Képkódolási eljárások

Alulmintavételezés (subsampling)

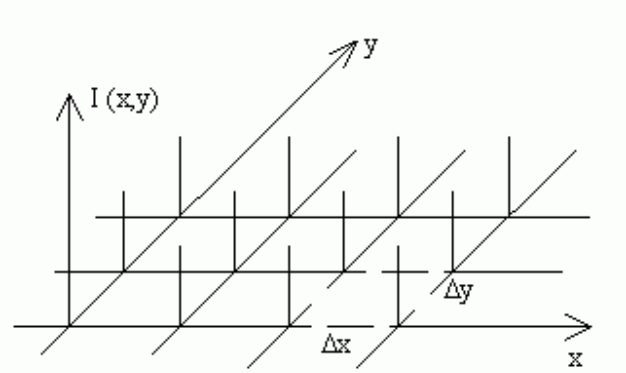
Egy elektronikus jelet (amely lehet hang időfüggvény vagy képjel) a Shannon-tétel értelmében lehet hibamentesen digitalizálni. A digitális jel egyrészt diszkrét időben (az x-tengely mentén), másrészt diszkrét amplitúdóban (az y-tengely mentén) is. Előbbit mintavételezésnek, utóbbit kvantálásnak nevezzük. Ha a jelünk nem egy dimenziós, mint a hang, hanem kétdimenziós, akkor a mintavételezést is két dimenzióban kell végezni. Hangjel esetén ez megfelel az időfüggvény másodpercenkénti mintavételezésével, pld. a CD a jól ismert 44100 Hz-es mintavételi frekvenciát alkalmazza csatornánként: másodpercenként 44100 darab mintát vesz a jelből. A Shannon-tétel kimondja, hogy a mintavételi frekvencia (a megmintázás gyakorisága) legalább kétszeres kell legyen a mintázandó jel sávszélességének. Ha ez igaz, akkor egy aluláteresztő szűrővel (interpoláló szűrő) a jel visszaalakítható. CD esetén ez az érték 22050 Hz, azaz a bemenő jeleket mindig egy aluláteresztő szűrőn kell átvezetni, mielőtt digitalizálnánk. EZ ideális esetben lehet, hogy pont a mintavételi frekvencia felénél vág, a gyakorlatban azonban kell „helyet hagyni" a szűrő lefutásának is, így a CD esetén 20 kHz-nél van a vágás. Ezt az aluláteresztő szűrőt nevezzük AAF-nek (Anti Aliasing Filter), melynek szerepe, hogy az aliasing-jelenséget kizárja. Ez akkor lép fel, ha a mintavevő-tartóra ráengedünk a mintavételi frekvencia felénél nagyobb komponenseket is, amelyik így spektrális átlapolódásba kerülnek a hasznos jel periodikus spektrumával, és megjelennek olyan „ál" komponensek, amelyek az eredetiben nem voltak benne.

Képek mintavételezése térbeli művelet, azaz egy analóg képet ha „beszkennelünk", akkor azt sorokra és oszlopokra fogjuk felbontani, a képelemeket pedig pixeleknek nevezzük. Egy kétdimenziós kép tehát áll valamennyi függőleges és vízszintes képpontból egy képzeletbeli rács keresztezési pontjaiban (a leírás történhet egy kétdimenziós mátrix-al, ahol a pontok helyén lévő szám a pixel értékét jelöli). A fenti hangpéldában a „felbontás" Δt értéke 1/44100 másodperc, kép esetén azonban Δx és Δy helyettesíti (és elvileg mm-ben lehet megadni).

Alulmintavételezéskor azt a hibát követjük el, hogy az átviendő jelet túl alacsony mintavételi frekvenciával mintavételezzük. Ha a CD jelét 44100 Hz helyett csak 20000 Hz-el mintázzuk meg, akkor két probléma lehet: vagy alias fordul elő a 10 és 20 kHz közötti komponensek okán; vagy pedig ezt elkerülendő az AAF szűrő vágási frekvenciáját kell lecsökkenteni 10 kHz-re. Ekkor azonban az átviendő sáv felső tartománya kiesik. Látható, hogy a mintavételi frekvencia egész számmal való leosztása ugyanekkora csökkenést okoz a szűrő felső határfrekvenciájában, cserébe az átviendő adatmennyiség is ennyi részére csökken!

Ez kép esetében sincs másképp, ha a minták sűrűségét csökkentjük a térirányokban, pontosan ezt a hatást érjük el. Az új rácsháló lehet egyszerű alhalmaza az eredetinek, pld. ha egyszerűen minden másodikat eldobunk. De lehet ennél bonyolultabb és új elrendezésű is. Ez a legegyszerűbb módja az adatcsökkentésnek (tömörítésnek). A mintavételező blokk megfelezése egy kép esetén felére csökkenti az AAF szűrő határfrekvenciáját és mindkét térirányban felezi a mintamennyiséget (azaz egy pixelből egy 2*2-es pixelblokk jön létre, ez összesen négyszeres adatnyereség két dimenzióban). Sajnos, az AAF szűrő (és részben a visszaalakító interpoláló szűrő a vevőben) a maga torzítását beleviszi a rendszerbe, ez azonban kevésbé zavaró, mint az

alias lenne. (Megjegyzés: a hangtechnikában egyedül a Super Audio CD által alkalmazott DSD rögzítési elv nem alkalmaz PCM bitszavakat és ott nincs szükség a bementi AAF szűrőre, sőt a vevő oldali is más.)



A két dimenziós kép mintavételi pontjai (pixelek) és azok intenzitása

Figyelem, ha ezt tesszük egy 200*200 pixeles képpel, az nem azt jelenti, hogy a méretre csökken le, hanem azt, hogy a 200*200 pixel méretűben nagyobb pixelek keletkeznek. Ha például az eredetileg 200*200 pixeles képben az adatcsökkentés faktora 16, akkor minden pixelből egy 16*16 szoros „nagy pixel" keletkezik a 200*200-as méretű képen. Ebből tehát 12 darab 16 pixeles négyzet keletkezik, majd mivel ez csak 192 pixelnek felel meg, a maradék 8 nem szenved semmiféle átalakítást (és ez látható lesz a kép alján). Egy kis pixel világosságértéke ekkor „öröklődik" a másik 15-re nézve, így amelyikeket elhagytuk (összesen 16*16-1 darabot) az egyszerűn megkapja annak értékét.

PCM kódolás

A PCM jelentése Pulse Code Modulation. Ilyenkor a leírandó információ (ami lehet hangminta az időfüggvényben vagy egy pixel számértéke egy képen) lehetséges értékeit kantáljuk és a kvantálási lépcsőknek megfelelően bináris kódszavakba kódoljuk. A képtechnikában a legkisebb információegység a pixel (képpont), melyből az egész kép felépül. Fekete-fehér (vagy legalábbis monokróm) képnél egy pixel csak világosságértékkel rendelkezik. Például 8 bites kvantálás esetén 2^8-on, 256 darab értéket tudunk megkülönböztetni a fehér és a fekete között a szürkeskálában (0 és 255 lehetnek az értékek). Ezeket a számokat aztán nyolcbites kódszavakba (szimbólumokba) alakítjuk és visszük át. Színes kép esetén a három alapszín komponenssel is dolgoznunk kell, az R, G és B jeleket egyenként le kell írni egy-egy kódszóval. Látható, hogy egy 800*600 pixeles monokróm kép mérete is 8 bites kvantálás mellett 3840000 bitet tesz ki, csak a pixelekből. Ha kevesebb bitet használunk, akkor adatmennyiséget csökkentünk (tömörítünk) de rontunk a minőségen. Ha csökkentjük a bitszámot, akkor a példában lévő nyolcbites kódszónak a legkisebb helyiértékű (LSB: Least Significant Bit) bitjét, bitjeit hagyjuk el (nullára állítjuk).
Ha a nyolcból hetet csinálunk, akkor már csak fele annyi kvantálási lépcsőnk van, azaz a felbontás máris a felére csökkent a színskálán! Ennek egyenes következménye, hogy bizonyos folytonosabb kontúr-átmenetek eltűnnek, és helyettük újak, élesebbek jelennek meg. Ezután az új kvantálási szintet beállítjuk vagy valamelyik szélsőértékre vagy két érték között félútra. Előbbit *mid-step*, utóbbit *mid-riser* elvűnek nevezzük. Utóbbi a gyakoribb és jobb eljárás. A különbség

jól érzékelhető, ha a bitszámot a legszélsőségesebb esetre, 1-re csökkentjük. Ilyenkor monokróm képet kapunk, hiszen egy pixel világosságértéke vagy 0 vagy 1 lehet, azaz összesen kétféle értéket vehet fel. Mid-step esetben akkor egy pixel vagy fekete (0) vagy fehér (1), de szubjektíven kellemesebb a kép, ha a mid-riser kódolóval az értékek a világos-szürke (1) és a sötét-szürke (0) lehetnek. Ezek a világosságértékek pedig a félút felénél vannak, azaz a szürkeskála egynegyedénél és háromnegyedénél (0=fekete, 0,25=sötétszürke, 0,5=szürke, 0,75=világosszürke, 1 = fehér).
Egy nyolcbites kép hatra csökkentése még nem látható igazán. A részletek később kezdenek eltűnni és fals kontúrok megjelenni. Ez tipikusan kvantálási hiba, amely (nem lineáris) torzításhoz vezet. Mit lehet tenni a minőség javításának érdekében?

Dither

A dither jelentése zajmoduláció. A magyar elnevezés azt sugallja, hogy egy hasznos jelhez valamilyen úton zajt adunk hozzá, és mint minden moduláció, ez végeredményben előnyös lesz. A hasznos jel lehet éppúgy hang, ahogy képinformáció is.
A kiindulási pont, hogy az emberi érzékszervek (a szem és a fül) különböző módon érzékenyek a hibákra. Objektíven mérhető „rossz" kép vagy hang is lehet szubjektíven jobb, mint az, amelynél az objektív mérőszámok jobbat mutatnak. Érzékszerveink tipikusan érzékenyebbek a torzításokra. Ezek közül a legismertebb és leggyakoribb a nem lineáris torzítás. Hallásunk és látásunk is rosszabbnak érzékeli az ilyen jelet, mint az olyat, ahol nincs nem lineáris torzítás, de a zaj esetleg nagyobb.
Hiba az átvitt információban két módon keletkezhet: egyrészt durva kvantálási hibával és/vagy az átviteli csatorna hibázása folytán. A kvantálási hiba torzító hatása jól érzékelhető durva kvantálási lépcsők melletti szinuszos átvitelnél, ahol a kvantáló erőteljesen „elnégyszögesíti" a jelet. Az ilyen négyszög-típusú jeleknél pedig a felharmonikus megjelennek az alapfrekvencia mellett – ez a nem lineáris torzítás. A digitális csatorna hibáit az átviteli valószínűségek (csatornakapacitás) és a bithibaarány (BER: bit error ratio) jellemzi. Utóbbi egy 0-1 közötti szám, vagy százalékosan is megadható. Ha a csatornahiba (channel error) pld. 0,5 (azaz 50%), akkor átlagosan minden második bitet elront (és az ellenkezőjére változtat). A valóságban már egy 0,05%-os hibát is nagynak tekinthetünk, ami pedig csak minden kétszázadikat rontja el átlagosan. Igazi alkalmazásokban 10 sokadik negatív hatványa a megszokott bithibaarány (10^{-6} - 10^{-12}). A csatornahibát mesterségesen is szimulálhatjuk memóriamentes, szimmetrikus algoritmussal. Az alkalmazások persze használnak különböző hibavédelmet, hibajavító kódolást is, amely növelü a hatásfokot és az átvitel tulajdonságait.
A dither elve, hogy az adó (kódoló) oldalon egy álvéletlen (pszeudo-random) zajjelet adnak a hasznos jelhez a kvantálás előtt. Az átvitel után a dekódolás-visszaalakítás végeztével pedig ugyanezt a jelet kivonják belőle. Természetesen, a zaj hozzáadása után a jel-zaj-viszony leromlik. Az álvéletlen zaj ebben az esetben egy bitsorozat, több bites szimbólumok véletlen sorozata, amelyet tipikusan visszacsatolt shift-regiszterekkel állítunk elő. Erre léteznek jó algoritmusok. Magát a zajjelet nem kell átvinni, ha az alábbi feltételek teljesülnek: egyrészt ugyanazt a léptetőregisztert használja a vevő, mint az adó. Másodszor, mivel ezeknek szükséges egy előzetes inicializáló feltöltése, az is meg kell egyezzen a két oldalon. Ekkor garantált, hogy bitszinkronban azonos „véletlen" sorozatot lehet létrehozni.
A hasznos jelhez hozzáadott zaj „átlagoló" hatású. Tény, hogy lerontjuk szándékosan a jel-zaj-viszonyt, ugyanakkor szubjektíven jobb eredményt fogunk kapni az átvitel végén. EZ a

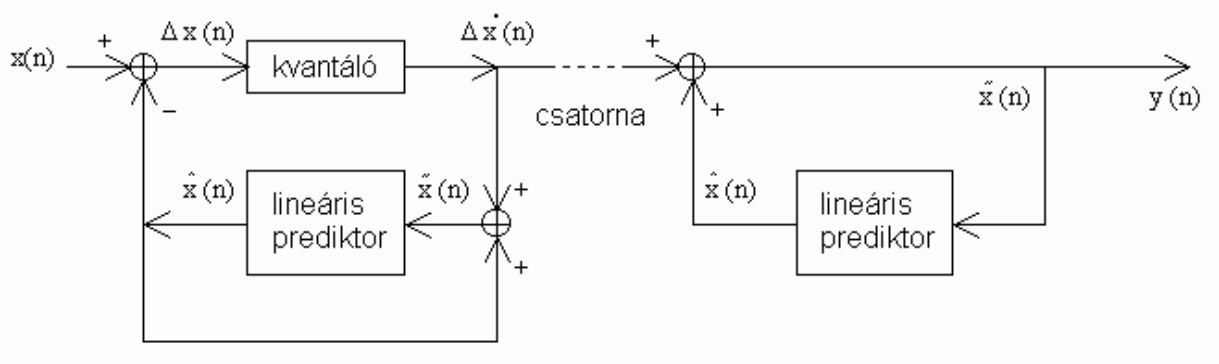
hozzáadott zaj ugyanis spektrálisan is kifejti hatását: szélessávú és nem túl nagy amplitúdójú, elkeni a spektrumot és annak „szőrösödését” okozza Ugyanakkor az élés nem lineáris torzítás komponenseit, a felharmonikusokat eltünteti, csökkenti – ezáltal jobb érzetet kelt a megfigyelőben.
Hibamentes, tökéletes átvitelnél ennek nincs jelentősége, és felesleges. Azonban a valóságban a bitsebesség csökkentés (pld. a kvantálási lépcsők bitszámának csökkentése) vagy a csatorna esetleg az erősítők, berendezések okozta nem lineáris torzítása esetén ez hasznos eljárás.

A mintavételezés és a kvantálás együtt jelenti a jelek digitalizálását. A valóságban a kettő együtt jár agy A/D-átalakítóban, amely áll egy mintavevő-tartóból és egy kvantálóból (kódolóból). A Shannon-tétel kimondja, hogy pusztán a mintavételezés nem visz semmilyen hibát a digitális jelbe. Amennyiben a tétel igaz, egy aluláteresztő szűrővel a jel *tökéletesen* (hibamentesen) alakítható vissza analóggá. Ekkor azonban a minták értéke (az amplitúdó) még mindig végtelen tizedestörttel van leírva, tehát semmiféle átviteli nyereséget nem értünk el. Kénytelenek vagyunk tehát kvantálni, és folytonos az amplitudóértékeket diszkrét értékekkel leírni. Ez pedig kerekítés, mely kerekítési hiba (ennek négyzetes várható értékét nevezzük kvantálási zajnak) örökre elveszik és visszaalakítás után is megmarad. Elveszítjük a kvantálási lépcső felének megfelelő finom változásokat, ezért cél, hogy minél nagyobb bitszámmal (azaz minél több lépcsőre osszuk fel) írjuk le az amplitúdó értékeket. A digitális jel tehát, amelyet nem vetettünk alá semmiféle tömörítésnek, két paraméterrel írható le: a mintavételi frekvenciával (amely megszabja az átvihető jel sávszélességét) és a bitszámmal (amely megmondja mekkora a dinamikatartomány, a legkisebb és a legnagyobb jelérték hányadosa). Ezt nevezzük tömörítetlen (nyers, angolul: raw) PCM adatnak. A PCM kódolású pixelekből álló kép a közismert *bitmap* (BMP).
A fentiekből látható, hogy egyszerű módon lehet adatmennyiséget csökkenteni, azaz tömöríteni. Egyrészt a mintavételi frekvencia, másrészt a bitszámok csökkentésével. Előbbi az átvihető jel sávszélességét fogja lerontani (a nagyfrekvenciákat kiszűri a módosított AAF szűrő), és ezáltal a térbeli felbontást, utóbbi pedig a megjeleníthető árnyalatokat csökkenti. Ezek a módszerek rendkívül egyszerűek, ugyanakkor erőteljes minőségromlást eredményeznek (noha a dither kvantálási hibán némileg javíthat szubjektíven). Éppen ezért a fenti eljárások a gyakorlatban elvi jelentőségűek csak, lényegesen nagyobb hatásfokú tömörítési módszereket lehet kidolgozni (gondoljunk csak a hangtechnikában alkalmazott MP3-ra vagy a képtechnikában a JPG kódolású képekre).

DPCM kódoló

A PCM kódolók nem használják ki azt a tényt, hogy „normális” jelek esetén a szomszédos minták egymással korreláltak. Egy beszédmintában vagy zeneszámban az egymás melletti minták közötti távolság 1/44100 másodperc. Egy állóképen a szomszédos pixelek közötti távolság a mm-nél is kisebb. Aligha változik meg a szomszédos minta értéke ugrásszerűen, az nagyon hasonlítani fog a környezetében található mintákra. Ha kihasználjuk ezt a tényt, akkor arra számítunk, hogy nem fog olyan jel érkezni az átvitel során, amelyre ez nem igaz. Feltételezzük, hogy nem célunk átvinni egy olyan ábrát, amely fekete-fehér sakktáblát ábrázol pixelenként…
Mozgó kép esetén ez a korreláció nem csak térben, hanem időben is megfigyelhető: a képen belüli pixelek szomszédsága éppúgy hasonló (intra-frame), mint az egymás utáni képek között (inter-frame).

A DPCM kódoló (Differential PCM) a differencia, a különbség átvitelén és tárolásán alapul. Ilyenkor nem az átviendő minta (pixel) értékét kódoljuk, hanem csak egy ún. hibajelet. Ez a hibajel egy predikció, előrejelzéses becslés eredménye. Annyiban hasonlít a PCM-re, hogy a bementi jelét adott bitszámú kódszavakba kódolja. A különbség pedig, hogy az aktuális (átvitelre szánt) minta értékét megbecsüljük az őt megelőzőkből (vagy környezetében lévőkből). Minél több, távolabbi mintát használunk fel a becslésre, annál jobb lesz a minőség és annál több számítási időt fog igénybe venni. Kvantálásra és átvitelre ekkor a „predikciós hiba" kerül, nem pedig az adott minta. A predikció készülhet eleve predikált mintá(k)ból és/vagy teljes egészében átvitt mintá(k)ból. A vevő oldalon rendelkezésre állnak azok a már korábban átvitt és helyreállított minták, amiből a predikció az aktuális mintára készült, valamint az aktuális hiba – ezek összegéből a minta helyreállítható.



A DPCM kódoló (egy dimenziós) blokkvázlata.

A működést az alábbi egyenletek írják le:

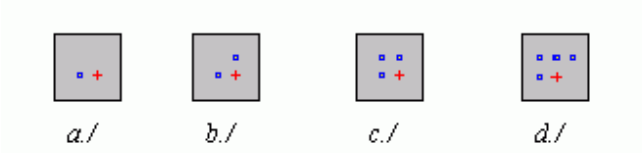
$$\Delta x(n) = x(n) - \hat{x}(n)$$
$$\Delta x^*(n) = \Delta x(n) + q(n)$$
$$y(n) = \hat{x}(n) + \Delta x^*(n)$$

Továbbá, lineáris predikció esetén:

$$\hat{x}(n) = \sum_{j=1}^{N} h_j \cdot \tilde{x}(n-j)$$

A legegyszerűbb esetben a becslés legyen egyenlő az előző pixel értékével (a tőle balra esővel). Ha annak világosságtartalma 200, feltételezzük, hogy a következő is valami ehhez közeli érték lesz, tehát mindenféle becslő-algoritmus nélkül legyen az is 200. A kódoló megvizsgálja az adott mintát, és annak tényleges értékéhez képest kiszámítja a hibát. Ha a következő minta tényleges értéke 202, akkor 202-200=2 lesz a becslési hiba. Ezt a kettes értéket pedig két biten át lehet vinni, ellentétben a nyolcbites tényleges mintaértékkel. A vevőben a helyreállítás egyszerű: a predikciós algoritmust az is ismeri, tehát tudja, hogy az előző mintából kell a hibát összeadni

(vagy kivonni), így a beérkező „2” és a saját predikciójából adódó „200” értékből egyszerű összeadással a 202-t helyre tudja állítani.
A predikció nem mindig ilyen egyszerű, és különböző módon valósítható meg (pld. a Max-Lloyd kvantáló). DPSM kódolók általában N-ed rendű lineáris predikciót alkalmaznak egy ún. optimális súlyozó vektorral számolva. Ez a vektor egy számhalmaz, amelyben lévő együtthatókkal meg kell szorozni a szomszédos pixelek értékét és átlagképzés után kerül ki a becsült szám.
DPCM esetén két hibával kell számolnunk a rekonstrukciónál. Ha a bemenő jel túl gyorsan (nagyon) változik, a kvantáló nem fogja tudni követni azt. Ha a fenti példát nézve egy állandó 2 bites kvantálás használunk a hibára, de annak értéke átlépi a hármat (0,1,2 és 3 kódolható két biten), akkor is három fog átmenni, ha a valóságos hibaérték 12 – ez pedig megjelenítési hibához vezet. Ehhez ugyanis négy bitre lenne szükség. Ha viszont négy bitet használunk mindig, akkor az átviteli nyereség fog lecsökkeni. Másrészt, ha a bement túl lassan változik, szemcsés-zaj jelenik meg a legkisebb kvantálási lépcső durvasága miatt. A csatornahibánál meg kell jegyeznünk, hogy ellentétben a PCM kódolással, az esetleges bithibák itt „öröklődnek” és magukkal viszik a predikciós folyamatban pixelről-pixelre (error propagation): nem csupán az adott megbecsült pixel romlik el, hanem azok is, amelyeket majd ebből fogunk predikálni. Ez a képen erősebb „foltosodásban” mutatkozik meg, ellentétben a PCM képek apró pixelhiba-pöttyeivel.



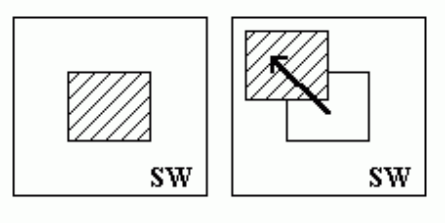
Pixelek becslése egy képen: a piros kereszt a becsült minta, a kék négyzetek a becslésben részt vevő pixelek.

Kétdimenziós állóképnél a pixel beolvasást a bal felső sarokból kezdjük és sorfolytonosan végezzük. Mozgóképeknél mozgásbecslést (is) végzünk a képek között.

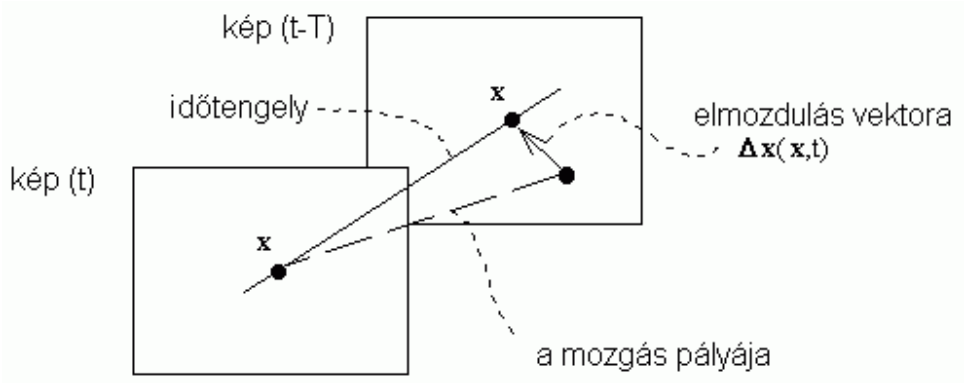
Mozgásbecslés

Mozgóképek esetén kihasználhatjuk a képek közötti (inter-frame) hasonlóságot is. Videójeleknél, ahol 25-30 kép van másodpercenként nagy a hasonlóság az egymás után 1/25 másodperccel érkező képek között.
Mozgásbecslésnél – ahogy a neve is mutatja – megpróbáljuk a képen lévő mozgást képről-képre (frame-by-frame) követni. A DPCM ötletét kihasználva, a kódoló a mozgásbecsléssel predikálja az aktuális képet egy referencia képből és szintén a predikciós hibát viszi át. A referencia kép lehet az előző vagy több megelőző kép egyszerre, sőt jövőbeli kép is. A legegyszerűbb mód itt is az előző egyetlen képből történő becslés.
A mozgásbecslés végeredménye egy mozgás-vektor halmaz, mely leírja a mozgást egyik képről a másikra. Az aktuális kép nem átlapolódó blokkokra van felosztva, melyek csak a világosságértékeket tartalmazzák. Egy vektor kerül kiszámításra minden blokk számára a „keresési eljárás” során. Minden blokk elmozdul egy kereső ablak felett a referencia képen, és az algoritmus megpróbálja megtalálni a legjobb egyezést egy ún. költség-függvény alapján. Ez a távolság-eltérést méri az aktuális és az eltolt blokk között. Gyakorlatban ez megfelel az abszolút-különbségek összegének vagy a négyzetes hiba összegnek (előbbi a gyorsabb, utóbbi a

pontosabb). A mozgásvektor (eltolási vektor) definíció szerint az eltolódás mértéke pixelben mérve a költség-függvény minimumánál.
Egy blokk tipikusan 8*8 vagy 16*16 pixeles. Kisebb blokk esetén több mozgás adat lesz, de pontosabb vektorok. Ugyanakkor, nagyobb számításigény és kisebb siker nagy mozgások esetén (melyek túlnyúlhatnak a blokkon). Általában a maximális megengedett eltolás, amely meghatározza a kereső ablak méretét is: (-8, +7), (-16, +15) vagy (-32, +31) mindkét irányban. A keresőablak megnövelése ugrásszerűen növeli a számításigényt.
Minden pixelblokk a referencia kép megfelelő blokkjából kerül becslésre. Minél pontosabb a vektor, annál jobb a becslés. A becsléshiba mellett azonban a mozgásvektort is át kell vinni. A dekódoló oldalon csak mozgás kompenzáció van, tehát ez kevésbé bonyolult, mint a kódoló. A kódolás sok időt vesz igénybe, így egyszerű esetben elég ha a mozgásbecslés egyenlő az aktuális és a referencia kép különbségével bármiféle tényleges mozgáskompenzáció nélkül.
A mozgásbecslés legegyszerűbb formája a nyers erő módszere (brute force), amelyet neveznek még teljes keresésnek is. Ekkor a legjobb egyezést találjuk meg, mert az összes lehetséges esetet végig néz az algoritmus. Ez globális, optimális minimumot talál, de nagy számításigényű. Ezért kitaláltak már több, heurisztikus módszert is, melyek csak lokális minimumot találnak meg, de sokkal gyorsabban. A „one at a time” módszer először az X-irányban keres minimumot, majd onnan indulva az Y-irányban is. Egy vektor esetében ez azonban nem feltétlenül lesz a globális optimum. Az N-lépéses keresésnél egy előre meghatározott (n-lépésből) álló folyamat során határozzák meg a költség-függvényt egy intervallum-felezős módszerrel.



A kereső ablak elhelyezkedése és elmozdulása a képen.



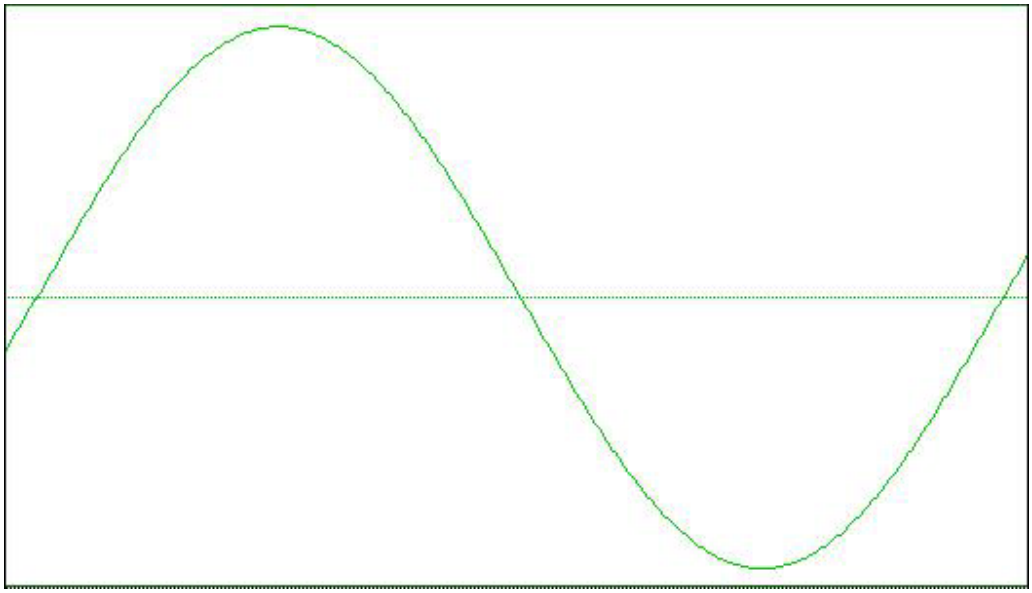
A mozgásbecslés egyik képről a másikra *T* idő elteltével.

A dither

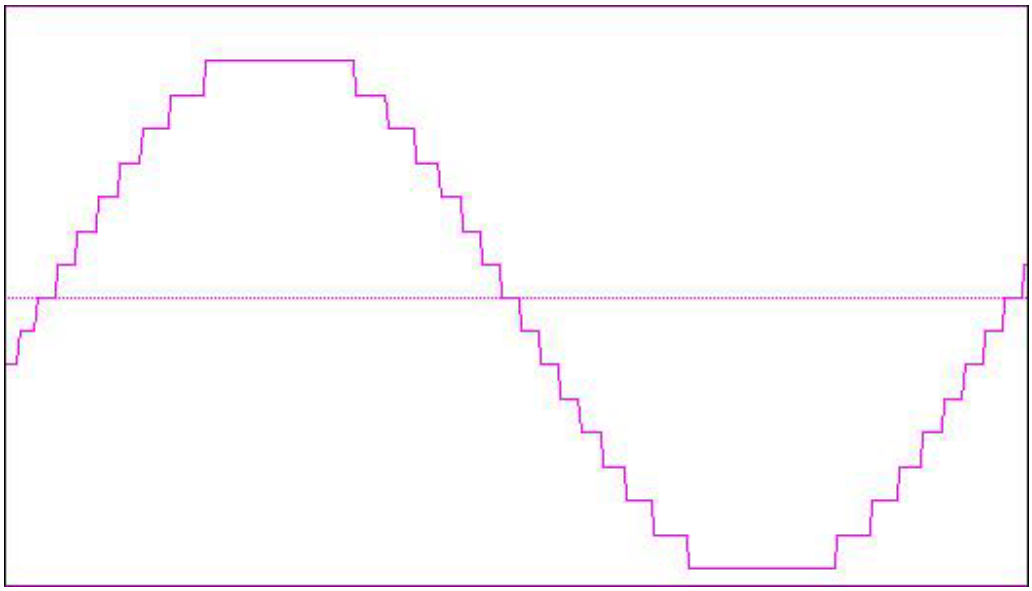
A PCM kódolás mindig fix bithosszúságú kódszavakat használ. A CD-nél megszokott 16 bit-hez tartozó 96 dB-es jel-zaj-viszony és dinamika nagyon jó, de ez csak akkor igaz, ha ki is használjuk mind a 16 bitet. Ezt azonban nem tesszük meg, nem hallgatunk állandóan csúcstól-csúcsig maximális amplitúdóval kivezérelt szinuszhullámot…sokkal inkább úgy próbáljuk „belőni" a kódolót, hogy az előforduló legnagyobb értékhez rendeljen 16 bitet, az idő többi részében a halk hangokhoz viszont kevesebbet. A „leghalkabb" elképzelhető szinuszhullám pld 1-bites, így nagyon messze lesz egy tényleges szinusz hullámformától: egyszerű négyszögjellé torzul. Minél kisebb az amplitúdó, annál nagyobb a relatív torzítás. Továbbá, amplitúdóértékek, melyek kisebbek egy bit szintjénél még csak nem is kerülnek rögzítésre.
Az audiotechnikában alkalmazott ditherzaj szintje kisebb, mint az LSB bit értéke. A hozzáadást a 16 bitre való kerekítés előtt végezzük. Hatása, hogy szétszórja a sok, rövididejű hibát a spektrumban, mint közönséges szélessávú zaj. Némileg javíthatjuk az eljárást pld. zajformálással (noise shaping), de ettől még az elv ugyanaz: a lehető legkevesebb zajt hozzáadni, ami elégséges.
Az egyik hatás tehát az alacsony frekvenciás tagok torzításának csökkentése, a másik pedig az idáig hallhatatlan részek „előbányászása". Azok a részek ugyanis az analóg jelben, melyek ezidáig nem voltak képesek átbillenteni az LSB bitet, a hozzáadott zaj hatására már esetleg igen (statisztikai valószínűséggel adható meg). A hallórendszerünk képes ezeket az újonnan előkerült jelrészleteket a háttérzajtól elválasztani.

Egy érdekes analógia: egyik szemünket behunyva tartsuk a nyitott elé közelre kezünket, ujjainkat széttárva. Nézzünk egy olyan nagyobb objektumra, melyből egyszerre csak egy részlet látszik az ujjaink között. Ha most gyorsan előre-hátra mozgatjuk a kezünket egy zavaró stroboszkóp hatás mellett azt vehetjük észre, hogy „látjuk" az egész tárgyat, noha egy adott pillanatban soha nem látszik az egész. Az agyunk képes összerakni" és egységben érzékelni azt. Egy kis zavarral számolnunk kell (hiszen nem elég hogy a kezünk egy része takarja a kilátást még ráadásul idegesítően kapkodjuk is a szemünk előtt), ugyanakkor a végeredmény sokkal jobb: az egész tárgyról – vagy ha a képernyő előtt ezzel a szöveggel csináljuk meg, akkor azt el tudjuk olvasni – megkapjuk az információt. A dither pont ezt teszi: egy kevésbé zavaró hatást ad a jelhez, mi által az jobban érzékelhetővé (láthatóvá, hallhatóvá) válik!

Alapjában, dither-re akkor van szükség, ha a bitszámot csökkentjük egy adott jel ábrázolásánál. Ha pld. a 16 bites szavakat 8 bitessé csökkentjük. Ilyenkor a csonkolás és a kerekítés helyett – amely harmonikus és intermodulációs torzítást okozhat – a hozzáadott zaj „szétkeni a hibát időben", mint szélessávú zaj. Hasonlóan, már egy analóg jel rögzítésekor is alkalmazhatunk dither zajt. Két 16 bites szám összeszorzásának eredménye egy 32 bites szó. Egyszerű lekerekítés helyett dither-t lehet alkalmazni. Ilyen szorzás előfordul mixelésnél (keverésnél) vagy erősítéskor, normalizáláskor. Utóbbinál különösen kell ügyelni arra, hogy a zajt és a torzítást ne normalizáljuk (minden egyes keverési lépcsőnél), csak a legvégén, amikor a kész anyagot akarjuk megfelelő hangerősségre hozni. A szoftverek ezt tudják, ezért automatikusan dithereznek, ha szükséges. Kellően magas bitszámnál (24 bit, 32 bites lebegőpontos) nem kell dither. Ekkora dinamikánál már rég a zajt kvantáljuk. Elég akkor a dither-t hívni, amikor a kész anyagot 16 bitesre átszámoljuk. Manapság a technológia lehetővé teszi a 24 bites átalakítók otthoni használatát is, ami kiküszöböli ezt a lépést (DVD Audio).

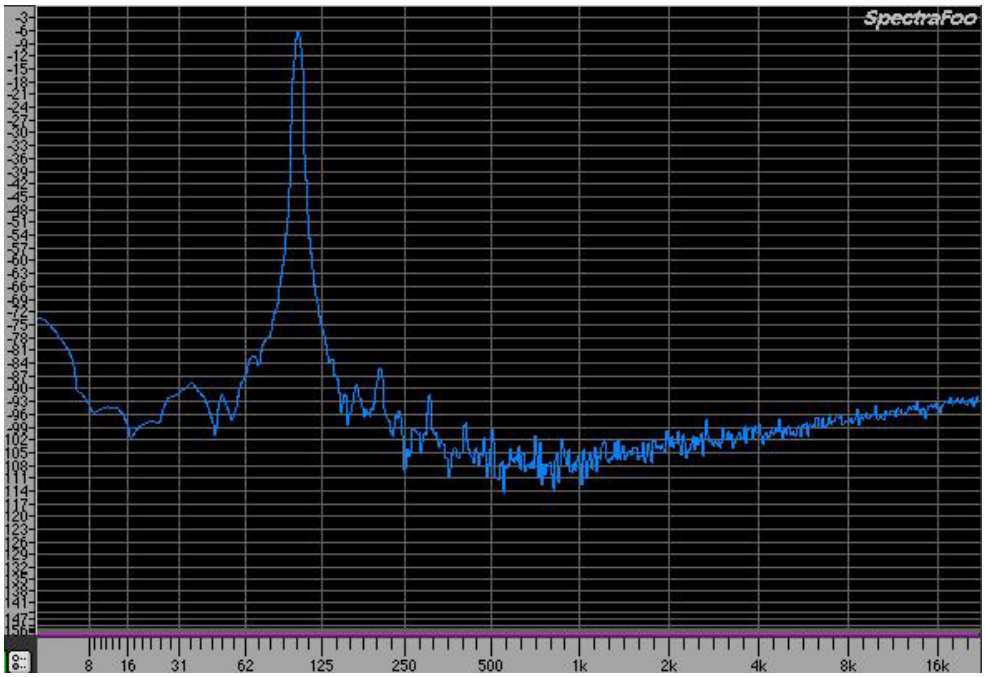


1. ábra. Alacsony szintű 100 Hz-es szinusz jel, 24 biten.

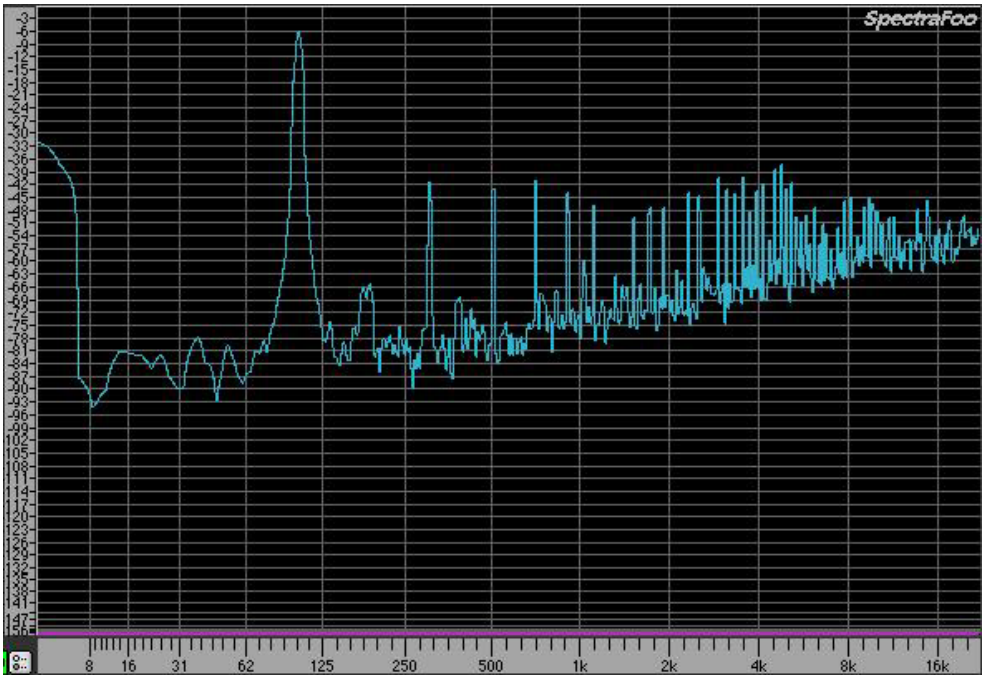


2. ábra. Alacsony szintű 100 Hz-es szinusz jel, 16 bit-re csonkolva.

Látható, hogy a négyszögesedés fellép, aminek hatása harmonikus torzítás, mely hallható egy hangjelben. Lássuk a spekturmokat!



3. ábra. Alacsony szintű 100 Hz-es szinusz jel spektruma, 24 biten.

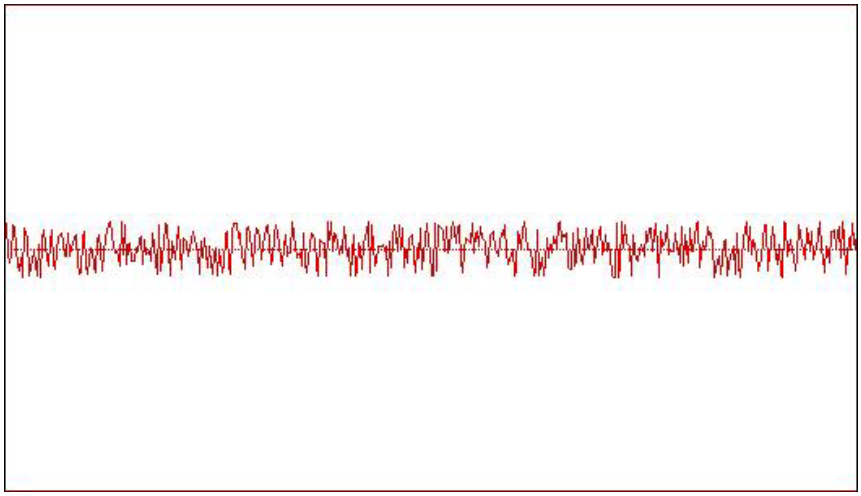


4. ábra. Alacsony szintű 100 Hz-es szinusz jel spektruma, 16 bit-re csonkolva.

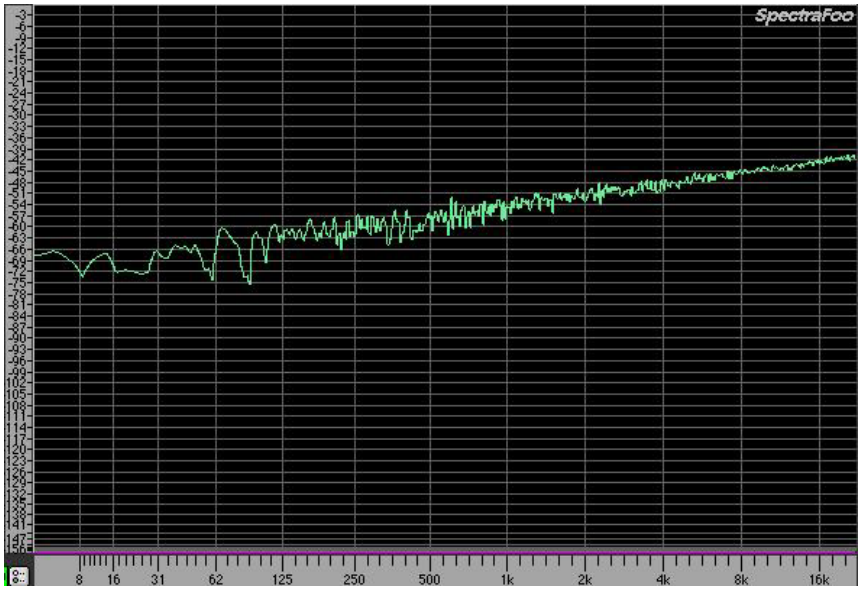
Torzítási komponensek kb. 36 dB-el alacsonyabbak a hasznos jel szintjénél. Azért ilyen magasak, mert a hasznos jel szintje is alacsony volt ebben az esetben. Teljes kivezérlésű (24 bitet elfoglaló) jelnél ez az arány lényegesen kisebb lenne. Ettől függetlenül látható, hogy a 24 bites jelnél a

zajszint 48 dB-el alacsonyabb (mert a dinamikatartomány 16 bitnél ennyivel kevesebb a 24-hez képest). Más szóval, ha 48 dB-el megnöveljük a dinamikatartományt, akkor az alapzajszint 48 dB-el lesz lejjebb ugyanannál a jelnél.

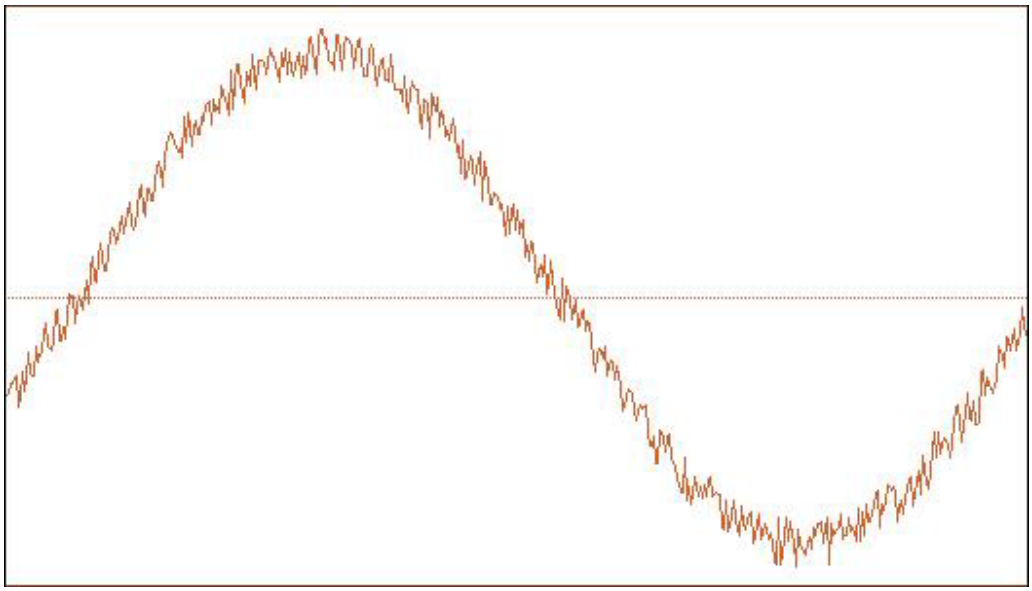
A dither célja, hogy a hozzáadott véletlen zaj „elrontsa” a négyszögletes lépcsők ezen hatását. Ebben a példában 24-ről 16 bitre csonkolunk, ezért 9 bites véletlen bitsorozat lesz a zaj. Ez azt jelenti, hogy véletlenszerűen választunk egy számot 0 és 512 között és ezt hozzáadjuk az eredeti jelhez (vagy -256-tól +256-ig számokkal is dolgozhatunk). Ezt fehérzajnak tekinthetjük, noha igazán véletlen nem lesz ez a sorozat.



5. ábra. A hozzáadott fehérzaj időfüggvénye.

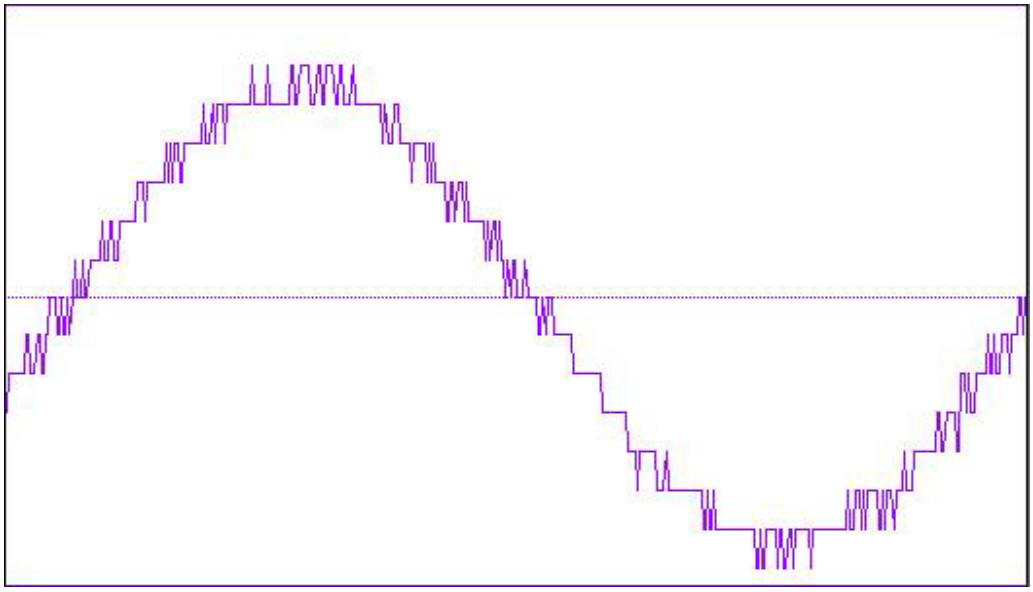


6. ábra. A hozzáadott fehérzaj spektruma.



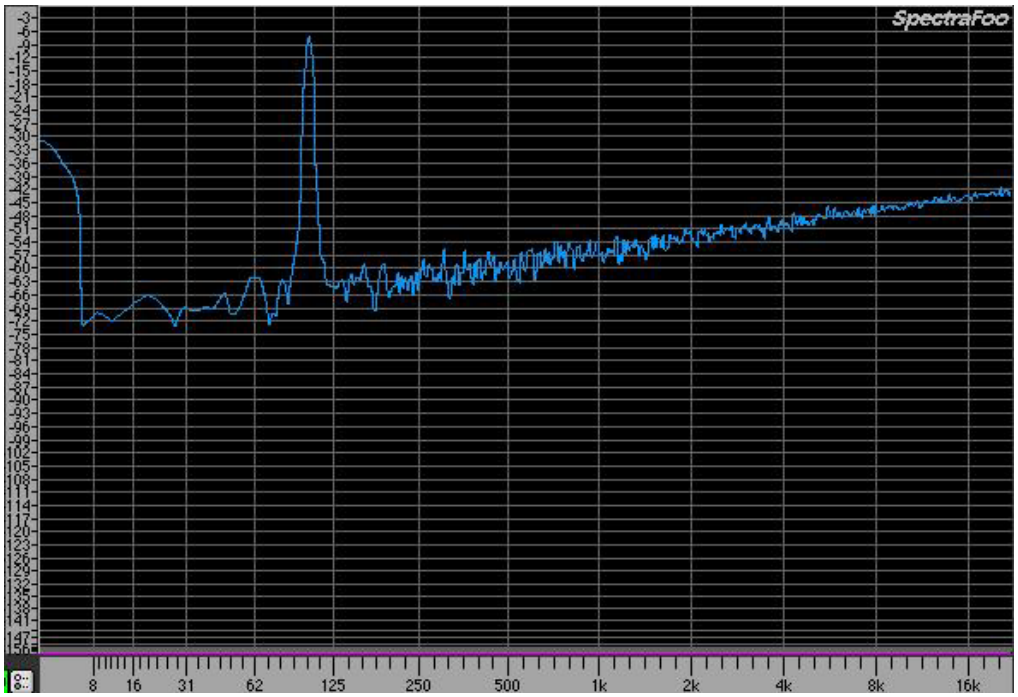
7. ábra. A 24 bites szinusz hullám hozzáadott fehérzajjal.

Ekkor még mindig 24 bites szavakat kezelünk, de ebből a legkisebb 9 bithez a fehérzajt is hozzáadtuk. Ha ezután eltávolítjuk (csonkoljuk) az utolsó 8 bitet, a kapott 16 bites jel már nem annyira lépcsőzetes, mint a fenti ábrán. Ellenben néhol kvantálási hibák jelennek 1 bit értékben.

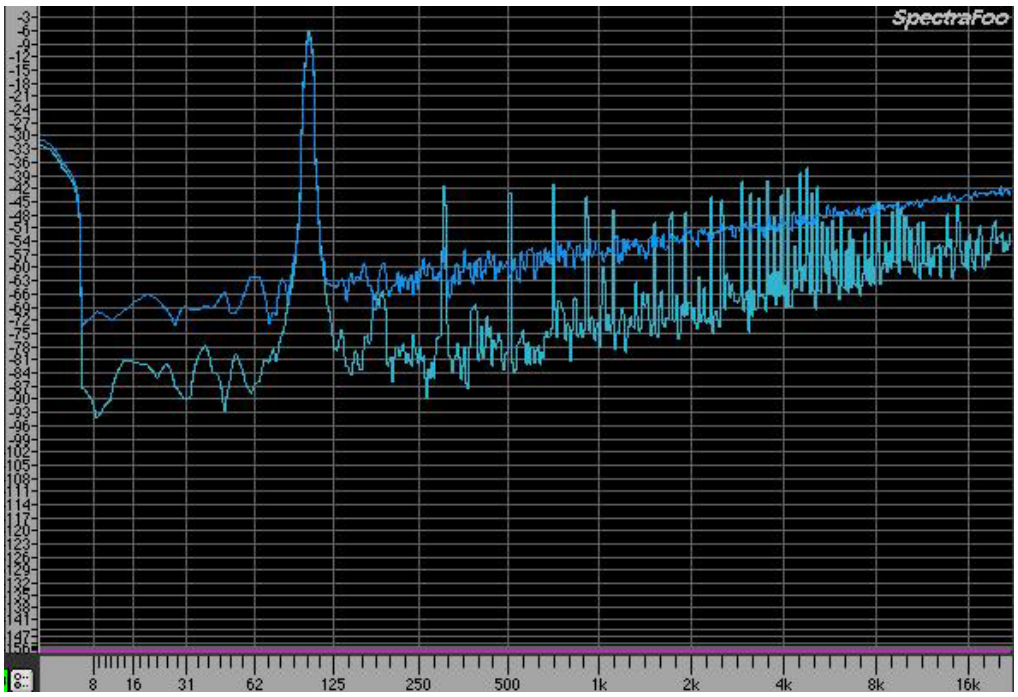


8. ábra. A 16 bitre csonkolt szinuszjel időfüggvénye dither hozzáadásával.

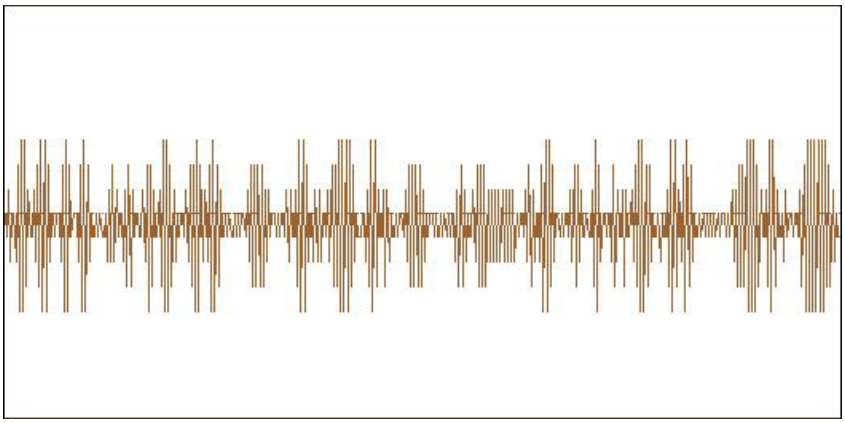
Ha ennek megvizsgáljuk a spektrumát, látható, hogy a harmonikus torzítási komponensek eltűntek. Helyette egy fehérzaj-spektrum jelent meg, amely nagyobb az eredeti 24 bitesnél, ugyanakkor kellemesebb a fülnek.



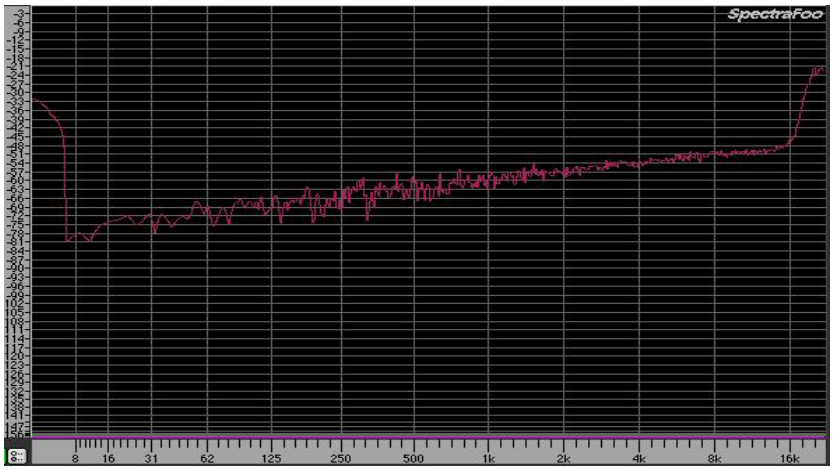
9. ábra. A 16 bitre csonkolt szinuszjel spektruma dither hozzáadásával.



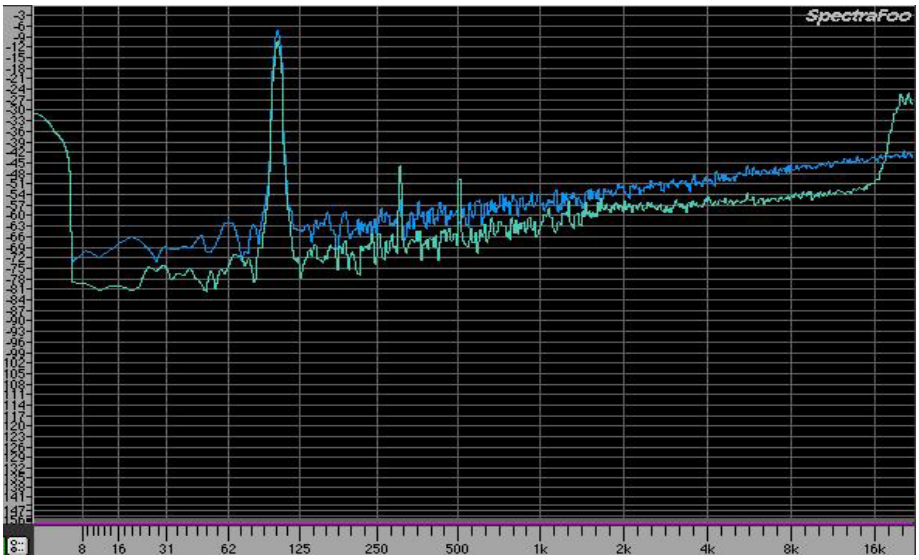
10. ábra. A 16 bitre csonkolt szinuszjel spektruma dither hozzáadásával és nélküle.



11. ábra. Egy másik dither-zaj időfüggvénye (UV22).



12. ábra. Az UV22 zaj spektruma: 16kHz-en fókuszálódik a zaj nagy része.



13. ábra. Szinuszjel spektrumok ditherrel illetve UV22-vel.