

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Híradástechnika Intézet



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

MÉLY ZENEI HANGOK ELŐÁLLÍTÁSA DSP-VEL

Szerző: **Maklári-Klekner Mariann**
villamosmérnöki BSc. szak, IV. évf.

Konzulens: **Dr. Wühl Tibor**
egyetemi docens

Budapest, 2014.

TARTALOMJEGYZÉK

Tartalomjegyzék	2
Kivonat.....	4
1. Teljesen sávkorlátozott jelgeneráló megoldások.....	7
2. Hullámtábla szintézis	7
A hullámtábla méret csökkentésének módszerei.....	10
Autokorreláció	11
A hullámtábla-kompresszió tulajdonságai	12
A torzítás minimalizálása	13
3. Kvázi-sávkorlátozott megoldások	13
4. A BLIT-módszer	13
Sinc-ek és SincM-ek összege	13
Ablakozott Sinc-ek összege.....	15
5. A BP-BLIT előállítása.....	16
Más kitöltési tényezőjű BP-BLIT előállítása (PWM)	18
6. Fűrészjelek és négyszögimpulzusok generálása BLIT-ből	19
Fűrészjel	19
Négyszögimpulzus	20
Háromszögjel.....	20
Megfelelő skálázások/offsetek és nem stabil állapot javítások	22
Szivárgó integrátorok	22
7. BLEP-módszer	23
Minimál fázisú impulzusok (minimum-phase impulses)	23
Minimál fázisú ugrásfüggvények	25
PolyBLEP	26

8.	DPW	26
	Az algoritmus túlmintavételezett változata	29
9.	A módszerek értékelése	31
10.	Az alkalmazott mikrokontrolleres környezet bemutatása	32
11.	A megvalósított algoritmusok bemutatása	32
	A keretprogram ütemezése	32
	A megfelelő mintavételi időköz biztosítása	33
	Az SPI busz inicializálása	34
	A digitális-analóg átalakítás megvalósítása.....	35
	A hangminták kalkulálása	36
	Összefoglaló.....	40
	Források	41

KIVONAT

A dolgozat bemutatja a hangszintetizátorokban általánosan használt hullámalakok – szinusz, háromszög, négyszög, illetve fűrész – előállításának módját a mély zenei hangok frekvenciatartományában (E1-től C4-ig, vagyis 41,203 Hz-től 261,63 Hz-ig). Ismerteti a legismertebb hangszintetizálási módszereket (a folytonos jel mintavételezése – triviális megoldás, additív szintézis, hullámtábla szintézis, BLIT, BLEP, PolyBLEP, DPW), és megvizsgálja alkalmazhatóságukat és azok korlátait dsPIC mikrokontrolleres megvalósítás esetén.

BEVEZETÉS

Az elektronikus hangszintézis története a múlt század közepe tájára nyúlik vissza. Kezdetben analóg, digitális, majd hibrid (analóg/digitális) eszközök készültek a hangok előállítására, majd a technika fejlődése és az analóg technika reneszánsza életre hívta a virtuális analóg (az analóg szintetizátorok modelljét követő), illetve más szoftverszintetizátorokat is. [1]

Az idők folyamán számos megoldást fejlesztettek ki a szubsztraktív szintézistechnikát alkalmazó analóg szintetizátorok szokásos jelalakjainak (négyszögjel, négyszögimpulzus, fűrész-, illetve háromszögjel) hatékony előállítására.

Megvizsgálva az előállítani kívánt jelalakok spektrumát, látható, hogy ezek rendkívül széles frekvencia-tartományban tartalmaznak komponenseket, ideális esetben végtelen számú harmonikusból állnak. Ez a gazdag felharmonikus-tartalom egyfelől öröndetes, hiszen ezek megfelelő szűrésével, illetve keverésével rendkívül nagyszámú hangzás állítható elő (ami az említett szubsztaktív szintézistechnika alapelve), ám felvet bizonyos kérdéseket a digitális megvalósításuk kapcsán, hiszen végtelen számú érték előállítására, tárolására, illetve kezelésére nincsen mód. Azonban erre nincs is szükség a mintavételezés jellemző tulajdonsága miatt – a mintavételi tételből következően, ha a jelben nagyobb frekvencia-összetevők vannak, mint a mintavételi frekvenciájának fele, az eredeti információ többé nem nyerhető ki, mert a kapott jel eltorzul a belapolódó komponensek miatt. Ez azt is jelenti, hogy a folytonos („analóg”) jel mintavételezése, illetve az ezzel egyenértékű, pl. digitális számláló értékének növelésével (fűrészjel esetén) operáló, ún. triviális vagy naiv megoldások nem adnak kielégítő eredményt. Ugyanis ezek eredményeképpen végtelen spektrum adódik, így hiába az egyszerű számíthatóság, az előállt hang kellemetlenül zajos, élvezhetetlen lesz.

A szóba jöhető megoldások tehát teljesen sávkorlátozottak a mintavételi frekvencia feléig, vagy az általuk előállt jelek csak olyan kis mértékben szenvedtek átlapolódást, amit a hallgató személy füle nem képes érzékelni.

További probléma, hogy a fenti jelalakoknak ún. diszkontinuitása van, vagyis a hullámformájukban végtelen meredekségű változások vannak. Ezek megfelelő kezelésére is gondot kell fordítani.

A fenti szempontokon túl a kivitelezést több tényező is nehezíti. Az előállítani kívánt hangok frekvenciája alacsony, így azoknak sok harmonikusa esik a hangfrekvenciás, 20 Hz-től 20 kHz-ig terjedő tartományba. Továbbá a mikrokontroller számítási sebessége, adattároló kapacitása, szóhosszának mérete messze elmarad az asztali számítógépekétől. A dsPIC előnye viszont, hogy speciálisan jelfeldolgozásra készített célhardver, így a kialakításából kifolyólag gyorsan képes elvégezni a jelfeldolgozás során igen gyakran előforduló szorzásokat (a két szorzótényezőt a két egyidőben elérhető, különálló, ún. X, ill. Y RAM területekre helyezhetjük el), illetve a szorzások eredményeit a szorzótényezőkénél nagyobb méretű, ún. akkumulátor regiszterbe helyezi el, csökkentve ezáltal a kerekítési hibákat.

Mielőtt sorra vennénk az ismert jelgenerálási módszereket, ejtsünk pár szót az elnevezés- illetve jelölésrendszerről is. Az f_s mindenkor a mintavételi frekvenciát jelöli. A négyszögjel (square wave) egy 50%-os kitöltési tényezőjű négyszögimpulzus hullám 50%-os kitöltési tényezővel (vagyis a négyszögimpulzus jel (rectangle wave) olyan hullámot jelent, aminek másmilyen kitöltési tényezője is lehet). A háromszög jelnek (triangle wave) lehet aszimmetrikus fel/lefelé dőlése (közte az 50%-os kitöltési tényezőjű változattal), a fűrészelnek (sawtooth wave) pedig végtelen meredekség váltása kell, hogy legyen (felfelé vagy lefelé). Így a fűrészel egy háromszögjel 0 %-os vagy 100%-os kitöltési tényezővel. Minden hullámnak lehet unipoláris, bipoláris és tetszőleges eltolású változata. [2]

A sinc a periodikus négyszögimpulzus-sorozat Fourier-sorba fejtéséből keletkező függvény.

1. TELJESEN SÁVKORLÁTOZOTT JELGENERÁLÓ MEGOLDÁSOK

A következőkben azon szintézis módszerek kerülnek ismertetésre, melyek nem állítanak elő a Nyquist-frekvenciánál ($f_s/2$) nagyobb frekvencia-komponenseket, ezzel küszöbölve ki az aliasing zavaró hatását. Ide tartoznak az additív szintézis, az inverz Fourier-transzformáció segítségével történő előállítás, illetve határesetként a hullámtábla-szintézis. Utóbbi esetenként megszegheti a mintavételi tételt, ahogyan ez a későbbiekben kifejtésre kerül. Ezek a megoldások magas minőségű eredményt adnak, ám számításuk általában nagy erőforrást igényel.

Additív szintézisnél a valamilyen módon előállított szinusz vagy koszinusz harmonikusokat adjuk össze a kívánt jelalak Fourier-sorának megfelelően.

Az inverz diszkrét Fourier-transzformációnál a spektrumot állítjuk elő, illetve hullámtábla-szintézisnél a jelalak előre kiszámított, táblázatban eltárolt mintáit olvassuk ki a kívánt alaphfrekvenciának megfelelő módon. Ezek közül a későbbiekben a hullámtáblás módszerrel részletesebben is megismerkedünk.

2. HULLÁMTÁBLA SZINTÉZIS [2]

A hullámtábla szintézis egyike a legeslegrégebbi számítógépes zenében használt szintézis technikáknak (nem keverendő össze a minta visszajátszásos szintézissel, amit szintén hullámtábla szintézisnek neveznek manapság), ám máig széles körben használatosak a zeneszintetizáló alkalmazásokban.[6] Ennél a technikánál a hullámtábla csak egy periódusát tartalmazza a kívánt hangszínnek, nagyszámú minta formájában, pl. $N=512$. A tábla ismételt kiolvasása periodikus hullámformát eredményez F_s/N alaphfrekvenciával. Minden második mintát lejátszva $2F_s/N$ alaphfrekvenciájú jelhez jutunk, és így tovább. Köztes f_0 frekvenciát nem egész ugrástényező (vagy „fázis növekmény” (phase increment), Inc) alkalmazása esetén kapunk, ami a következőképp adható meg:

$$Inc = N \cdot \frac{f_0}{F_s}$$

Olyan esetekben, amikor a hullámtábla címnek törtrésze van, ami gyakran előfordul, vagy eldobjuk a törtrészt, vagy kerekítjük azt a legközelebbi egészre (ez nagyon nagy N esetén vagy nagyon túlmintavételezett hullámformáknál célszerű), de ilyenkor a kívánt érték meghatározható interpolálással is. A legegyszerűbb interpolációs technika a lineáris interpoláció, ám a kereskedelmi forgalomban kapható rendszerekben jellemzően a magasabb rendű interpolációs módszereket, főleg a Lagrange és a sávkorlátozott interpolációt használják. A magasabb rendű változatok további számításokat igényelnek, viszont több, mint 2 szomszédos minta esetén szükséges az alkalmazásuk.[6] Az interpolált hullámtábla szintézis nem garantáltan sávkorlátozott, amikor a fázisugrás nagyobb, mint 1 minta. Ezekben az esetekben elvégezzük a hullámforma egyenértékű decimálását, ahol a decimálási tényező egyenlő a fázisnövekménnyel, mintákban megadva. Ezért nemszinuszos hullámtábláknál a felső korlát a hullámtáblában lévő legmagasabb harmonikus fázisnövekménye:

$$\max(Inc) = N \cdot \frac{f_0}{F_s n_h}$$

Ahol n_h a legmagasabb harmonikus harmonikusszáma.

Az interpolált hullámtábla szintézis elve azonos az additív szintézisével, digitális szinuszos oszcillátort alkalmaz minden harmonikusra. A gyakorlatban a hullámtáblákat általában harmonikusok súlyozott összegeként számítják, valamilyen maximális számú harmonikusig. Ezért vagy az N db oszcillátor kimeneteit adjuk egymáshoz, vagy a hasznos hullámtábláikat adjuk össze azonos súlyozással, hogy egyetlen hullámtábla oszcillátort kapjunk. Ha a legmagasabb harmonikus frekvenciája jóval a mintavételi frekvencia alatt van, és/vagy a harmonikus amplitúdók gyorsan csökkennek – ahogyan az általában történik a természetes hullámformáknál –, az olcsó interpolációs technikák, mint a lineáris interpoláció magas minőségű eredményt adnak.

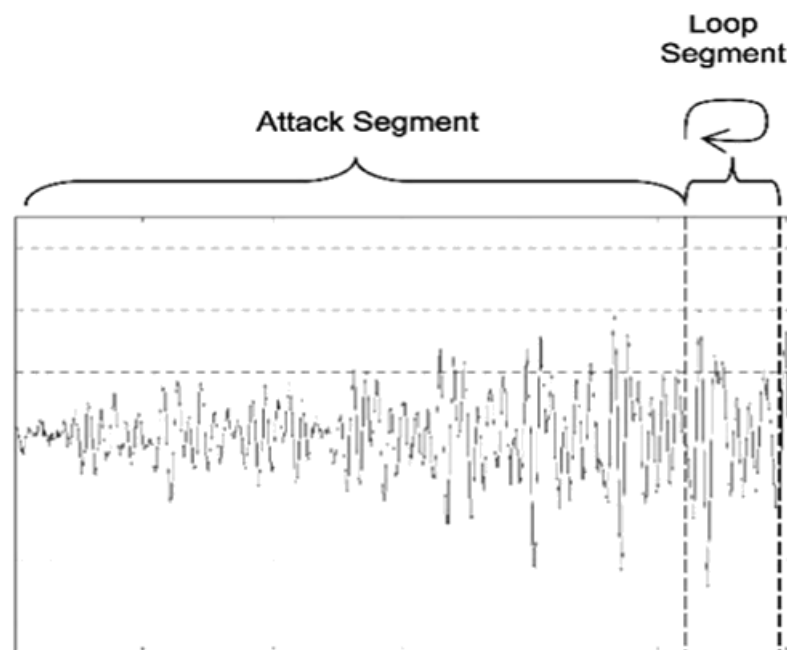
Fontos megjegyezni, hogy a mintavételi frekvencia konverzió sűríti vagy kiterjeszti a hullámtáblában tárolt spektrumot, ami hangszín változásokat eredményezhet, mely általában nemkívánatos jelenség. Az interpoláció átlapolódást is okozhat, ha az nem szigorúan sávkorlátozott. [6]

Mivel praktikussági okokból előnyös a hullámtábla méret csökkentése, kompromisszumot kell kötni: hosszabb hullámtáblát használunk túlmintavételezett adatokkal

a könnyebb újramintavételezés érdekében vagy rövidebb hullámtáblát használunk kritikusan mintavételezett adatokkal, hogy tárhelyet spóroljunk. [6]

A módszer javítása érdekében több megoldást dolgoztak ki. Az egyik ilyen, hogy a ciklikus hullámtáblából kimásolt minták elé egy ún. „attack waveform”-ot (amit talán berezgési hullámként fordíthatunk) helyeznek. Ezt csak egyszer játsszák le, míg a ciklikus táblát többször egymás után, lásd az ábrán.[6] Ez a berezgési hullám a hangszereknek azon tulajdonságát jeleníti meg, hogy azok másképpen viselkednek a hang megszólaltatásának kezdetén, mint a hang állandósult állapotában.

Alkalmazható még amplitúdó burkológörbe a simább átmenetek érdekében, a tábla elejéről és végéről származó minták „összeillesztésénél”, az ismételt lejátszáskor, ún. loopoláskor. Frekvencia-szelektív szűrők segíthetnek a frekvencia (vibrato) és az amplitúdó (tremolo) változások esetén, de „rétegezhetünk” több hullámformát is, és így tovább. [6]



1. ábra: Példa a kétféle tábla használatára [6]

A hullámtábla méret csökkentésének módszerei

Magas minőségű táblák és nagyszámú hangszín esetén, a táblák akár több MB helyet is elfoglalhatnak. Személyi számítógépeknél ez nem probléma, de a mobil és más kis tárolókapacitású eszközöknél kulcskérdés a táblaméret csökkentése. Ennek érdekében a következőket tehetjük:

- a berezgési szakaszhoz használt tábla méretét csökkentjük vagy eltávolítjuk
- ugyanazt a hullámtábla adatot több különböző hangszín esetén is felhasználjuk, módosítás nélkül vagy real-time változtatásokkal (amplitúdó burkológörbék, szűrő beállítások, stb.)
- a szintetizátort alacsonyabb mintavételi frekvencián használjuk, ami ciklusonként kevesebb eltárolt mintát és kisebb hang-sávszélességet jelent
- csökkentjük a hullámtáblában tárolt minták bitszámát
- autokorrelációt alkalmazunk

Észlelési audio kódolók nem használhatók a hullámtábla szintetizátor adatainak tömörítésére, mert szükséges, hogy megmaradjon a folytonosság a tábla visszajátszásakor. Ezek helyett szokásosan a nem egyenletesen kvantált differenciális impulzus kód moduláció (DPCM) eljárást alkalmazzák. Ez egy jól ismert hullámkódoló módszer, mely alacsony komplexitásával kényelmes megoldás a hullámtábla tömörítési arányának változtatásához. Működési elve a következő: a kódoló kiszámolja a különbséget a becsült és a tényleges hullámtábla között. A különbséget ezután kvantálja egy nem egyenletes karakterisztikával (pl. A, μ vagy egyéb logaritmikus karakterisztika). Az ilyen jellegű karakterisztika lehetővé teszi a nagyobb felbontást kis jelrészletek esetén, és viszonylag kisebb pontosságot eredményez nagyobb jelváltozásoknál. A kimenet tömörítése a minta-különbségek dekódolásával és azok összeadásával történik, így áll elő a hullámforma. Viszont a veszteséges kódolás felvet néhány sajátos problémát a loopolt hullámtáblák esetén. Először is a kvantált különbségi adatokból visszanyert, különbségi jel lehet, hogy nem eredményez pontos hurkot (az amplitúdók nem illeszkednek a végpontokon), ami pl. hallható kattanást eredményezhet a ciklikus visszajátszásakor.

Átlapolódási hiba is jelentkezhet a hullámtábla adataiban, mert a kvantálási hiba megjelenése a hullámtábla adataiban a loopolási folyamat során, a hang tervezője által használt antialiasing (vagy újramintavételező) szűrők után lép fel, és elég valószínűtlen, hogy a keletkező hibajel sávkorlátozott. Megemlítendő, hogy a hagyományos statisztikai eljárások, mint a ditherezés, ami a kvantálás hallható hatását javítja, nem megfelelő a hullámtábla-kódoláshoz.

Autokorreláció

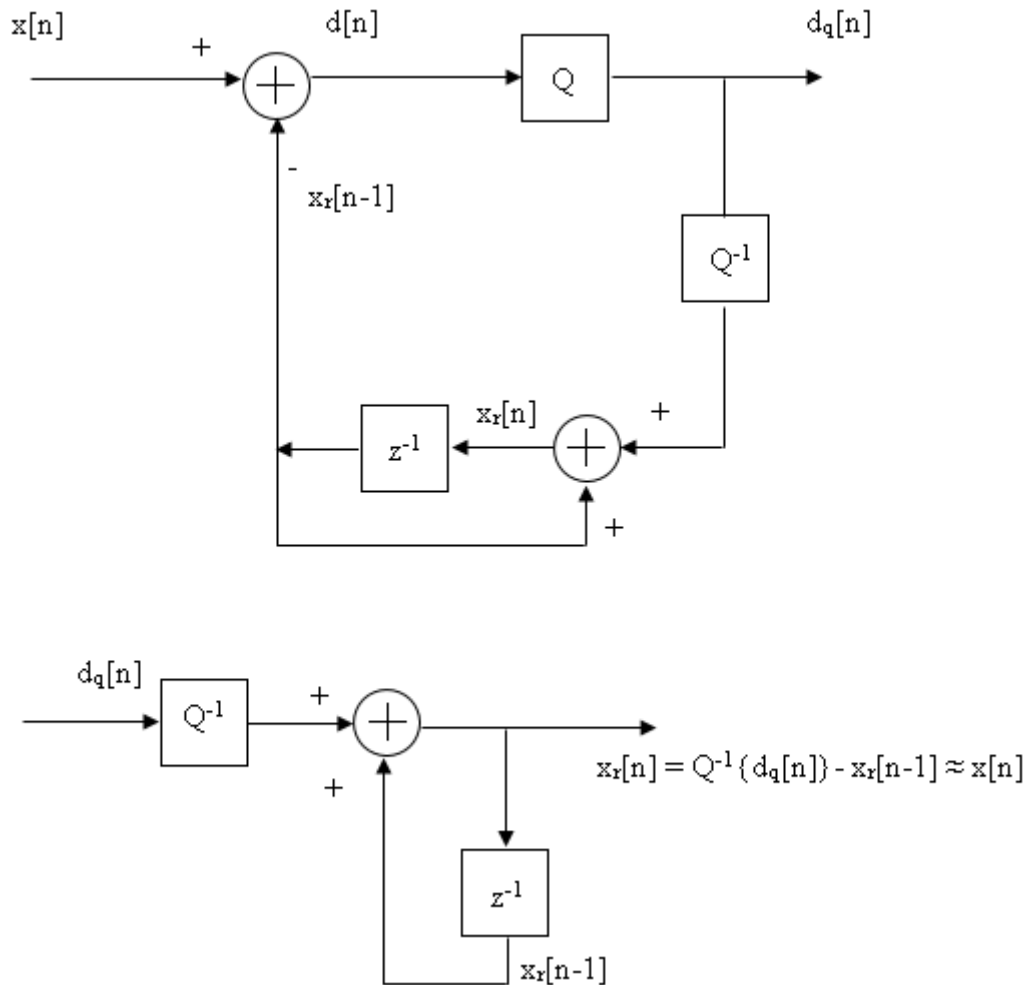
Megállapítható, hogy a minták közti korreláció magas, vagyis a két szomszédos minta közti különbségi jel ($d[n] = x[n] - x[n - 1]$) kisebb eltérést mutat, mint maga az $x[n]$ jel, vagyis némiképpen megjósolható a következő mintaérték. Ezt a redundanciát kihasználhatjuk bitráta-csökkentésre, ami által a jelet kevesebb mintán ábrázolhatjuk. Ez kb. 30-50% csökkenést eredményez elfogadható minőség mellett, ami azt jelenti, hogy az eredeti 16 bites hangmintákat csupán 8 bites nemlineáris differenciális értékekkel, míg kb. 12 bites minőséget eredményez.

A differenciális kódoló és dekódoló egy kényelmes megvalósítását láthatjuk a 2. ábrán.

Ily módon megvalósítva a kódoló és a dekódoló követik egymást, feltételezve, hogy nincs hiba a tárolásban és a visszahatási folyamatban.

A szokásos adatsökkentési szemlélet, hogy megpróbáljuk tovább csökkenteni a redundanciát a jóslás javításával, és hogy durvábban kvantáljuk a különbségi jelet, megengedve a veszteséges kódolást. Az adat kompressziónál fontos, hogy a dekompressziós folyamat olyan egyszerű és gyors maradjon, amilyen csak lehet, de még megfelelő legyen a hangminőség.

A 2. ábrán látható x_n és a x_r közötti hallható különbség minimalizálása nyilvánvalóan előnyös. Nemlineáris, kvantálót használva lényegében veszteségmentes a rekonstrukció, mikor $d[n]$ kicsi, míg némi eltérést megenged, ha $d[n]$ nagy. Erre a feladatra a kvázi-logaritmikus kvantáló elég jónak bizonyult.



2. ábra Differenciális kódoló és dekódoló Q kvantálással. A Q csökkenti a $d[n]$ ábrázolásához szükséges bitszámot, eltérést okozva az x_n és az x_r visszaállított jel között [6]

A hullámtábla-kompresszió tulajdonságai

A veszteséges különbségképzési módszer zajt visz a jelbe. A hagyományos hullámforma-kódolók úgy kezelik a kódolási torzítást, mint az eredeti jellel nem korreláló fehérzaj. A jel-korrelált torzítási komponensek átlapolódást okoznak, amikor a szintetizátor újramintavételezi a jelet, hogy megváltoztassa a hangmagasságot. Ráadásul a torzítási komponensek és a kívánt hullámforma közti korreláció hallható zaj modulációt eredményezhet, amikor a mintavételi frekvencia megváltozik vibrato vagy más hangmagasság-effekt hatására. Az is fontos, hogy gondosan kezeljük a loop-vége állapotot, és ne legyen diszkontinuitás a ciklus határán.

A torzítás minimalizálása

A torzítás minimalizálható a következő módon. A hangtervezési folyamat során a hullámtábla adatai automatikusan ismétlődve kódolódnak különböző amplitúdó-skálázásokkal. Mindegyik esetén kiszámítjuk a dekódolt hullámtábla és az eredeti adat közti a teljes négyzetes eltérést. Később a szintetizátor azt az amplitúdó-skálázást használja, ami a legkisebb torzítást eredményezi (vagyis a legnagyobb jel-zaj viszonyt).

3. KVÁZI-SÁVKORLÁTOZOTT MEGOLDÁSOK

Ezek közös jellemzője, hogy bár viszonylag könnyen számíthatók, valamilyen mértékű átlapolódást megengednek az előállt jelben. A legismertebb ilyen megoldások közé tartozik a BLIT (Bandimited Impulse Train), BLEP (Bandlimited Impulse Step), illetve a DPW (Differential Parabolic Waveform) módszer, illetve ezek javított változatai. Ezek rendre, a nevüknek megfelelően sávkorlátozott impulzussorozat, sávkorlátozott egységugrás, illetve parabolák integrálásával vagy differenciálásával számítják a kívánt hullámformát.

4. A BLIT-MÓDSZER [3]

Ezen megoldás sávkorlátozott impulzussorozat (Bandimited Impulse Train, BLIT) integrálásával állít elő fűrészjeleket, illetve négyszögimpulzusokat. A következőkben sorra vesszük az impulzussorozat előállításának módjait.

Sinc-ek és SincM-ek összege

A mintavételezés előtti alapvető művelet, hogy az anti-aliasing szűrőt alkalmazunk. Az ideális anti-aliasing szűrőnek folytonos idejű impulzusválasza van, ami egy sinc függvény, egy minta értékű nullátmenetekkel:

$$h_s(t) = \text{sinc}(F_s t) = \frac{\sin(\pi F_s t)}{\pi F_s t}$$

Az ideális egység-amplitúdójú impulzus sorozat T_1 szekundum periódusidővel:

$$x(t) = \sum_{l=-\infty}^{\infty} \delta(t + lT_1)$$

Alkalmazva a h_s anti-aliasing szűrőt, a következőt kapjuk ($a * a$ konvolúció műveletét jelöli):

$$\begin{aligned} x_f(t) = (x * h_s)(t) &= \sum_{l=-\infty}^{\infty} h_s(t + lT_1) \\ &= \sum_{l=-\infty}^{\infty} \text{sinc}(t/T_s + lP) \end{aligned}$$

Ahol P a periódusidő mintákban (nem egész szám). Mivel x_f frekvencia-tartományban sávkorlátozott ($-F_s/2; F_s/2$), átlapolódás nélkül mintavételezhető, s így:

$$y(n) = x_f(nT_s) = \sum_{l=-\infty}^{\infty} \text{sinc}(n + lP)$$

A fenti $y(n)$ kifejezés értelmezhető, mint egy időbeli átlapolódásos a sinc függvény P mintányi intervallumban.

Az átlapolódás kiküszöbölése a következőképpen valósítható meg:

$$y(n) = (M/P) \text{Sinc}_M[(M/P)n]$$

ahol:

$$\text{Sinc}_M(x) = \frac{\sin(\pi x)}{M \sin(\pi x/M)}$$

Ez a függvény a mintavételezett BLIT zárt alakját adja, és közvetlenül felhasználható szintézisre. Míg P a periódusidő mintákban, M a harmonikusok száma. Mindig páratlan, mert az impulzus sorozatnak van egy "harmonikusa" DC-nél, és páros számú nem nulla harmonikusa, feltéve, hogy nincs megengedett harmonikus pontosan a mintavételi frekvencia felénél, amit érvényesítünk. Megjegyzendő, hogy M/P mindig közel 1. Amikor P páratlan egész szám, $P=M$ és $y(n)$ egyszerűen $\text{Sinc}_M(n)$. Mivel P eltér M -től, az utolsó előtti egyenlet idő skálázást valósít meg kompenzáló amplitúdó skálázás mellett.

A harmonikusok M számát összekapcsolhatjuk az impulzussorozat P periódusával, mint

$$M = 2 \lfloor P/2 \rfloor + 1$$

Ahol M a legnagyobb páratlan egész szám, ami nem haladja meg P értékét, mintákban.

Eli Brant a módszer összefoglalásakor hozzáteszi, hogy kellően magas f_0 -ok esetében koszinuszok additív szintézisével, alacsonyabbaknál pedig a fenti leírás szerint célszerű szintetizálni a BLIT-et, utóbbinál pedig némi körültekintést igényel a 0/0 eset.

Ablakozott Sinc-ek összege [3]

A sávkorlátozott impulzus egy sinc függvény, ami oly módon skálázott az időben, hogy egy nullátmenete legyen mintaperiódusonként. Ezt a jelet csonkoljuk, és Blackman-függvényt használva ablakozzuk, egy táblázatban eltároljuk, és felhasználjuk a BLIT generálásakor. Kiveszünk egy mintát a táblából, amikor egy impulzusra van szükségünk. Az ablakozott-sinc tábla Ω -szorosan túlmintavételezett. Minden impulzust a táblát Ω -val alulmintavételezve állítunk elő, ha a sávkorlát a Nyquist-frekvencia, ami azt jelenti, hogy Ω mintát átlépünk az időben. Ha Ω elég nagy, a lineáris interpoláció elegendő. Ha az impulzus két minta közé esik, α -val visszalépünk az előző mintához, és elkezdünk lépkedni $\Omega(1 - \alpha)$ -tól.

Az ablakozott sinc tábla felépítéséhez szükséges paraméterek: a Ω túlmintavételezési tényező, N_z a nullátmenetek száma, és az ablakfüggvény. Ω -nak elég nagynak kell lennie, hogy a lineáris interpoláció ne okozzon zajt. Az ablakfüggvény meghatározza a sávzárást – az aliasing mértékét – és az átmenet alakját az áteresztés és a sávzárás között. N_z meghatározza az átmenet szélességét, f_b pedig a helyét, úgy hogy blokkolni tudja az átlapolódásokat. Stilson és Smith Blackman-ablakot használnak, amivel 32 nullátmenet esetén -90 dB alá csökkentik az aliasing mértékét, illetve csak a frekvencia-tartomány felső 10%-a (a magas hangoknál) átlapolódásos.

Ezt a módszert alacsonyabb frekvenciákon érdemes használni, mert nyilvánvalóan sokkal hatékonyabb a harmonikusok összegzésénél, ha a harmonikusok száma nagyon nagy, illetve azon esetekben, amikor a frekvencia sweepel, mint vibrato és portamento esetén; ekkor a jel magas harmonikusai megjelenhetnek vagy eltűnhetnek (a frekvencia sweep irányától függően) a sweepelés közben.

5. A BP-BLIT ELŐÁLLÍTÁSA [2]

A következőkben a bipoláris sávkorlátozott impulzussorozat (BP-BLIT) előállításának módja kerül bemutatásra, mely a négyszögimpulzus generálásához alkalmazható.

Két módszer lehetséges:

1. BLIT-ek különbsége: vesszük két sávkorlátozott impulzussorozat különbségét, az egyiket eltolódva a másikhoz képest.
2. DSF: módszer: azokban az esetekben, amikor a DSF (Discrete Summation Formulae, geometriai sorok összegén alapuló, a szinusz-, illetve koszinusz értékek számítása miatt véleményem szerint igen számításigényes módszer) elfogadható (hatékonyabb) az algoritmus néhány finomítással jól használható arra, hogy közvetlenül generáljuk a BP-BLIT-et.

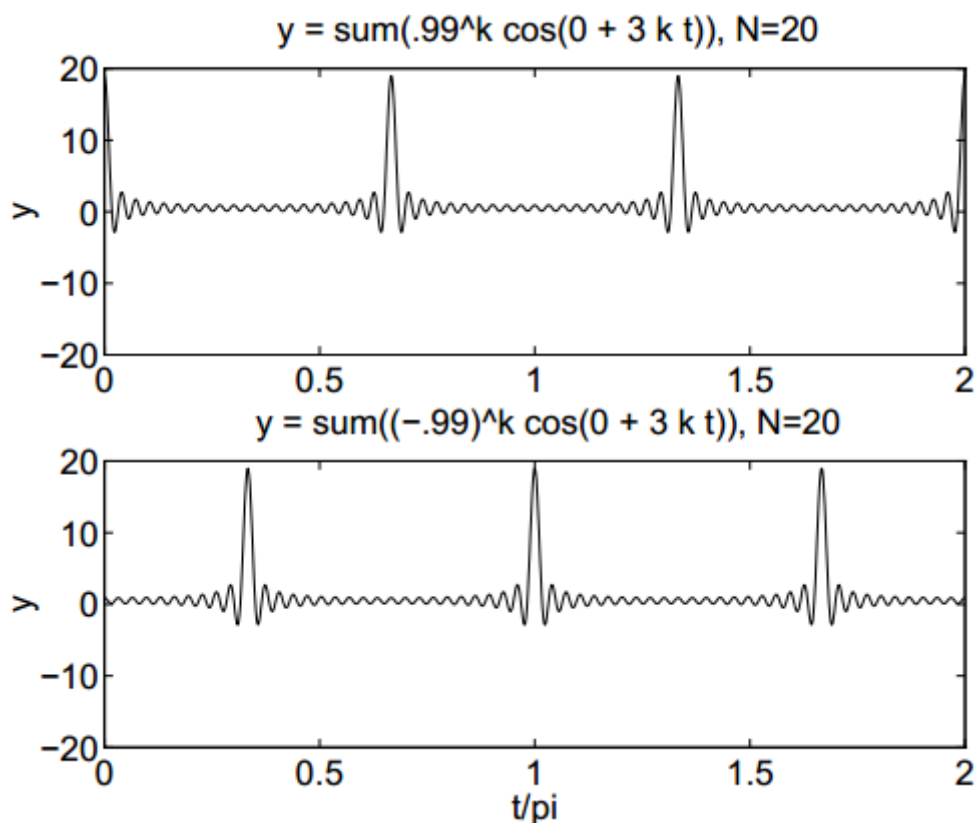
Ez esetben a következőképpen használhatjuk a DSF formulát. Két, 50% kitöltési tényezőjű BP-BLIT-et állítunk elő negatív a -t használva a DSF itt látható összefüggésében:

$$\sum_{k=1}^N a^k \sin(0 + k f_1 t)$$

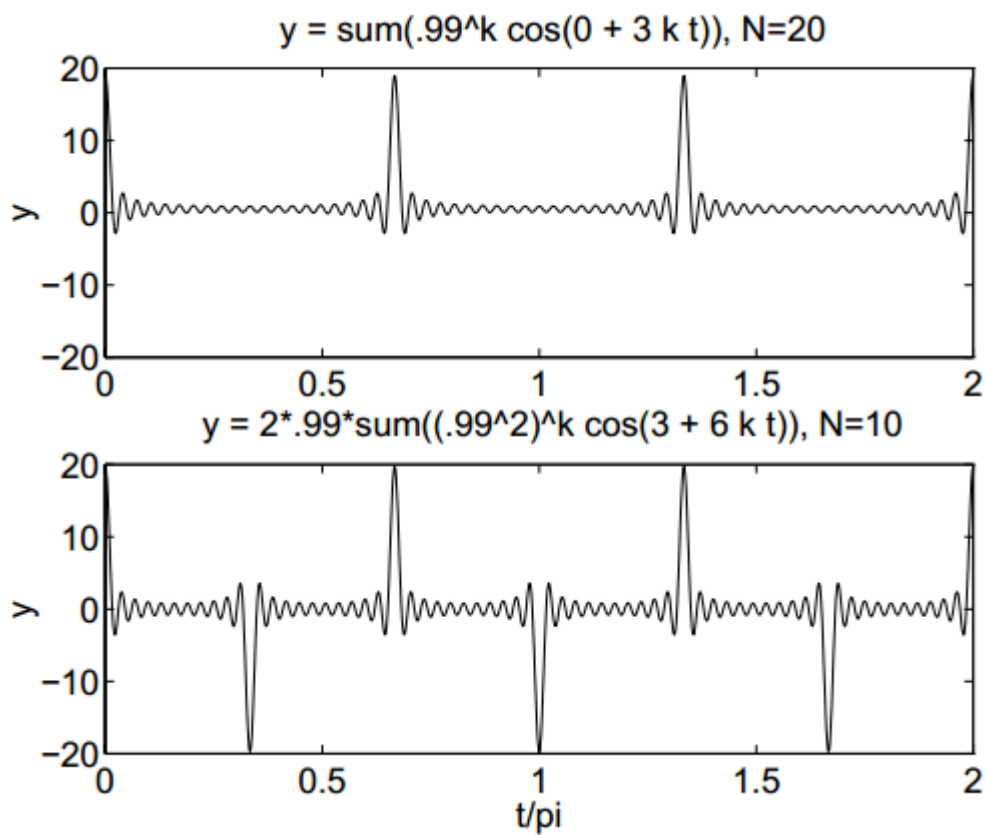
Így olyan jel áll elő, ami a pozitív a -jú jelhez képest pontosan fél periódusnyival van eltolva, a következőképpen:

$$\begin{aligned} & \sum_{k=1}^N a^k \sin(a + bk) - \sum_{k=1}^N (-a)^k \sin(a + bk) \\ &= 2a \sum_{k=1}^{N/2} (a^2)^k \sin((a + b) + 2bk) \end{aligned}$$

Ez valamivel hatékonyabb (vagy elegánsabb...) módja az eltolt BLIT generálásának, mint a t -vel való eltolás, és a következő oldalon látható eredményre vezet:



3. ábra Eltolts sávkorlátozott impulzussorozat képzése



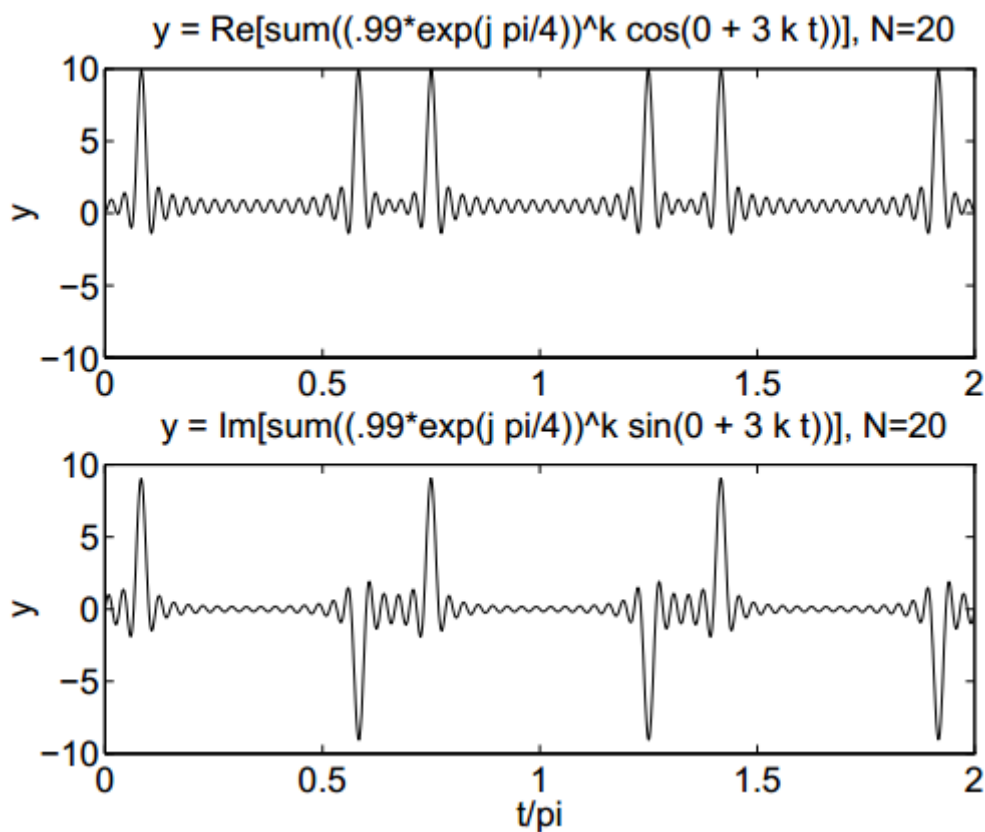
4. ábra A hagyományos és a bipoláris BLIT

A koszinuszok összege DSF formula szintén használható szintézisre.

Más kitöltési tényezőjű BP-BLIT előállítása (PWM)

A módszer a következő: legyen a komplex és vegyük a DSF valós vagy képzetes részét, ami elhelyez egy szinuszos vagy koszinuszos amplitúdó burkolót a harmonikuson, ami ekvivalens a fésű szűréssel, ami viszont egyenértékű az eredeti DSF-nek és az eltoló változatának az összegével. Ez a módszer a BP-BLIT-et lényegében ugyanolyan bonyolultsággal valósítja meg, mint a két DSF-BLIT különbségét számító módszer, csak egy kicsit elegánsabban.

Az eredmény a következő:



5. ábra Tetszőleges kitöltési tényezőjű sávkorlátozott impulzussorozatok

6. FŰRÉSZJELEK ÉS NÉGYSZÖGIMPULZUSOK GENERÁLÁSA BLIT-BŐL [2]

A fűrészjelek és négyszögjelek könnyen származtathatók a BLIT-ből integrálásokon keresztül, amik lineáris transzformációk (triviális szűrőkkel megvalósíthatók), így megőrzik a BLIT sávkorlátozott természetét.

Fűrészjel

A fűrészjel a következőképpen generálható:

$$Saw_{CTS}(t) = \int_0^t CIT(\tau) - C_1 d\tau$$

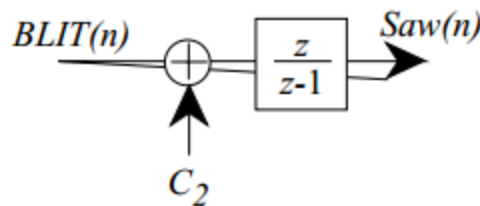
Ahol $CIT(\tau)$ a folytonos idejű impulzus sorozat (Continuous-Time Impulse Train) és C_1 az impulzus sorozat DC komponense. Közvetlenül diszkrét idejűvé alakítható (az impulzus-invariáns transzformáción keresztül):

$$Saw(n) = \sum_{k=0}^n BLIT(k) - C_2$$

$$Saw(z) = \frac{z}{z-1} (BLIT(z) - Z(C_2))$$

Ez triviálisan megvalósítható egy egyszerű összegzéssel és egy 1 pólusú digitális szűrővel. A C_2 a BLIT átlagértéke, amit ki kellene vonni, hogy az integrálás során a jel értéke a végtelenbe tartson (vagy telítődjön).

C_2 a frekvencia függvénye. Az integrátor kezdeti feltételeinek függvényében DC eltolás lesz az integrátor kimenetén.



6. ábra Közvetlen fűrészjel generálás [2]

Négyszögimpulzus

A négyszögimpulzus a következőképpen számítható:

$$Rect_{CTS}(t) = \int_0^t CIT(\tau) - CIT(\tau - t_0) - C_3 d\tau$$

Diszkrét alakban:

$$\begin{aligned} Rect(n) &= \sum_{k=0}^n BLIT(k) - BLIT(k - k_0) - C_4 \\ &= \sum_{k=0}^n BP-BLIT_{k_0}(k) - C_4 \\ Saw(z) &= \frac{z}{z-1} (BP-BLIT_{k_0}(z) - Z(C_4)) \end{aligned}$$

Ahol BP-BLIT a bipoláris BLIT, ami egy olyan impulzussorozat, aminek az impulzusai előjelet váltogatnak.

A bipoláris impulzus sorozat DC komponense 0, ami azt jelenti, hogy $C_3=0$ (és később $C_4=0$). A téglalap szélessége a k_0 -al szabályozható, ami PWM-mel (Pulse Width Modulation, impulzusszélesség-moduláció) változtatható. k_0 értelmezési tartománya ezekben az egyenletekben $[0, \text{periódus}]$.

A BLIT megvalósításától függően a PWM vezérlés értelmezési tartománya $[0,1]$ lehet. Az integrátor kezdeti feltételeitől függően DC eltolás lehet a kimeneten.

Háromszögjel

A háromszögjel úgy generálható mint:

$$Tri_{CTS}(t) = \int_0^t Rect_{CTS}(\tau) - C_5 d\tau$$

Ahol C_5 a téglalap jel kitöltési tényezőjének függvénye:

$$C_5 = \int_0^T Rect_{CTS}(\tau) d\tau$$

Diszkrét formában:

$$Tri(n) = \sum_{k=0}^n Rect(k) - C_6$$

$$Tri(z) = \frac{z}{z-1} (Rect(z) - Z(C_6))$$

A C_5 és C_6 offsetek a négyszögimpulzus kitöltési tényezőjének, valamint a DC offsetnek a függvényei, mely utóbbi a téglalap jelet generáló integrálás kezdeti feltételeiből ered.

$$C_6 = k_0/Period + C_7$$

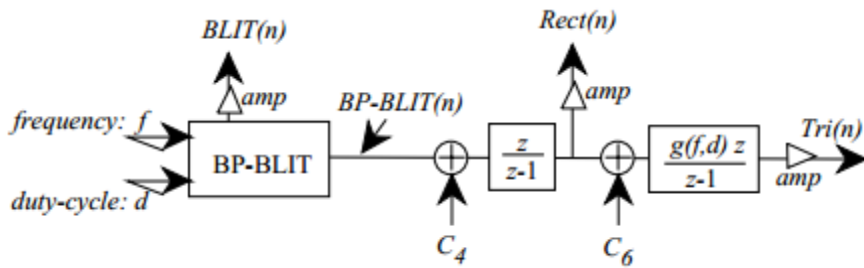
A C_7 a BLIT(0) függvénye, az integrálás kezdeti feltétele.

Hogy megfelelő háromszögjel amplitúdót kapjunk (azért, hogy a szélsőérték ugyanakkora legyen, mint a BLIT-é és a téglalap jelé), egy frekvencia- és kitöltési tényező-függő skálázást kell végezni a háromszög integrációnál

$$Tri(n) = \sum_{k=0}^n g(f,d) (Rect(k) - C_6)$$

$$g(f,d) = \frac{2f}{d(1-d)}$$

Ahol f a frekvencia (jel frekvencia/mintavételi frekvencia) egységekben (f_s -re normalizált frekvencia), és d a kitöltési tényező ($d \in [0;1]$).



7. ábra Négyszögimpulzus és fűrészjel generálás [2]

Megfelelő skálázások/offsetek és nem stabil állapot javítások [2]

Ahhoz, hogy az megakadályozzuk, hogy az integrátorok kimenetei a végtelenhez tartsanak (vagy legalábbis a számrendszer vagy a DAC határához), minden DC eltolást el kell kerülni az integrátor bemenetén (az ideális integrátor DC eltolása végtelen [3]). Mivel a BP-BLIT-nek nincs DC eltolása, ezért nem szükséges egyéb eltolás a négyszögjel integrálásához. Az első integrátor kezdeti feltételei viszont állíthatnak elő DC eltolást a kimenetükön, amit meg kell szüntetni a második integrálás előtt. Ennek az eltolásnak az értéke a jel kitöltési tényezőjének értékétől is függ, ezért a helyes bemeneti feltétel a következők szerint változik meg: (1) a kívánt fázis, (2) a kívánt kitöltési tényező. A második integrálásához frekvenciafüggő skálázás is szükséges (mert a háromszögjel meredeksége a frekvenciával arányos), amelyek határait figyelembe kell venni a frekvencia változtatásakor. Fontos megjegyezni, hogy a régi (közvetlenül a változás előtti) állapot új "kezdeti feltételként" viselkedik az integrátor számára, amikor a paraméterek megváltoznak.

Megjegyzés: minden amplitúdó-függőség elkerülhető egység amplitúdót feltételezve minden integrálásnál, majd egyszerűen után skálázva a kimeneteket. Így a skálázási erősítések: az első integrátor bemenetén 1,0, a második integrátor bemenetén $2/T_d(1-d)$, ahol d a kitöltési tényező és T a periódusidő mintákban.

Szivárgó integrátorok [2]

A tiszta integrátorok használata problémákat okozhat a számbeli hibák (pl. kerekítés) miatt. Ezek a hibák felhalmozódnak az integrátorokban, nem kívánt (és kiszámíthatatlan) eltolásokat okozva a jelben, ami miatt előfordulhat, hogy nem a kívánt hullámformákat kapjuk a kimeneten, különösen igaz ez a második integrálásra.

A hiba kiküszöbölése érdekében a szűrők pólusait, amik az integrálást végzik, enyhén elmozdítjuk az egység körből. Ezek a „szivárgó integrátorok” lassan „elfelejtik” a rossz bemeneti feltételeket és számhibákat, így azok nem folytathatják a halmozódást a végtelenségig. Vagyis ezen integrátorok impulzus-válasza exponenciális, ami lassan lecseng 0-ra.

A másodfokú szivárgó integrátor DC erősítése 0, és egy egypólusú szivárgó integrátor és egy egypólusú felüláteresztő szűrő segítségével valósítható meg a következő összefüggés alapján:

$$H(z) = \frac{\pi}{1 - cz^{-1}} \frac{k(1 - z^{-1})}{1 - cz^{-1}}$$

Ahol $k=(1+c)/2$ és $c=0.99992$, mely konstansok segítségével az integrátort a hangfrekvenciák átengedésére hangoljuk, vagyis 0,5 dB-es csillapítást alkalmazunk 20 Hz-en ($f_s=44,1$ kHz) Az integrálás kezdeti feltétele:

$$y_b(0) = 1 - r_{01} \quad [3]$$

7. BLEP-MÓDSZER [3]

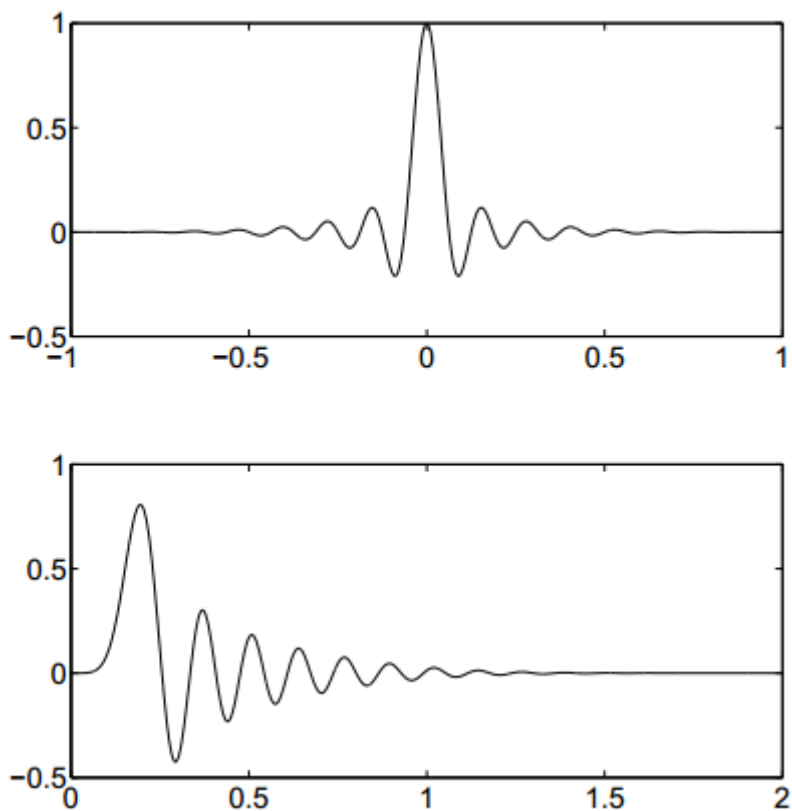
A BLEP (Bandlimited Impulse Step – sávkorlátozott egységugrás függvény) módszer alkalmazásakor nem kell elvégezni a BLIT-módszernél megismert nehézkes integrálási műveleteket a kívánt hullámforma előállításához. A sávkorlátozott egységugrások az ablakozott sinc elő-integrálásával és skálázásával adódnak, az eredmény végértéke pedig biztosíthatóan 1. A mintákat táblázatban tároljuk.

A megoldás az ideális jelben található végtelen meredekségű ugrások sávkorlátozásán alapul azáltal, hogy a triviális jel ugrásait sávkorlátozott egységugrásokra cseréli.

Minimál fázisú impulzusok (minimum-phase impulses)

A sávkorlátozás elvileg megvalósítható ablakozott sinc-ek összegével, az additív szintézishez hasonlóan. Ám a módszer bonyolult, mivel az ablakozott sinc függvény szimmetrikus, és az energiájának nagy része a középső nyálábában van, így figyelni kell a középpont helyét Ez a figyelési probléma megszűnik, ha úgy tekintünk az ablakozott sinc-re, mint egy FIR-szűrőre, és ezt a minimál fázisú szűrőt ugyanolyan impulzusválasszal készítjük el, mint amilyen az ablakozott sinc, és azt használjuk helyette.

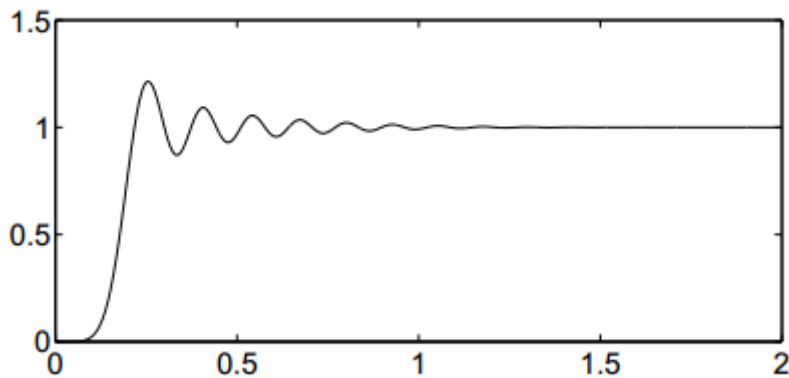
A z-síkú módszerek bonyolultak az olyan hosszú impulzusválaszú minimál fázisú szűrők esetén, mint a jelen eset, ellenben cepstrum síkon¹ végezve azt, robosztus és gyors. A MATLAB *rceps* függvénye közvetlenül elvégzi ezt a műveletet.



8. ábra Ablakozott sinc impulzus (felül) és minimál fázisú ablakozott sinc impulzus [3]

¹ „A cepstrum többféle definíciójával találkozhatunk a szakirodalomban. A leginkább elfogadott azonban a logaritmikus frekvenciaspektrumból képzett spektrum. Ez azt jelenti, hogy a bázisként használt spektrumnak lineáris a frekvenciaskálája, de az amplitúdó tengely logaritmikus.” [4]

Minimál fázisú ugrásfüggvények



9. ábra A minimál fázisú sávkorlátozott ugrásfüggvény [3]

A korábban említett fejlesztések kombinálhatóak. Nem szükséges az impulzusközépfigyelés, és nem kell a szivárgó integráló fokozat sem. A minimál fázisú sávkorlátozott impulzust integrálva minimál fázisú sávkorlátozott ugrásfüggvényt (Minimum Phase Bandlimited Step – MinBLEP) kapunk, amely segítségével közvetlenül megoldható a végtelen meredek ugrások (szakadások) miatti hullámforma-átalakítási probléma azáltal, hogy a triviális jelalak végtelen meredek ugrásait korrigáljuk a sávkorlátozott és az ideális ugrások mintáinak különbségét tartalmazó táblázat (BLEP maradék tábla) megfelelő értékeivel. [10]

A módszer alkalmazása viszont csak akkor megfelelő megoldás, ha a kívánt függvény első- és a második deriváltja folytonos az eredeti függvény szakadási pontjában. A négyszög- és a fűrészjelre ez teljesül, így azok átlapolódásos digitális szintézise egyszerűen javítható azáltal, hogy helyettesítjük az ugrásaikat a MinBLEP-ekkel.

Hatékonyaság

A MinBLEP egy mintára eső költsége durván arányos f_1 -gyel. A MinBLEP algoritmus egy osztást igényel, minden periódusban, és a számítási terhelés függ az alaphfrekvenciától. Ez fogyaszt némi memóriát, mert a túlmintavételezett MinBLEP sorozatot előre meg kell tervezni a valós idejű megvalósításhoz.

PolyBLEP

A PolyBLEP (Polynomial Bandlimited Step, polinomos sávkorlátozott egységugrás függvény) módszer a BLEP maradék táblát egy zárt alakban megadott polinom közelítésre cseréli. A módszert alkalmazva csak a szakadási pontok előtti, illetve utáni mintákat korrigáljuk. Az alkalmazott, úgynevezett Lagrange-polinomok integrálhatók, és használhatók a közelítő sinc függvény számításához.

Hangszintézisre történő felhasználás esetén az alacsonyrendű esetek érdekesek (N=1, N=2, N=3). A fokszám növelésével több mintát korrigálhatunk, de ezáltal nő számításigény is. [8]

8. DPW

Egy, a fűrészjelhez hasonló, de kisebb átlapolódási hibájú jel állítható elő egy parabolikus hullámforma differenciálásával. A fűrészjel integrálásával egy szakaszosan parabolikus hullámot kapunk, ennek egy periódusa a következőképpen írható fel:

$$p(t) = \int t dt = \frac{t^2}{2},$$

Ahol:

$$-\frac{T}{2} \leq t \leq \frac{T}{2}$$

Ami t szerint differenciálva visszaadja a lineáris függvényt:

$$s(t) = \frac{dp(t)}{dt} = t,$$

Ahol:

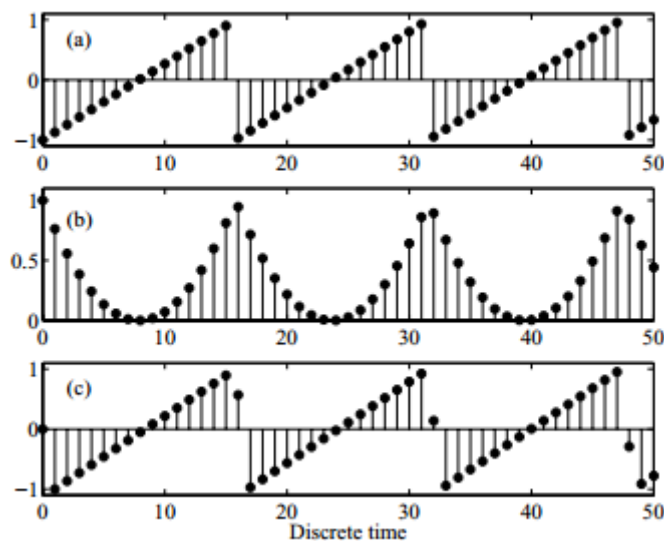
$$-\frac{T}{2} \leq t \leq \frac{T}{2}$$

A szakaszonként parabolikus jel használatának oka, hogy a spektruma kb. 12 dB/oktávval esik, míg a fűrészjelé kb. 6 dB/oktávval. A gyorsabb spektrumesés elnyomott átlapolódáshoz vezet. A differenciálás visszaállítja a kívánt spektrumesést, de alacsony és közepes frekvenciákon a spektrum átlapolódásából származó komponensek szintje alacsonyabb, mint a triviális fűrészjel esetén.

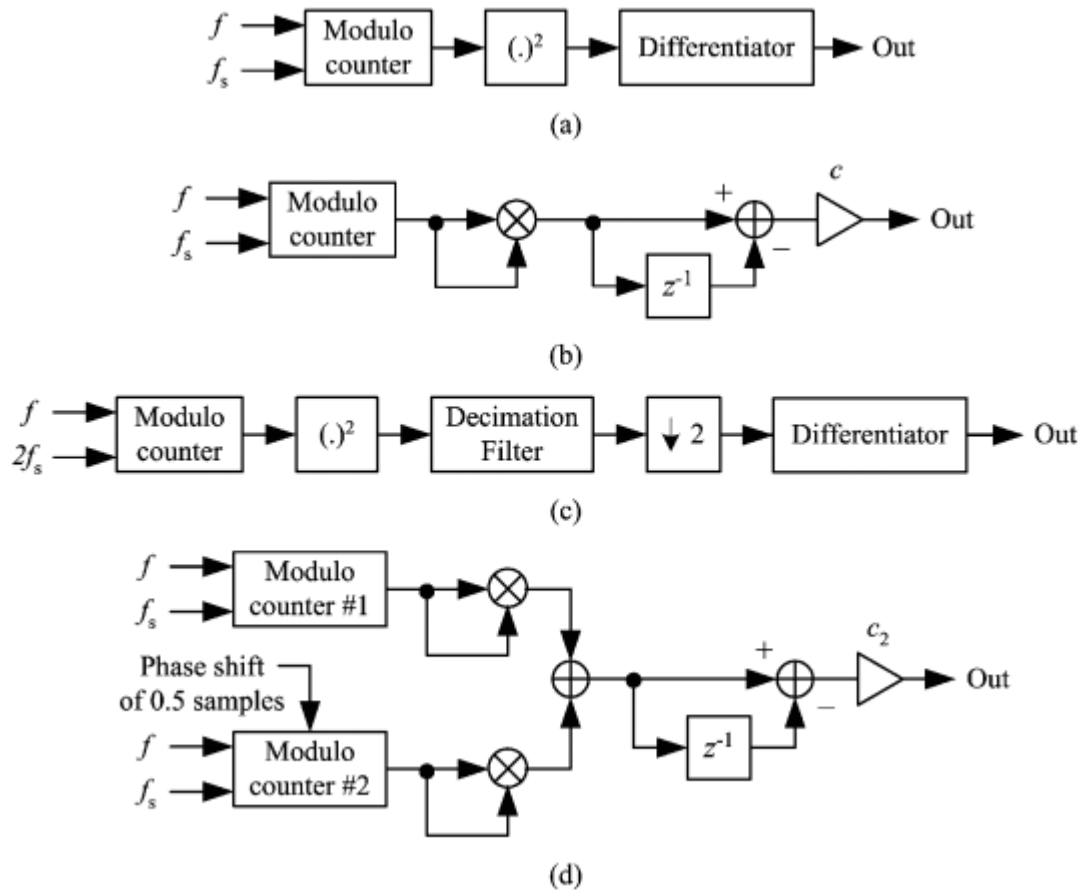
A szakaszosan parabolikus jelgenerálást és differenciálást DPW (Differential Parabolic Waveform) algoritmusnak nevezzük. Alapelvét, a fűrészjel generálásán keresztül a következő ábra mutatja. A triviális fűrészjelnél megismert, annak előállítására használt modulo-számlálót megvalósító összefüggés itt is használható a sávkorlátozatlan jel előállítására:

$$s(n) = 2 \left(n \frac{f}{f_s} \bmod 1 \right) - 1$$

Ahol $n=0, 1, 2..$ a diszkrét minta index, f a jel alapfrekvenciája és f_s a mintavételi frekvencia.



10. ábra (a) A triviális fűrészjel hullámformája - $s(n)$ (b) a négyzetre emelt fűrészjel - $s^2(n)$ és (c) differenciált parabolikus hullám 2793.8 Hz-es jel esetén - $c[s^2(n) - s^2(n-1)]$ [7]



11. ábra A fűrészel generálást megvalósító blokkdiagramok: (a) a DPW algoritmus általános formájára (b) az egyik részben hatékonymegvalósítására (c) a DPW2X általános formájára és (D) hatékony megvalósítására

A DPW algoritmus alternatív realizálása a differenciáló típusának megválasztásával lehetséges. A legegyszerűbb választás az elsőfokú véges impulzusválaszú (FIR) differenciáló, melynek átviteli függvénye:

$$(1 - z^{-1})/2$$

Ez a szűrő jobban csillapítja a jelet magas frekvenciákon, mint az ideális differenciáló, de ez megfelelő lehet azokban a hangtechnikai alkalmazásokban, amikor a mintavételi frekvencia magas, pl. 44,1 kHz.

Az alap DPW algoritmus hatékony változata az előző ábra (b) részén látható. Az algoritmus két szorzást és 2 összeadást igényel, egyiket a számláló növeléséhez.

A c együttható: $c = f_s/[4f(1 - f/f_s)]$, amely tartalmazza a FIR differenciáló 0,5 értékű tényezőjét, mely úgy skálázza a kimenő jel amplitúdóját, hogy az kisebb vagy egyenlő legyen, mint 1. Amikor f kicsi, c együttható megnő, és így a differenciáló utáni jel értéke is nagyon alacsony lesz. Ez kvantálási zaj problémákhoz vezethet, véges pontosságú számításokkal, alacsony frekvenciákon.

Az átlapolódás széles frekvencia-tartományban csökken a triviális fűrészjelhez képest. Az átlapolódás csak a Nyquist-frekvencia közeli frekvenciákon erős az átlapolódás és közelíti meg a kívánt spektrum komponensek szintjét, ám az ilyen magas frekvenciákon az emberi hallás érzékenysége alacsony, a mély hangok pedig csak alacsony amplitúdójú komponenseket tartalmaznak.

Az algoritmus túlmintavételezett változata

A túlmintavételezés általános eljárás az átlapolódás csökkentésére. A kétszeresen túlmintavételezett DPW algoritmusnál (DPW2X) a parabolikus hullámformát kétszer nagyobb mintavételi frekvenciával mintavételezzük, mint a kimeneti mintavételi frekvencia. A jelet ezután egy decimáló szűrővel szűrjük, így a mintavételi frekvencia lecsökken a felére, azáltal, hogy minden második mintát átugrunk, így végül a jel differenciálásra kerül (11. c ábra).

A decimáló és a differenciáló szűrők megválasztásával alternatív megvalósításokat kapunk az algoritmusra. A decimáló szűrőnek alacsony fókuszának kell maradnia, mert az algoritmus teljes komplexitásának nem szabad jelentősen nőnie. A következő átviteli függvényt megvalósító átlagoló szűrő megfelelő eredményt adhat:

$$(1+z^{-1})/2.$$

Ez magas csillapítást jelent a túlmintavételezett Nyquist-frekvencián, míg az átviteli sávra – vagyis az eredeti Nyquist-frekvencia alatt – a csillapítása kevesebb mint 3 dB, ami elfogadhatónak tekinthető. Az előző esethez hasonlóan FIR differenciálást használunk, lásd a 11. d ábrán.

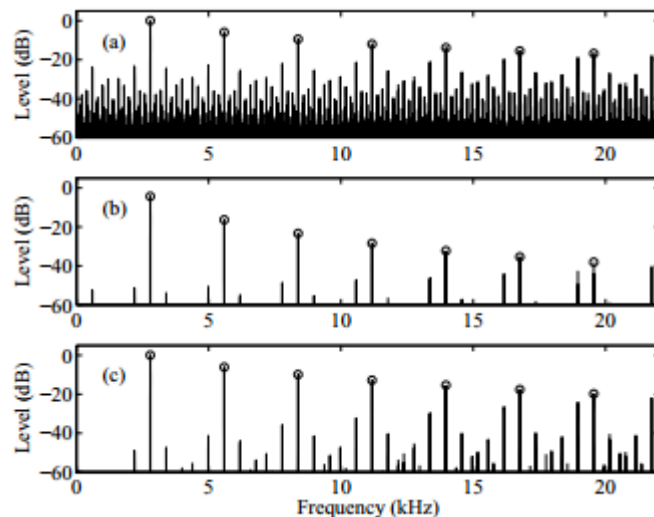
A skálázó együttható:

$$c_2 = f_s / [8f(1 - f/f_s)]$$

mely egyértékű konstans amplitúdót eredményez.

Ez a módszer 4 összeadást (1-1 a modulo számlálókra, és az egyes szűrőkre), és 3 szorzást használ mintánként. Alkalmazása tovább csökkenti az átlapolódást alacsony és közepes frekvenciákon, ahol a hallásunk a legérzékenyebb. Az algoritmus erőssége az egy mintára jutó műveletek kis száma, és a csekély memória-felhasználás, ez vonzóvá teheti akár mobil alkalmazások esetén.

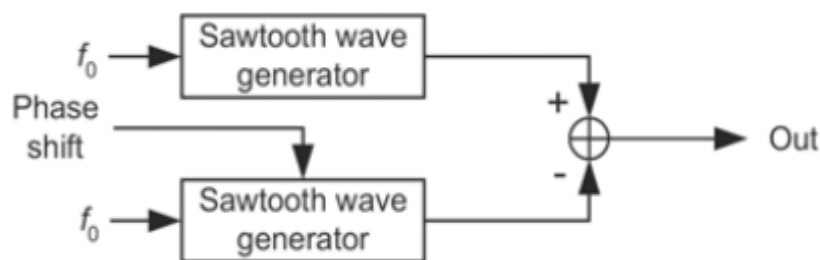
A DPW algoritmusok hatékony megvalósításai az SNR-t (a jel/zaj viszonyt) hozzávetőleg 10-15 dB-lel javítják a triviális fűrészelhez képest.



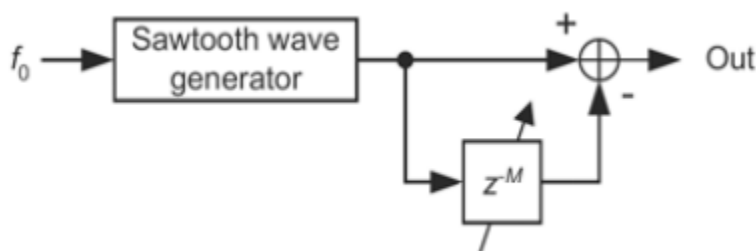
12. ábra (a) A triviális fűrészel spektruma (b) a négyzetre emelt változaté (c) a differenciált parabolikus hullám. A kívánt frekvencia komponenseket (2793.8 Hz, 5587.6 Hz, 8381 Hz, ...) kör jelöli (o), míg a többi az aliasing által okozott, zajként hallható komponensek [7]

Az algoritmus négyszögjelekre is alkalmazható a következőképpen [9]:

1. két fűrészel kivonásával:



2. FIR fésű szűrőt alkalmazva a fázistolás előállítására, majd elvégezve a kivonást:



9. A MÓDSZEREK ÉRTÉKELÉSE

Könnyen belátható, hogy bár a triviális megoldások rendkívül könnyen számíthatóak, az átlapolódási jelenség miatt önmagukban egyáltalán nem használhatók.

A teljesen sávkorlátozott megoldások, úgymint az additív szintézis, illetve az inverz Fourier-transzformációt használó szintézis bár jó minőségű eredményt képesek megvalósítani, túlságosan számításigényesek a hatékony implementáláshoz, főként a kérdéses frekvenciatartományban. Véleményem szerint igazán hatékonyan csak magas alaphfrekvenciák generálása esetén használhatók, mert ilyen esetekben az előállítandó jelalakoknak csak kevés harmonikusa esik a hallható tartományba, ennek megfelelően az algoritmus is egyszerűsödik.

A hullámtáblás módszer megfelelő átalakításokkal alkalmas a kis tárkapacitású eszközökön történő alkalmazásra, de mivel a mobil eszközökre való alkalmazhatósághoz rengeteg finomítást kell végezni, és még így is viszonylag nagy memóriaterületet igényelnek az eltárolt minták. Így bár viszonylag jó eredményt biztosít alacsonyabb bitrátákon is, nem, vagy csak a lehetséges hibalehetőségekre való igen nagy körültekintéssel javaslom az implementálását a jelen, mikrokontrolleres környezetben.

Az ablakozott sinc függvények segítségével előállított BLIT sorozat integrálása látszólag jó eredményt produkál mély hangok esetén, ám a szivárgó integrátorok megvalósítása miatti problémák, illetve négyszögimpulzusok esetén a bipoláris BLIT előállításának szükségessége a BLEP és DPW módszereket teszik ígéretesebbé.

Konklúzióként, egyszerű számítási módszere miatt a DPW, illetve a PolyBLEP módszereket tartom legígéretesebbnek a mikrokontrolleres környezetben történő megvalósításra. Ezek kellően egyszerűen számíthatók, illetve vélhetően elég magas minőséget biztosítanak a mély hangok hatékony generálásához.

10. AZ ALKALMAZOTT MIKROKONTROLLERES KÖRNYEZET BEMUTATÁSA

A kiválasztott módszer megvalósítására a Microchip gyártmányú, dsPIC33F családba tartozó, dsPIC33FJ256GP710 típusszámú mikrokontrollert választottam. A processzor 16 bit szóhosszúságú adatokkal dolgozik, belső memóriája 256 kByte.

A kerekítési hibák csökkentése érdekében a szorzások eredményeit 40 bites akkumulátor regiszterekben tárolja, melyből 2 db áll rendelkezésre.

Az algoritmusok implementációja a dsPICDEM Starter Demo Board V2 panelen, illetve az MPLAB X IDE 2.10 fejlesztői környezet segítségével történt. Programozó és debugger eszközként ICD3-t használtam.

Az alkalmazott D/A konverter (digitálisról analógra átalakító) csak a felső, 12 legértékesebb bit átalakítását tette lehetővé.

Az algoritmusok assembly nyelven íródtak.

11. A MEGVALÓSÍTOTT ALGORITMUSOK BEMUTATÁSA

A számszerű minták előállításához szükség van egy keretprogramra, amely előállítja a rendszer órajelét, biztosítja a megfelelő mintavételi időközt, vezérli az adatokat szállító buszt (SPI2), illetve a hallható hanggá alakításért felelős digitális-analóg átalakítót (D/A konverter).

A következőkben ezen keretprogram legfontosabb elemeit mutatom be.

A keretprogram ütemezése

A keretprogram ütemezését a Timer3 beépített hardveres időzítő segítségével valósítottam meg. Az időzítő órajelét az FCY rendszer-órajelből származtatom, ebből vezetem le a megszakítások gyakoriságát is. Ehhez a PR3 regiszterbe betöltöttem az FCY/F_s konstanst, ahol $FCY=39936000$, valamint $F_s=44100$. Ezen beállítás szerint másodpercenként 44100 mintát küldök a kimenetre, vagyis a mintavételi frekvencia 44100 Hz-re adódik.

```

125      Init_Timer3:
126          clr      T3CON
127          mov      #FCY/Fs, WO
128          mov      WO, PR3
129          clr      TMR3
130          bclr     IFSO, #T3 IF
131          bset     IECD, #T3 IE
132          bset     T3CON, #TON
133          return

```

13. ábra A Timer3 időzítő beállítása

A megfelelő mintavételi időköz biztosítása

A jelgeneráló algoritmust __T3Interrupt: címkével jelzett megszakítás kezelő rutinban valósítottam meg, melynek meghívási gyakorisága a Timer3 beállításai alapján 44100 Hz.

Ez azt jelenti, hogy minden megszakításkor keletkezik egy minta, másodpercenként pedig 44100 jelmintát kalkulálunk. Ez elegendően nagy mintavételi frekvencia ahhoz, hogy elkerüljük az átlapolódási jelenséget a kimeneten (a legnagyobb hallható tartományba eső frekvenciakomponens 20000 Hz, ennek kétszerese 40000 Hz, a fennmaradó 4100 Hz pedig a sávkorlátozó szűrő miatti védősáv).

```

196      __T3Interrupt:
197          bclr     IFSO, #T3 IF

232          btss    SPI2STAT, #SPITBF
233          call     DA_write
234          retfie

```

14. ábra Az interrupt rutin megvalósítása

A 197 jelű sor törli a Timer3 interrupt flag-jét, míg a 232. sor az adattovábbításért felelős SPI buszt teszteli. A 233. sorban meghívjuk a digitális-analóg átalakításért felelős szubrutint, majd a 234.-ben az interruptból való visszatérés történik a főprogramba.

A 198-231. sorok között helyezkednek el a hangminták kalkulálásáért felelős utasítások.

Az SPI busz inicializálása

```

137      Init_SPI2:
138
139          mov     #0x0523, w0

149          mov     w0, SPI2CON1
150
151          clr     SPI2STAT
152
153          bclr    IFS2, #SPI2IF
154          bset    IEC2, #SPI2IE
155          bset    SPI2STAT, #SPIEN
156          return

```

15. ábra Az SPI busz inicializálását végző kódrészlet

Az SPI 2 buszt a minták szállítására használjuk a jelgeneráló egységek és a D/A konverter között.

Ennek beállításához először a #0x0523 értéket kell beírni az órajel-előosztásért, a master/slave mód beállításáért, a szó/bájt kommunikációért, az órajel-polaritásért és az órajel/adat láb működéséért felelős konfigurációs regiszterbe (SPI2CON1), mivel balról jobbra a biteknek a következő értéket kell felvenniük: a 15-13 bit nem megvalósított lábakat jelöl, így értéke 0, a 12. bit (DSSCK) 0 értékű az órajel engedélyezéséhez. Ez a 4 bit hexadecimálisan 0 formájában írható. A 11. bitet (DISSDO) 0 értékre állítjuk, hogy Data Output lábat a modul vezérelje, míg a 10. bit (MODE16) 1 értéke kiválasztja, hogy 16 bites egységekben kommunikáljunk. Master módot használunk, így a 9. bit (SMP) 0 értéke azt jelenti, hogy a bejövő adatot az üzenet végére helyezzük, míg a 8. bit (CKE) 1 értéke az órajel váltást (SCK) definiálja. Ez a 4 bit hexadecimálisan 5 értéket jelent.

A 7. bit (SSEN) 0 értékű, vagyis az SSx lábakat a port függvény vezérli, a 6. bit (CKP) pedig szintén 0 értéket vesz fel, vagyis a clock tétlen állapota az alacsony szint, az aktív állapota a magas szint. Az 5. bit (MSTEN) "1", vagyis Master módot használunk. A 4-2. bit mindegyike 0-t vesz fel, tehát a másodlagos előosztó 8:1-hez értékű míg az 1-0 bitek "1"-ben vannak, ami az elsődleges előosztót 1:1-be állítja. Ez hexadecimálisan 23 érték.

Az SPI2STAT törlése az overflow státusz törlését jelenti, míg a 153. sorban lévő #SPI2IF bit törlése biztonsági okokból történik a véletlen interruptok elkerülésére. (Ennek hatására a beérkező megszakítási események állítják 1-be ezt a bitet.) A 154. sorban az interruptok, míg a 155. sorban az SPI2 interfész engedélyezése történik.

A digitális-analóg átalakítás megvalósítása

A minták analóg jellé (hangjellé) alakítását a digitális-analóg átalakító (D/A konverter) végzi. Mivel ez az egység csak 12 bites egész számokat képes kezelni, ezért az eredetileg előjeles törtszámok formájában kalkulált mintáinkat ilyen formátumúra kell alakítanunk. A 16 bites üzenet fennmaradó 4 bit (15-12. bitek) a D/A konverter vezérlésére szolgál.

Az átalakítandó hangminta a w0 munkaregiszterben tárolódik. Célunk, hogy az átalakító az eredetileg 16 bites számérték legértékesebb 12 bitjét konvertálja folytonos jellé. Ehhez a w0-ban tárolt minta értékét 4 bittel jobbra toljuk, aminek hatására a 15-12. bitek 0 értéket vesznek fel. Az átalakítandó adatot tartalmazó w0 regiszterhez 0x0800 értéket (a w1 tartalma) hozzáadva egész szám formátumúvá alakítjuk a mintát. A D/A konverter vezérlő bitjeinek nullázását a 0FFF számmal történő maszkolással végezzük (ÉS műveletet végzünk a mintaérték és a 0FFF maszk között).

3000 hexadecimális értéket adva a mintát tartalmazó regiszterhez, elvégezzük az átalakító beállítását a következőképpen: a jelet az A csatornán szolgáltatassa 1 erősítéssel, és ne kapcsoljon alvó módba.

Ezzel előáll az SPI buszra továbbítandó üzenet. A PORTG 9. bitje, a slave select (SS). Ennek a lábnak az alacsony szintbe húzása engedélyezi a D/A konverter digitális interfészét, ekkor léptethető be az adat az SPI interfészen keresztül az átalakítóba.

```

251      DA_write:
252          asr        w0, #4, w0
253          mov        #0x0800, w1
254          add        w0, w1, w0
255          mov        #0x0fff, w1
256          and        w0, w1, w0
257          mov        #0x3000, w1
258          add        w0, w1, w0
259          bclr       PORTG, #SS
260          mov        w0, SPI2BUF
261          return

```

16. ábra A D/A konverziós szubrutin megvalósítása

A hangminták kalkulálása

A hangok generálását a legegyszerűbb DPW módszeren alapuló algoritmusok segítségével végeztem, melyek a 11. (a) ábra alapján készültek, vagyis a triviális fűrészjel mintáinak négyzetre emeléséből és azok differenciálásából származtatom a kimeneti, sávkorlátozott fűrészjelet.

A triviális fűrészjel egyszerűen előállítható egy számláló (ún. modulo számláló vagy counter) növelésével a következőképpen: kiszámítjuk a rendszer mintavételi időközét ($T=1/f_s=1/44100$ [s]) és azt, hogy hány mintából áll egy periódusnyi előállítani kívánt jel. Ez utóbbihoz szükségünk van a kívánt hang periódusidejére, ami a frekvenciájának reciprokából adódik. A kívánt hang periódusidejének és a rendszer mintavételi frekvenciájának hányadosából megkapjuk az említett egy periódusra eső mintaszámot (n). A munkaregiszterekben tárolható maximális számértéket (max) elosztva n -nel, megkapjuk, hogy mennyivel kell növelni a számlálót minden beérkező interrupt esetén:

$$növekmény = \frac{max}{n}$$

A kalkulált mintákat négyzetre emelve a 16. ábrához hasonló, szakaszosan parabolikus hullám adódik.

Differenciálva a mintákat, pl. a 8. fejezetben említett egyszerű, $H_z=1-z^{-1}/2$ átviteli függvényű FIR differenciáló segítségével a triviálisához képest csökkentett átlapolódású fűrészjelet kapunk.

```

DPW_init:
    mov #0x0093, w3    ;ez a lepeskoz (kvantulasi lepcso)
    mov #0, w10        ;ezt noveli a szamlalo
    mov #0, w13        ;elozo minta tarolo
    return

;DPW
;modulo counter
    add w10,w3,w10    ;a trivialis fureszjel kovetkezo mintaja
    mov w10,w4        ;a negyzetre emeles miatt a w4-be
;negyzetre emeles
    mpy w4*w4, A      ;A=w4*w4
    sac A, w8         ;az eredmenyt az akkumulatorbol a w8 regiszterbe
;differentiatio
    sub w8, w13, w0   ;kulonbsegkepzo
    mov w8, w13       ;a kovetkezo differentiatashoz elmentjuk
    lac w0, A         ;akkumulatorba toltes
    sftac A, #6       ;eltolas
    sac A, w0         ;kimeneti minta a w0-ba

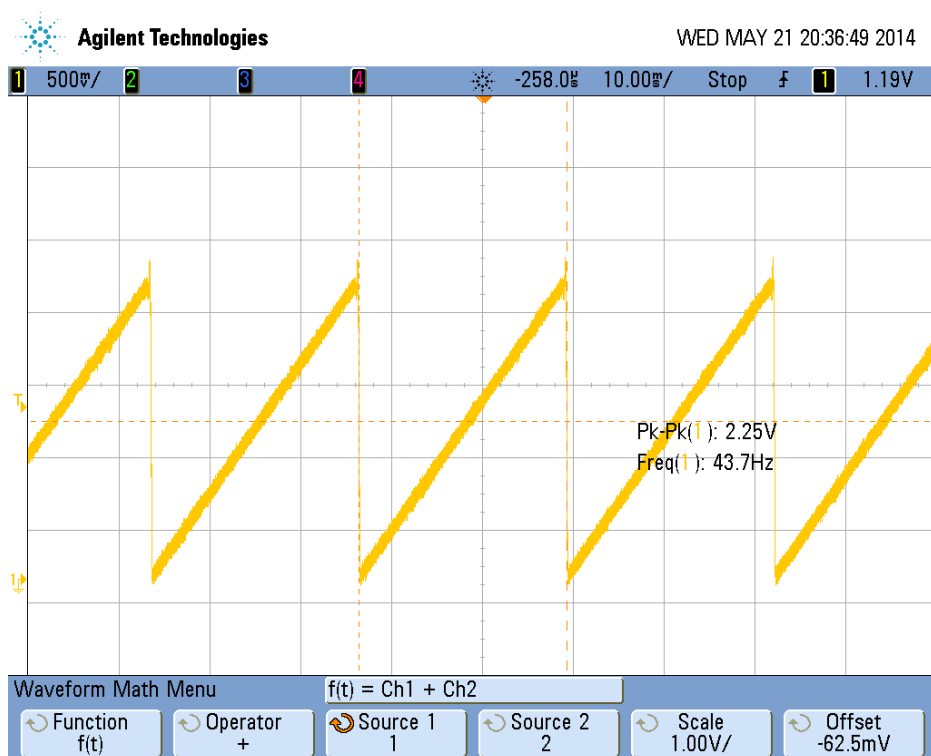
```

17. ábra A DPW algoritmus megvalósítása dsPIC33F-fel

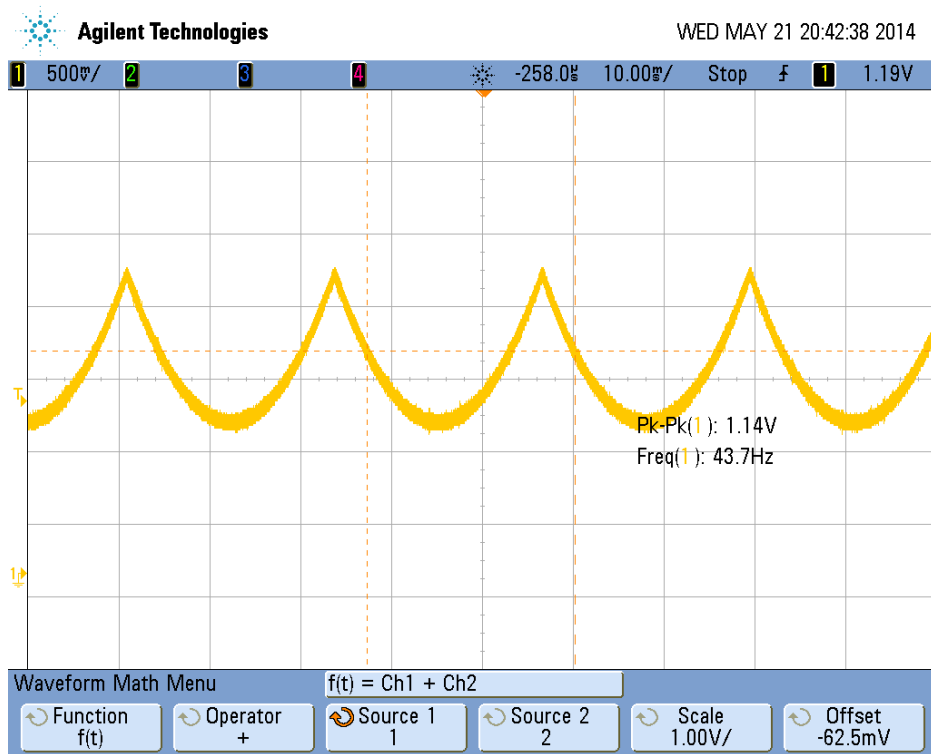
A 17. ábrán látható algoritmus első része az inicializálást hajtja végre, a *DPW_init* címkétől, illetve memóriacímtől kezdve. A lépésközt (a növekményt) a w3 munkaregiszter tárolja, az előállítandó hangnak megfelelően. A w10 regiszterben képződik a számlálás eredménye – minden interrupt vagyis hangminta generálása során 93 hexadecimális értékkel növekszik az értéke, a példában éppen az F1 hangnak megfelelően. A differenciáló számára az előző mintát a w13 tárolja.

A modulo counter egy összegzés, mely kiszámítja a triviális fűrészjelet. Mivel a parabolákat létrehozó négyzetre emelés szorzásra vezethető vissza, a processzor pedig csak a w4-w7 regiszterek értékeit tudja összeszorozni, ezért az előző utasítás által kalkulált triviális fűrészjel mintát a w4 regiszterbe helyezzük. A szorzat az A jelű, 40 bit tárolására képes akkumulátorban képződik, így az eredményt csonkolással a w8 regiszterbe helyezzük, mivel az ezt követő differenciáló kivonást végző sub utasítása csak munkaregisztereket tud kivonni egymásból. (Megjegyzés: az akkumulátorok is kivonhatók egymásból, így elvileg lehetséges lenne a w13 értékének a B akkumulátorba való töltése is, a kivonáshoz.) A kivonás eredményét visszatöltjük az akkumulátorba, és eltoljuk olyanképpen, hogy a D/A konverter számára a megfelelő formában álljon elő, majd a w0 regiszterben átadjuk számára, analóg jellé (hanggá) alakításhoz.

Az algoritmus az előállított jel minőségét tekintve továbbfejleszthető a 8. fejezetben bemutatott módokon, illetve megfelelő amplitúdó skálázást alkalmazva, a jelszintek tekintetében szabványos jel állítható elő, amely megfelelhet például a szintetizátorok és más stúdióeszközök összekapcsolhatóságát biztosító Musical Instrument Digital Interface (MIDI) szabványban foglaltaknak. Ezen szabványok ismertetése és alkalmazása azonban nem képezte célját ennek a dolgozatnak, itt csupán a jelgenerálás alapelvét kívántam bemutatni.



18. ábra A triviális fűrészel (F1 zenei hang)



19. ábra A triviális fűrészel mintáinak négyzetre emelésével kapott szakaszosan parabolikus hullám (F1 zenei hang esetén)



20. ábra A triviális (fent) és a sávkorlátozott fűrészel (lent) összehasonlítása. A sávkorlátozás megtörténtét az ugrások meredekségének változása igazolja, a további eltérésért az analóg szűrő a felelős

ÖSSZEFOGLALÓ

Dolgozatomban a jól ismert hangszintézis módszerek ismertetése után bemutattam az egyik legegyszerűbb módszer, a DPW algoritmus alkalmazását dsPIC mikrokontrolleres környezet esetén.

A lehetséges megoldások vizsgálata során megállapítottam, hogy a teljesen sávkorlátozott megoldások bár jó minőségű eredményt szolgáltatnak, magas számításigényük miatt nem használhatók eredményesen jelen feltételek mellett.

A BLIT módszert a valós idejű integrálás szükségessége teszi kevésbé vonzóvá, míg a BLEP és a minBLEP ugyan megoldja a sávkorlátozott impulzussorozat integrálási problémáját, de szintén memóriaigényes eljárás, bár kevesebb tárterületet igényel, mint a hullámtábla szintézis.

A polyBLEP és a DPW alacsony számítási költségű, viszonylag jó minőségű, real-time megoldások, ami megfelelőnek bizonyul a mikrokontrolleres megvalósításra.

A dolgozat második felében a legegyszerűbb DPW algoritmus segítségével ismertettem a mikrokontrolleres hanggenerálás módját.

Jobb minőségű eredmény az átlapolódási jelenség csökkentésével és megfelelő amplitúdó skálázás segítségével kapható, ezen szempontok figyelembe vételével tökéletesíthető a módszer.

FORRÁSOK

- [1] <http://hu.wikipedia.org/wiki/Szintetiz%C3%A1tor>
- [2] Stilson, Tim and Smith, Julius (1996) „Alias-Free Digital Synthesis of Classic Analog Waveforms”, Music Department, Stanford University
- [3] Brandt, Eli: „Hard Sync Without Aliasing”, (2001) School of Computer Science, Carnegie Mellon University, Proc. ICMC 2001.
- [4] Ladányi Gábor: „Gépek állapotfelügyelete (I.) Rezgésdiagnosztikára alapozott mérések” Műszaki Magazin, 2002. májusi száma, 111-112. oldal.
- [5] Välimäki, Vesa: „Discrete-Time Synthesis of the Sawtooth Waveform With Reduced Aliasing” (2005), IEEE Signal Processing Letters. Vol. 12, No. 3, March 2005
- [6] C. Mahrer, Robert: „Wavetable Synthesis Strategies for Mobile Devices, Department of Electrical and Computer Engineering” (2005), J. Audio Eng. Soc., vol. 53, no. 3, pp. 205-213, 2005.
- [7] Huovilainen, Antti and Välimäki, Vesa: „New Approaches to Digital Subtractive Synthesis” (2005) Proceedings of the International Computer Music Conference; 399-402
- [8] Ambrits Dániel: „Hatékony oszcillátor algoritmusok analóg szintetizátorok modellezésére” (2013), TDK dolgozat, Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar Méréstechnika és Információs Rendszerek Tanszék
- [9] Prof. Välimäki, Vesa „Virtual Analog Modeling Research at Aalto University”, Department of Signal Processing and Acoustics, Aalto University (Espoo, FINLAND)
- [10] Ambits Dániel: Analóg szintetizátor modellezése különböző oszcillátor algoritmusokkal, szakdolgozat, Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar Méréstechnika és Információs Rendszerek Tanszék
- [11] <http://www.musica.ed.ac.uk/wp-content/uploads/Manual Upload/Valimaki-Virtual-analog.pdf>

Az internetes források 2014. 04. 13-án lettek összegyűjtve, és voltak elérhetők a jelenlegi helyükön, és a hivatkozott formátumban.