsolható erőltetett módban is, amikor a normalizálás megtörténik olyan esetben is, amikor a jelérték nem haladja meg a kvantálási maximumot. Ez kis jelszintek esetén lehet előnyös a jó dinamika-kihasználásért.

3.5.10. Összesített kijelzés számításakor

A számítás végén összesített hullámforma-kijelzést hajt végre a nyomonkövetési ablakban, valamint záróinformációkat a számítási folyamatról.

Processing finished in 772 seconds. Output gain set by: -36.4526 dB

34. ábra: Processz-infó megjelenése

3.5.11. Wav adatkivonási számítás

Ez a funkció két wav fájl adatainak különbségét képezi, és menti el egy harmadik fájlba. Amennyiben a normalizált különbségi érték nulla, és a normalizálási szorzó „infinity”-t jelöl, kétségtelenül azonos tartalmú fájlokat vontunk ki egymásból. Ezt a funkciót használok fájlok összehasonlítására az eredmény pontosságát illetően. Ez a funkció szintén ablakolt a többi ilyenhez hasonló tulajdonságokkal.

4. ROOMER programozási megoldása

Második lépésként a program megvalósításának folyamatáról és a konkrétabb forráskód-szintű megoldásokról készítünk egy rövid összefoglalót.

4.1. Fájlkezelés

4.1.1. Megnyitás, létrehozásra kijelölés

A program a „handles” struktúrát használja a fájlinformációk tárolására. Az ehhez a funkcióhoz tartozó objektumok mind ezeket az információkat használják fel, vagy töltik fel új értékekkel. A fájllista ablakok a kattintások számától függően frissítik a fájllistát, megváltoztatják a munkakönyvtárat, vagy átadják a fájlnevet típusazonosításra.

4.1.2. Típusazonosítás, információolvasás

A funkció be van ágyazva a megnyitási objektumok függvényeibe, minden azonosításra megkapott fájlnev létezését és kiterjesztését ellenőrzi a program az „exist” és a „fileparts” függvények segítségével. Amennyiben wav kiterjesztést érzékel, de a fájl nem létezik, az oszloptól függően felajánlja új fájl létrehozását, és annak regisztrálását az előző funkcióra bízza. Ha a fájlnev létezik, megpróbál további információt nyerni a „wavread” opciója segítségével, a nyert információt kkiírja a fájllista-ablak alatti mezőben. Figyeli továbbá, hogy be van-e kapcsolva a fájl-betekintés plot funkciója, és szükség esetén meghívja a szükséges kirajzoló funkciót.

4.1.3. Törlés

A funkció a handles struktúrából kiolvassa a fájlnevet és a „delete” parancs segítségével törli azt. A „dir” parancs segítségével frissíti a fájllista-ablakot.

4.2. Hangfájl lejátszása

A handles struktúrában tárolt fájlnev és a hozzá tartozó lejátszási pozíció a kiindulási alap. A program a pozícióhoz képest a memóriába olvassa a handles struktúrában meghatározott méretű adatot, lejátszsa azt, és az új pozíciót menti.

4.3. Hangfájladatok ábrázolása

4.3.1. Hullámforma ábrázolása

Ez egy több helyről is meghívható funkció. Fájlnév alapján megnyitja az adatot, majd ablakolt rendszerben beolvassa az adatokat, közben egy előre lefoglalt kisméretű adatbufferben tárolja a tömörített jelalakot, blokkonként kirajzolja a már beolvasott adatot. Érzékeli a csatornák számát, és szükség szerint felosztja két részre a diagramterületet.

4.3.2. Spektrumábrázolás

A plot funkció fájlnévadatát felhasználva megnyitja a fájlt, és a kezdőpozíciótól kezdve beolvassa a fájlt egy másodperces korlátig. Ebből készít fft segítségével spektrumképet, és a kirajzoltatást a beállítások alapján elvégzi. A kedőpont módosítására is van lehetőség, ekkor az egész funkció újra lefut az új beállításokkal.

4.3.3. Terem-válaszjel (RIR) felvételi funkció

A funkció a válaszjel fájllista-ablaka által hívható meg, illetve a „Rec” funkció segítségével.

4.3.3.1. Hangfelvétel készítés

A beállítások a grafikai objektumokban tárolódnak, ezek alapján indul a beépített „wavrecord” funkció, amely a memóriába rögzít.

4.3.3.2. Kirajzolás

A feldolgozott felvételadatokat rajzolja ki a plot függvény segítségével.

4.3.3.3. Dirac-generátor

Zéró adatblokkot készít, melynek első elemét egyes értékre állítja. És opcionálisan lejátssza a kimeneten.

4.3.3.4. Aluláteresztő szűrő modul (DC-10Hz szűrés)

FFT-nullázás-IFFT funkciót lát el a megfelelő komponensek nullázásával.

4.3.3.5. Impulzuskereső modul (kezdeti szünet levágása a válaszeljéről)

Az algoritmus egymás után két darab 200 mintás ablakból abszolút értékes átlagot képez, és a két ablak átlagát összehasonlítja. Ha ez az összehasonlítás bizonyos értékkülönbséget mutat, a két átlagolt ablak határát tekinti a program a RIR függvény kezdetének, ellenkező esetben egy mintával későbbi adatpozícióval újra átlagolást végez.

4.3.3.6. RIR normalizálása

A max és abs függvények segítségével maximum amplitúdót keresünk, és ezzel osztjuk a felvételi jelalakot. A normalizált érték megjelenik a grafikonon.

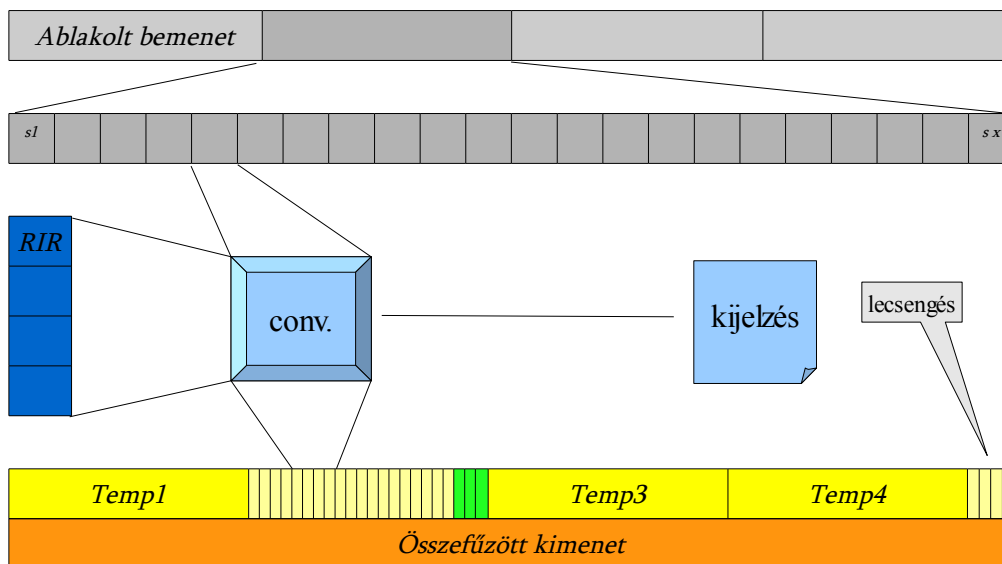
4.3.3.7. RIR visszajátszása

Wavplay funkció segítségével lejátszsa a tárolt adatot.

4.3.4. Teremszimulációs számítási folyamat

4.3.4.1. Diszkrét mátrixműveletes konvolúció

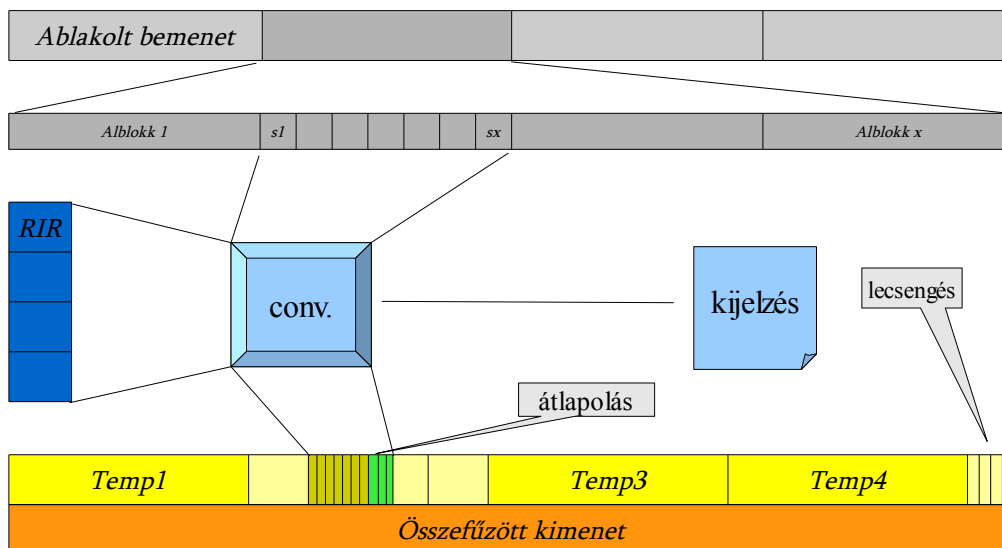
A funkció egyszintű memóriaablakolással dolgozik, valamint bemenő értékhez rendelt számítást végez a kimeneti adatblokkon. Bizonyos számítás szám után a kijelzésnek és megszakításnak adja a folyamatot.



35. ábra: Mátrixkonvolúció adatszerkezése

4.3.4.2. Belső „conv” függvényes konvolúció

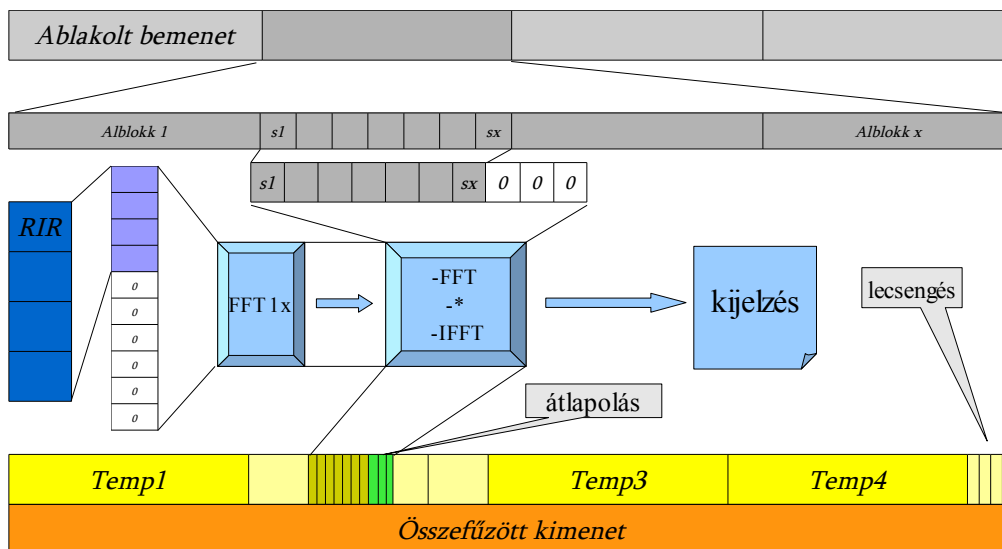
A belső „conv” függvény adatblokkokkal dolgozik, ezért másodlagos ablakolás szükséges.



36. ábra: Belső konvolúció adatszerkezése

4.3.4.3. Komplex frekvenciatartománybeli szorzás

A legbonyolultabb és leggyorsabb algoritmusnál az előző ablakolás teljesen megmarad, viszont az egyes blokkokat további kiterjesztéseknek kell alávetni annak érdekében, hogy az FFT-IFFT lánc a konvolúciós motorhoz hasonlóan működjön. A bemeneti adatblokkok N-1 darab 0 értékű mintával vannak kiegészít-



37. ábra: FFT-IFFT motoros konvolúció adatszerkezése

ve, ahol N a RIR hossza, RIR pedig $X-1$ nullával van kiegészítve, ahol X a bemeneti adatblokk kibővítés előtti hossza.

4.3.4.4. Számításközi százalékos nyomonkövetés

A tic függvény segítségével egy stoppert indít, és minden blokkmegszakításkor új becslést végez az elvégzett számítási adatmennyiség alapján.

4.3.4.5. Diagrammszintű nyomonkövetés

Egyszerű plot és semilogx függvényekkel kirajzoltatjuk a blokkok adatait, ahol erre lehetőség van, spektrumábrát is. Az adatpontok vektorhosszakát ábrázolnak a spektrumábrán.

4.3.4.6. Megszakítási művelet kezelése

A megszakítási művelet három opcióra bomlik.

- visszaadja a vezérlést a számításnak.

- kilép a számítási algoritmusból, és az elkészült adatokat átadja a normalizációs algoritmusnak

- törli az ideiglenes fájlokat, kilép a teljes számítási funkcióból és visszaállítja az alapértelmezett grafikai beállításokat a főablakon.

4.3.4.7. Amplitúdókövetés, normalizációs szorzó frissítése

Minden blokk számítása közben a legnagyobb jelszint értékét figyeli, és amennyiben ezt egy jel túllépi, módosítja a tárolt értéket. A maximális érték reciprokát tárolja a normalizációs szorzóban.

4.3.4.8. Ideiglenes fájlok kezelése

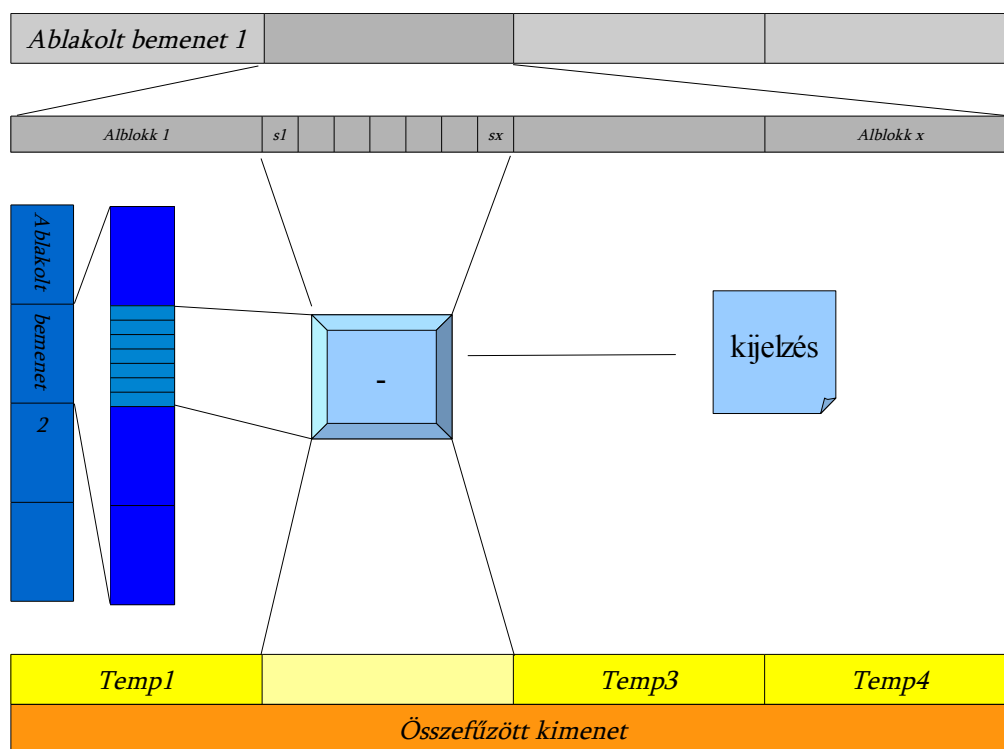
Figyeli az elsődleges blokkon belül a másodlagos blokk pozícióját, és amennyiben az eléri az elsődleges blokk végét, elmenti a lecsengési periódust, lemezre írja a wavwrite függvény segítségével az ideiglenes fájlt az aktuális normalizációs szorzóval súlyozva, létrehoz egy maximum 1000 elemet tartalmazó regisztrációs blokkot, ahol a fájlhoz tartozó normalizációs értéket eltárolja. Wavread segítségével új adattal frissíti a bemeneti buffert, a kimeneti buffert lenullázza, és a lecsengési periódust a blokk elejébe másolja.

4.3.4.9. Normalizált fájlösszefűzés

A normalizált fájlösszefűzés a regisztrált amplitúdótáblázatot felhasználva egy szintre hozza az egyes fájlok adatait úgy, hogy egy blokkot beolvassa a wavappend segítségével normalizált értéken a végleges fájlba kiírja azt. Az átvezetett ideiglenes fájlt azonnal törli, így a művelet nem foglal egyszerre nagy lemeztérületet. Adatfűzés közben információt gyűjt a kimeneti amplitúdószintről, amit összesítetten kijelez a számítás befejeztével jelalak és statisztika formájában.

4.3.4.10. Wav adatkivonási számítás

Ez a funkció a belső konvolúcióhoz hasonló adatszerkezéssel dolgozik, azzal a különbséggel, hogy két ablakolt tartalmaz, és fájlhosszeltérés esetén a rövidebb fáj adatait nullákkal tölti fel.



38. ábra: Wav-kivonási adatszerkezés

5. Program tesztelése, eredmények

A program tesztelésére a következő PC konfigurációt használtam:

- AMD64 Athlon 3000+ 754pin (2000_@_2100MHz) Processzor
- Abit NF8V alaplapp nVidia nforce3 250 lapkakészlet
- 1024Mb DDR3200 Memória
- Samsung SP1614N 160Gb HDD
- Ati Radeon 9600 vga
- Windows XP Home x86 (32bit) SP2
- Matlab 6.5

5.1. Mért jellemzők

A mérésekhez 3 különböző bemenő jelet és 3 különböző RIR függvényt használok egy és kétsatornás változatokban, és vizsgálom az egyes opciók hatását a számításra. A mért eredmények a következők:

RIR hossza	1sec Mono	1sec Stereo	2sec Mono	2sec Stereo	10sec Mono	10sec Stereo
Számítási idő (s)	–	8965	–	–	–	–

15. táblázat: Matriks konvolúciós mód számítási ideje a RIR hosszának függvényében 30 másodperces MONO bemenet esetén

RIR hossza	1sec Mono	1sec Stereo	2sec Mono	2sec Stereo	10sec Mono	10sec Stereo
Számítási idő (s)	161	312	305	614	1709	4411

16. táblázat: Belső konvolúciós mód számítási ideje a RIR hosszának függvényében 30 másodperces MONO bemenet esetén

RIR hossza	1sec Mono	1sec Stereo	2sec Mono	2sec Stereo	10sec Mono	10sec Stereo
Számítási idő (s)	1	4	4	7	13	23

17. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében 30 másodperces MONO bemenet esetén

Jól látható a hatalmas számításidő-különbség. A kihúzott értékeken nem mértem adatot.

<i>RIR hossza</i>	<i>1sec Mono</i>	<i>1sec Stereo</i>	<i>2sec Mono</i>	<i>2sec Stereo</i>	<i>10sec Mono</i>	<i>10sec Stereo</i>
<i>Számítási idő (s)</i>	–	41	–	41	–	138

18. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében 170 másodperces SZTEREO bemenet esetén

<i>RIR hossza</i>	<i>1sec Mono</i>	<i>1sec Stereo</i>	<i>2sec Mono</i>	<i>2sec Stereo</i>	<i>10sec Mono</i>	<i>10sec Stereo</i>
<i>Számítási idő (s)</i>	–	3	–	4	–	11

19. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében 10 másodperces SZTEREO bemenet esetén

	<i>1sec Stereo</i>	<i>2sec Stereo</i>	<i>10sec stereo</i>
<i>RIR:10sec Stereo</i>	3	4	11
<i>RIR:30sec Stereo</i>	4	7	23
<i>RIR:170sec Stereo</i>	41	41	138

20. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében különféle SZTEREO bemenetek esetén

	<i>Mátrix konvolúció</i>	<i>Belső konvolúció 10sec ST x 1 sec ST</i>	<i>FFT-s konvolúció 170sec ST x 2 sec ST</i>
<i>Display off</i>	–	105	41
<i>Display on</i>	–	111	83

21. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében különféle SZTEREO bemenetek esetén

	<i>1sec Stereo</i>	<i>2sec Stereo</i>	<i>10sec stereo</i>
<i>RIR:10sec Stereo</i>	3	4	11
<i>RIR:30sec Stereo</i>	4	7	23
<i>RIR:170sec Stereo</i>	41	41	138

22. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében különféle SZTEREO bemenetek esetén

	30sec Mono x 1sec ST	10sec ST x 1sec ST
Mátrix-Belső	<u>31,5687dB</u>	infinity
Belső-FFT	<u>26,308dB</u>	infinity
Mátrix-FFT	infinity	infinity

23. táblázat: WAVDIFF futtatása az eredményfájlokon,
normalizálási érték

5.2. Értékelés

A mérési adatok hatalmas számítási időbeli különbséget mutattak a módzatok között. A nagyságrendi váltás körülbelül egyenletes. A jelen eredmény kialakulása az utólag beépített FFT-javító funkciónak is köszönhető, az ablakolási méretet ugyanis a kezdetekben kizárólag a program reakcióidejének rendelttem alá, a számítás viszont nagyon kiszámíthatatlan számítás-mennyiséget igényelt. Mikor kettő hatványaira korlátoztam a választható ablakméretet, bizonyos adatkombinációk esetén tízszeres sebességnövekedést tapasztaltam. Százalékosan kifejezve az új funkcióval mért összehasonlítás így írható:

	Sebességarány
Mátrix konvolúció	0,4465%
Belső konvolúció	12,82%
FFT konvolúció	100%

24. táblázat: Sebességarányok

Az FFT gyors számításából adódik, hogy sokkal érzékenyebb a járulékos számítások megjelenésére, mint a kijelzés bekapcsolása. Százalékosan így adódik:

	Sebességcsökkenés
Mátrix konvolúció	-
Belső konvolúció	5,7%
FFT konvolúció	102,4%

25. táblázat: Kijelzés sebességcsökkentő hatása

A 23. táblázat a számítások eredményei között számolt különbségfájl normalizációs értékét mutatja. Rögtön az első adat hibát mutat, vagyis nem végtelen a normalizálási érték, nem nulla a különbség. Hasonlót eddigiek folyamán nem tapasztaltam, a jelen tesztfájlok kombinációja valószínűleg rámutatott egy hibára a kódban, a forrását nem sikerült megtalálnom. A hiba jól láthatóan a belső konvolúciós blokkban van. Hallás után nem lehet különbséget tenni a fájlok között, és debug módban tapasztaltak szerint a 30 másodperces gerjesztőjel végén jelentkezik a hiba. A normalizációs állandót nyomon követve a számítás megegyezik egészen a 28 másodperc környékén tapasztalható amplitúdóugrással, onnan kezdve viszont megváltozik a normalizációs érték, és ez hibát visz a különbségi fájlba is. Jól látható, hogy más gerjesztőfájl használatával a számítások azonos eredményt adnak.

6. Összefoglalás

A dolgozat folyamán áttekintettük a konvolúciós eljárások elméleti hátterét, és a MATLAB segítségével ellenőriztük az egyes módzatok közötti különbségeket. Jelen program tanulsága alapján az FFT-s módszer lekörözhetetlen előnye a nagy sebesség, és ahol nincs szükség az élőben megjelenő jelek azonnali, kis késleltetésű konvolválására, gyakorlatilag nincs értelme a másik két módszer használatának. Az eredmények azt mutatják (eltekintve a programkód hibás működésétől), hogy a kapott eredmény bit pontosságra megegyezik, egzakt, minőségbeli különbség nem jelentkezik. Az FFT-s módszer esetén néhány esetben tapasztaltam a számítási motor kimenetén komplex síkon néhány bitnyi hibaértéket, amellyel érdemben nem foglalkoztam, csupán elhagytam a komplex részt, ez nem befolyásolta a kimeneti adatokat.

7. Irodalomjegyzék

- [1] Huszty Csaba, Augusztinovicz Fülöp : Virtuális akusztikai valóság és auralizáció
http://vibac.hit.bme.hu/documents/84tv2007_hcs.pdf
- [2] Székely Vladimir : Képkorrektció, hanganalízis, térszámítás PC-n
ComputerBooks, Budapest, 1994
- [3] Dr. Simonyi Ernő : Digitális szűrők, A digitális jelfeldolgozás alapjai
Műszaki könyvkiadó, Budapest, 1984
- [4] Norbert Hesselmann : Digitális jelfeldolgozás
Műszaki könyvkiadó, Budapest, 1985
- [5] Dr. Wersényi György : Térbeli hallás
<http://vip.tilb.sze.hu/~wersenyi/Terbeli.pdf>
- [6] Wikipedia : Discrete-time Fourier transform
http://en.wikipedia.org/wiki/Discrete-time_Fourier_transform
- [7] Wikipedia : Fast Fourier transform (FFT)
http://en.wikipedia.org/wiki/Fast_Fourier_transform
- [8] Reliable Software : [Home](#) > [Science](#) > **FFT**: Fast Fourier Transform
<http://www.relisoft.com/Science/Physics/fft.html>
- [9] The Scientist and Engineer's Guide to Digital Signal Processing
<http://www.DSPguide.com> , <http://itl7.elte.hu/html/dspkonyv/>
- [10] <http://www.david-lindsay.co.uk/teaching/MATLABBasics.pdf>
- [11] <http://doaielab.iit.uni-miskolc.hu/education/EXPERT/common/MATLABBASICS.pdf>

8. Ábrák jegyzéke

1. ábra: Többutas terjedés.....	7
2. ábra: Utózengetési idő mérése zajgenerátorral.....	8
3. ábra: Dirac-delta frekvencia-összetevői (vektorhossz).....	11
4. ábra: Dirac-delta jelalakja 100 mintavétel hosszúságú diszkrét jelben.....	11
5. ábra: Barlangban felvett RIR függvény.....	13
6. ábra: Kisméretű templomban felvett RIR függvény.....	13
7. ábra: Barlangi RIR spektruma (vektorhossz).....	13
8. ábra: Templomi RIR spektruma (vekh.).....	13
9. ábra: 2Hz-es jel mintavételi pontokkal.....	14
10. ábra: 38Hz és a fedés.....	14
11. ábra: Kinagyított részlet.....	14
12. ábra: A konvolúció folyamata.....	16
13. ábra: Egyenletes amplitúdó vonalas spektruma.....	18
14. ábra: Impulzusjel spektruma.....	19
15. ábra: Késleltetés hatása a spektrumfázisra.....	19
16. ábra: Ablakolási módok időtartományban.....	20
17. ábra: Kiterjesztés nélküli FFT-konvolúció.....	21
18. ábra: Kiterjesztett FFT konvolúció.....	21
19. ábra: 8-pontos FFT számítási folyamata.....	24
20. ábra: Matlab megjelenése.....	26
21. ábra: M-file szerkesztő megjelenése.....	26
22. ábra: Kettőspont funkciói.....	29
23. ábra: WAV alapformátuma.....	34
24. ábra: ROOMER főablaki megjelenése.....	36
25. ábra: "plot" funkció megjelenése.....	37
26. ábra: FFT plot megjelenése.....	37
27. ábra: Felvételi modul megjelenése.....	38
28. ábra: Konvolúció kiválasztása.....	39
29. ábra: Számítási művelet kiválasztása.....	40
30. ábra: Százalékos nyommonkövetés a fejlécben és grafikusán.....	40
31. ábra: Konvolúció kimeneti jelfolyama.....	41
32. ábra: FFT kimeneti jelfolyama és spektrumábrája.....	41
33. ábra: Folyamatmegszakítási ablak.....	41
34. ábra: Processz-infó megjelenése.....	42
35. ábra: Mátrixkonvolúció adatszervezése.....	45
36. ábra: Belső konvolúció adatszervezése.....	46
37. ábra: FFT-IFFT motoros konvolúció adatszervezése.....	46
38. ábra: Wav-kivonási adatszervezés	48

9. Táblázatok jegyzéke

1. táblázat: Néhány ablakolási függvény számítása és tulajdonságai.....	20
2. táblázat: Fontosabb utasítások.....	28
3. táblázat: Algebrai alapműveletek.....	28
4. táblázat: Relációs műveletek.....	28
5. táblázat: Speciális karakterek.....	28
6. táblázat: Nevezetes változók	29
7. táblázat: Különleges utasítások, ciklusok	29
8. táblázat: ROOMER-ben használt médiakezelő utasítások	30
9. táblázat: ROOMER-ben használt fájlkezelő utasítások.....	30
10. táblázat: Gyakoribb diagramm és grafikai ábrázoló függvények.....	31
11. táblázat: Komplex adatokat kezelő függvények	32
12. táblázat: Statisztikai függvények.....	32
13. táblázat: Statisztikai függvények.....	32
14. táblázat: Grafikai elemek kezelésével kapcsolatos függvények.....	33
15. táblázat: Mátrix konvolúciós mód számítási ideje a RIR hosszának függvényében 30 másodperces MONO bemenet esetén.....	49
16. táblázat: Belső konvolúciós mód számítási ideje a RIR hosszának függvényében 30 másodperces MONO bemenet esetén.....	49
17. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében 30 másodperces MONO bemenet esetén.....	49
18. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében 170 másodperces SZTEREO bemenet esetén.....	50
19. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében 10 másodperces SZTEREO bemenet esetén.....	50
20. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében különféle SZTEREO bemenetek esetén.....	50
21. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében különféle SZTEREO bemenetek esetén.....	50
22. táblázat: FFT-s konvolúció számítási ideje a RIR hosszának függvényében különféle SZTEREO bemenetek esetén.....	50
23. táblázat: WAVDIFF futtatása az eredményfájlokon, normalizálási érték.....	51
24. táblázat: Sebességárányok.....	51
25. táblázat: Kijelzés sebességcsökkentő hatása.....	51
26. táblázat: Matlab matematikai függvényei.....	54

10. Függelék

<i>sin</i>	Színusz	<i>asin</i>	Arkusz színusz	<i>csch</i>	Koszekáns hiperbolikus	<i>acsch</i>	Arkusz koszekáns hiperbolikus
<i>sinh</i>	Színusz hiperbolikus	<i>asinh</i>	Arkusz színusz hiperbolikus	<i>pow2</i>	2-es alapú exponenciális	<i>log2</i>	2-es alapú logaritmus
<i>cos</i>	Koszinusz	<i>acos</i>	Arkusz koszinusz	<i>exp</i>	e-alapú exponenciális	<i>log</i>	e-alapú logaritmus
<i>cosh</i>	Koszinusz hiperbolikus	<i>acosh</i>	Arkusz koszinusz hiperbolikus	<i>log10</i>	10-es alapú logaritmus	<i>sqrt</i>	négyzetgyök
<i>tan</i>	Tangens	<i>atan</i>	Arkusz tangens	<i>lcm</i>	Legkisebb közös többszörös	<i>gcd</i>	Legnagyobb közös osztó
<i>tanh</i>	Tangens hiperbolikus	<i>atanh</i>	Arkusz tangens hiperbolikus	<i>real</i>	Komplex valós része	<i>imag</i>	Komplex képzetes része
<i>cot</i>	Kotangens	<i>acot</i>	Arkusz kotangens	<i>angle</i>	Komplex fázisszöge	<i>conj</i>	Komplex konjugált
<i>coth</i>	Kotangens hiperbolikus	<i>acoth</i>	Arkusz kotangens hiperbolikus	<i>rem</i>	Ostás maradéka	<i>sign</i>	Szignum (előjel) függvény
<i>sec</i>	Szekáns	<i>asec</i>	Arkusz szekáns	<i>round</i>	Kerekítés legközelebbi egészre	<i>fix</i>	Kerekítés egészre (törtrész elhagyása)
<i>sech</i>	Szekáns hiperbolikus	<i>asech</i>	Arkusz szekáns hiperbolikus	<i>ceil</i>	Kerekítés plusz végtelen felé	<i>floor</i>	Kerekítés mínusz végtelen felé
<i>csc</i>	Koszekáns	<i>acsc</i>	Arkusz koszekáns	<i>abs</i>	Abszolút érték (vektorhossz)	<i>atan2</i>	Arkusz tangens (4 negyed)
<i>conv</i>	Konvolúció, polinomiális szorzat	<i>deconv</i>	Dekonvolúció, polinomiális osztás	<i>max</i>	Legnagyobb érték	<i>min</i>	Legkisebb érték
<i>polyval</i>	Polinom értéke	<i>polyfit</i>	Polinomiális interpoláció	<i>mean</i>	Átlagérték	<i>median</i>	Középtérték
<i>polyvalm</i>	Mátrixpolinom értéke	<i>roots</i>	Polinom gyökei	<i>sum</i>	Értékek összege	<i>prod</i>	Értékek szorzata
<i>polyder</i>	Polinom deriváltja	<i>residue</i>	Résztörtekre bontás	<i>sort</i>	Rendezés növekvő sorrendbe	<i>std</i>	Standard eltérés
<i>interp1</i>	Elsőfokú polinomiális interpoláció	<i>interp2</i>	Másodfokú polinomiális interpoláció	<i>cov</i>	Kovariancia adatok összege (valamint tehén elírva)	<i>corrcoef</i>	Korrelációs együtthatók
<i>fmin</i>	Egyváltozós függvény minimuma	<i>fmins</i>	Többváltozós függvény minimuma	<i>trapz</i>	Numerikus integrálás trapéz módszerrel	<i>diff</i>	Differenciál és/vagy derivált
<i>ode23</i>	Alacsonyrendű differenciál megoldás	<i>ode45</i>	Magasrendű differenciál megoldás	<i>del2</i>	Diszkrét Laplace transzformáció	<i>gradient</i>	Gradiens
<i>quad</i>	Alacsonyrendű integrál megoldás	<i>quad8</i>	Magasrendű integrál megoldás	<i>filter</i>	Elsőfokú szűrő	<i>filter2</i>	Másodfokú szűrő
<i>fft</i>	Gyors Fourier transzformáció	<i>ifft</i>	Inverz fft	<i>fft2</i>	2. diszkrét Fourier transzformáció	<i>ifft2</i>	Inverz fft2

26. táblázat: Matlab matematikai függvényei