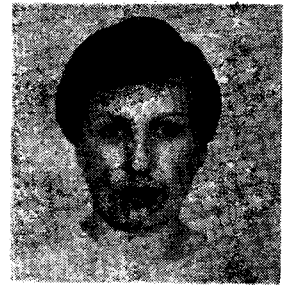


Digitális jelfeldolgozó programcsomag ismertetése

FÖLDVÁRINÉ OROSZ JULIANNA
Távközlési Kutatóintézet



ÖSSZEFOGLALÁS

A digitális jelfeldolgozás kérdéseinek széles spektrumát felölélő, TPA 1140 számítógépen működő programcsomagot ismertetünk.

Bevezetés

Az elmúlt tizenöt évben a digitális jelfeldolgozás különösen dinamikusan fejlődött. Az integrált áramkörti technológiában és a processzorok felépítésében bekövetkezett fejlődés megnövelte azoknak a területeknek a számát, amelyekben a digitális jelfeldolgozás eredményei sikeresen alkalmazhatók. A jelfeldolgozás alapvető módszereinek és algoritmusainak fejlesztése a jelfeldolgozó rendszerek hatékonyságát jelentősen növelte [2].

A digitális jelfeldolgozás fejlődésének lényeges vetülete olyan algoritmusok és számítógép-programok megalkotása, amelyek a jelek feldolgozását és a jelfeldolgozó szűrők és rendszerek tervezését végzik. E cikk bemutat egy programcsomagot, amely a digitális jelfeldolgozás széles spektrumát öleli fel. A programok az IEEE Acoustics, Speech and Signal Processing Society Digital Signal Processing Committee vezetésével különböző kutatóhelyeken készültek és egységes követelményeket elégítenek ki. Részletes leírásuk a Programs for Digital Signal Processing című könyvben jelent meg [1].

A Távközlési Kutatóintézetben elkészítettük e programcsomagnak a TPA 1140 számítógépen FORTRAN nyelven működő változatát. Valamennyi program egyszerű adatbevitelt tesz lehetővé, mert a képernyőn megkérdezi a bemenő adatokat. Nagy adathalmazok beolvasására és a kisebb módosításokkal megismételt futtatások megkönnyítésére előkészíthetjük és mágneslemezen tárolhatjuk az adatokat. A kért eredmények a képernyőn vagy a sornyomatón jelennek meg vagy mágneslemezen tárolhatók [14].

E cikkben felvázoljuk azokat a feladatokat, amelyek a programcsomag segítségével megoldhatóak. Néhány mondatban ismertetjük a programok funkcióját, lényeges jellemzőit.

A programok ismertetése

A FOUREA nevű program N pontból álló komplex számsorozat diszkrét Fourier-transzformáltját vagy inverz transzformáltját számolja ki az alábbi definíció alapján:

FÖLDVÁRINÉ
OROSZ JULIANNA

1977-ben szerzett villamosmérnöki oklevelet a Budapesti Műszaki Egyetemen. Munkáját a Távközlési Kutatóintézetben kezdte, ahol elektronikus

eszközök modellezésével foglalkozott. 1982-től 1985-ig az MTA SZTAKI-ban ösztöndíjas aspiráns volt. 1985 óta a TKI munkatársa. Az analóg és digitális jelfeldolgozás kérdéseivel foglalkozik. Pollák—Virág-díjas.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot \exp\left(-j \frac{2\pi}{N} nk\right)$$

$$k = 0, 1, \dots, N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot \exp\left(j \frac{2\pi}{N} nk\right)$$

$$n = 0, 1, \dots, N-1$$

N 2 hatványa kell, hogy legyen, $N \leq 2048$. A program a Cooley—Tukey algoritmus alapján dolgozik. Először átrendezi a bemeneti komplex számsorozatot, majd $\log_2 N$ lépésben a „pillangókat” számolja ki.

A következő három program valós vagy komplex számsorozatok 2—4—8 radixú diszkrét Fourier-transzformáltját és inverz transzformáltját számolja ki. Előnyük az, hogy igen gyorsak, ezért azonban a programok bonyolultságával és memória-igényük növekedésével kell fizetnünk. Mindhárom program esetén a pontok száma $N \leq 1024$. A FAST program valós bemenő adatsorozat 4—2 radixú diszkrét Fourier-transzformáltját és az inverz transzformáltat számolja ki a Cooley—Tukey algoritmus alapján. N pont esetén csak $N/2 + 1$ komplex DFT-értéket kell kiszámolni és tárolni, hiszen a bemeneti sorozat valós és $X(k)$ szimmetrikus, azaz $X(N-k) = X^*(k)$, $k = 0, 1, \dots, N/2$. A * komplex konjugáltat jelöl.

Az FFA program ugyancsak valós bemenő adatsorozat 8—4—2 radixú DFT-jét és szimmetrikus komplex adatsorozat 8—4—2 radixú IDFT-jét (a kimenet valós) számolja.

Az FFT842 program komplex adatsorozatokat 8—4—2 radixú DFT—IDFT-jét számolja a Cooley—Tukey algoritmus alapján.

Az FFTSUBS nevű program speciális tulajdonságú számsorozatok DFT-jét és IDFT-jét számolja ki. A speciális tulajdonságok a következők:

- szimmetrikus valós sorozatok,
- antiszimmetrikus, valós sorozatok,
- valós, csak páratlan harmonikusokat tartalmazó sorozatok,

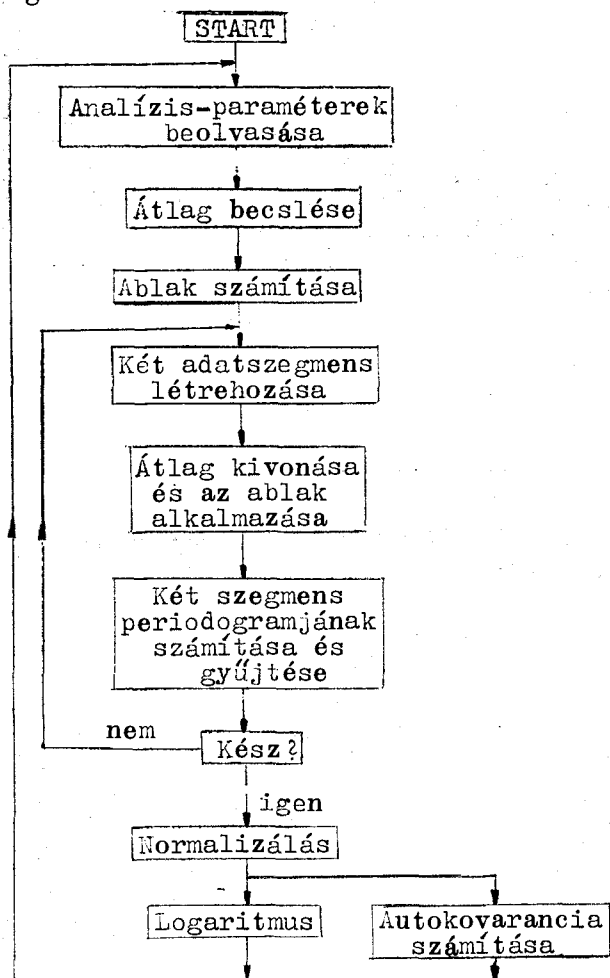
Beérkezett: 1987. V. 10. (□)

- valós, szimmetrikus, csak páratlan harmonikusokat tartalmazó sorozatok,
- valós, antiszimmetrikus, csak páratlan harmonikusokat tartalmazó sorozatok.

E sorozatok Fourier-transzformáltját az [1] irodalom részletesen elemzi. A sorozatok hossza legfeljebb 1024 pont lehet.

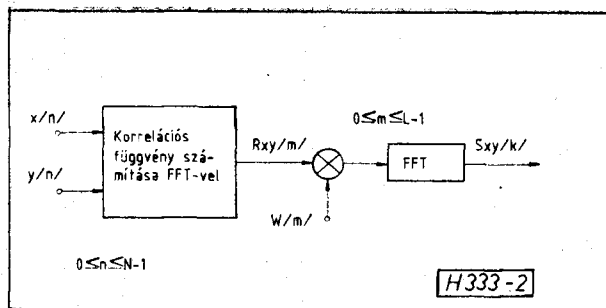
A felsorolt szimmetria-tulajdonságok gyakran előfordulnak. Az autokorrelációs függvény például mindig valós és páros. Bizonyos típusú Csebisev-approximációkban csak páratlan harmonikusokat tartalmazó cosinus-sorozatok fordulnak elő, s csak páratlan harmonikusokat tartalmazó sinus-sorozatok pedig igen hasznosak olyan lineáris parciális differenciálegyenletek megoldásában, amelyek határfeltételei zérus értéket kívánnak az egyik oldalon, zérus deriváltakat a másikon.

Az eddig ismert DFT-algoritmusokban az adatsorozatok hossza 2 hatványa kell, hogy legyen. A vegyes radixú FFT-program, MXFFT futtatásakor az adatsorozatok hossza a 2-től eltérő prímtenyezőket is tartalmazhat. E program másik előnye az, hogy könnyen és hatékonyan alkalmazható többdimenziós komplex és valós transzformációkban. A TPA-gépen futó változattal legfeljebb három dimenziós transzformáció végezhető el.



H333-1

1. ábra. A PMPSE-program blokkvázlata



2. ábra. A CMPSE-program blokkvázlata

A CHIRPZ program egy számsorozat z-transzformáltját határozza meg a z-sík bármely pontjából induló kör vagy spirál alakú kontúr mentén. Az [1] referenciában publikált chirp z-transzformációs algoritmus bizonyos értelemben a diszkrét Fourier-transzformáció általánosítása. A DFT-t úgy is értelmezhetjük, mint az egységkör N darab egyenletesen elosztott pontjában végzett z-transzformációt. A chirp z-transzformáció pedig tetszőleges, z-síkbeli spirál mentén egyenletesen elhelyezkedő, tetszőleges számú pont z-transzformáltját adja. E transzformáció diszkrét jelek általánosított spektrum-analízisét végzi és ugyanaz a szerepe diszkrét jelekre, mint a Laplace-transzformációnak folytonos jelekre. A bemeneti $x(n)$ jel ($0 \leq n \leq N-1$) z transzformáltja a $z = AW^a$ kontúr mentén a $z_k = AW^k$ ($k=0,1,\dots, M-1$) pontokban

$$X(z_k) = X_k = \sum_{n=0}^{N-1} x(n) A^{-n} W^{nk}$$

A CHIRPZ program a diszkrét konvolúciót alkalmazva hatékony algoritmussal számolja ki e transzformáltat.

A WFFT nevű program N adatból álló számsorozat diszkrét Fourier-transzformáltját számolja ki, ahol N a 2, 3, 4, 5, 7, 8, 9, 16 halmazból választott relatív prímelek szorzata, de legfeljebb 630 lehet. A program a Winograd-algoritmus alkalmazása általános N esetre. Ez az algoritmus az FFT-től különbözik. Ha a bemenő adatsorozat hossza 2 hatványa, akkor valamelyik FFT-t használó programot kell futtatni, ha N más prímtenyezőket is tartalmaz, akkor a WFFT program esetleg 40%-kal gyorsabb, mint a vegyes radixú DFT-t számoló MXFFT. Nem lehet azonban általános összehasonlítást adni e programokra, mindig a feladat és a rendelkezésre álló táblázatok alapján kell eldönteni, hogy melyiket használjuk. A WFFT memóriaigénye nagyobb, mint az FFT-programoké.

A hatékony, 4-es radixú Fourier-transzformációt végző R4FFT program szélsőséges példája a gyorsaság érdekében hozott memória-áldozatnak.

A program L. R. Morris autogén technikájával készült, amelynek ismertetése a [13] irodalomban található. A program igen hatékony, ha azonos pontszámmal több transzformációt végzünk. Ilyen problémák merülnek fel például a beszédanalízisben és -szintézisben.

A következő néhány program a nagy sebességű konvolúciós, korrelációs algoritmusokat és a spektrumanalízis algoritmusait dolgozza fel. Az első program (PMPSE) a jelek teljesítmény-spektrumát becsüli a módosított periodogram-módszerrel [3, 4]. A számítások során először az adatokat olvassa be (az FFT hossza, az ablak típusa, hossza, a felhasználható minták maximális száma, a mintavételi frekvencia), majd az adatsorozat átlagát számolja ki. Erre azért van szükség, hogy a spektrum kis frekvenciás részét ne nyomjuk el. Ezután a módosított periodogramokat számolja a program úgy, hogy a bemenő jelet egymást átfedő szegmensekre osztja, e szegmensekre alkalmazza a kívánt típusú (háromszög, Hamming) ablakozást, elvégzi szegmensenként a gyors Fourier-transzformációt, végül a módosított periodogramokat átlagolja a normalizált frekvenciasorozatra. Említésre méltó sajátossága a programnak, hogy két periodogramot számol egyszerre az időtartománybeli két valós jelszegmenst egy sorozat valós és képzetes részének tekintve. Így két periodogramot számol egy FFT végrehajtásakor. Az utolsó lépés a normalizálás. A bemenő jel autokovarianciája is becsülhető a teljesítményspektrum inverz Fourier-transzformáltjával. Az így kapott autokovariancia a valódinak torzítása, mivel nem a teljesítményspektrumból, hanem annak mintavett változatából számolja a program. A PMPSE program blokkvázlata az 1. ábrán látható.

A CMPSE program az FFT-technika felhasználásával meghatározza egy jel autokorrelációs függvényét vagy két jel keresztkorrelációját. Becsüli a jel teljesítményspektrumát is az ablakozott korrelációs függvény diszkrét Fourier-transzformáltjával. A program a Rader által [5]-ben publikált módszert alkalmazza, amelynek blokkvázlata a 2. ábrán látható.

$$R_{xy}(m) = \frac{1}{N} \sum_{n=0}^{N-1-m} (x(n) - \bar{x})(y(n+m) - \bar{y})$$

$$0 \leq m \leq L-1$$

$$\bar{x} = \frac{1}{N} \sum_{n=0}^{N-1} x(n), \quad \bar{y} = \frac{1}{N} \sum_{n=0}^{N-1} y(n)$$

A program lehetővé teszi, hogy a bemenő jelek átlagát kivonjuk a jelekből, ekkor a kovarianciafüggvényt számolhatjuk. Ha az átlagokat nem vonjuk ki, akkor a korrelációs függvényt becsüljük.

A CCSE program valós számsorozatok másodrendű statisztikai függvényeit becsüli, például a teljesítménysűrűség-spektrumot, a koherenciafüggvényt és az általánosított keresztkorrelációs függvényt. E számítások különösen a radartechnikában és a digitális jelfeldolgozás víz alatti alkalmazásaiban fontosak. Az algoritmus az átlagoló gyors Fourier-transzformációt használja. A fent felsorolt függvények becsülését négy lépésben végzi. Először a két digitális jelet N darab P pontból álló szegmensre osztja. A második lépésben valamennyi szegmenst egy simító függvényvel megszorozza.

A harmadik lépés a súlyozott, P pontból álló sorozatok z -transzformáltjának előállítás. Végül a program a Fourier-együtthatókból becsült végez a teljesítménysűrűség-spektrumra úgy, hogy átlagolja az N darab szegmensre külön-külön kiszámolt teljesítménysűrűség-értékeket.

A FASTFILT program diszken tárolt nagymennyiségű adat szűrésére készült. Az adatokat blokkonként olvassa be, szűri, majd egy másik file-ban tárolja az eredményeket. A szűrés a jól ismert átlapoló-összegző módszeren alapul [6, 7]. A szűrő megadható terminálról és diszk-file-ről egyaránt.

A lineáris predikció a beszédanalízis egyik legegyszerűbb körben használt technikája, amely a beszédjelek egyszerű és hatékony leírása néhány lassan változó paraméterrel. E paraméterek igen hasznosak a beszéd kódolásban, a beszéd felismerésben, a beszéd szintézisben és a beszélő azonosításában. Az utóbbi években a lineáris predikcióra számos módszert dolgoztak ki, amelyek figyelembe veszik a beszédjelek nemstacionárius jellegét. Nemstacionárius jeleket rövid idejű analízissel [8] lehet vizsgálni, amely a jelet olyan rövid szakaszokra osztja, amelyek alatt az megközelítően stacionárius. A különböző lineáris predikciós analízismódszerek abban különböznek egymástól, hogy az analízis intervallumán kívül milyennek tekintik a jelet a prediktor-együtthatók meghatározásakor.

Az LPC jelű program a lineáris predikcióanalízis autokorreláció- és kovariancia-módszerét mutatja be. A módszert vázlatosan ismertetjük, részletes leírása a [9] irodalomban található. Adott $\{X(0), X(1), \dots, X(N-1)\}$ adatsorozat és M egész szám esetén találunk olyan $\{a_1, a_2, \dots, a_M\}$ együtthatókat, amelyekkel az

$$\alpha = \sum_{n=n_0}^{n_1} [x(n) + \sum_{k=1}^M a_k x(n-k)]^2$$

összeg minimális. A minimalizálást a

$$\sum_{i=1}^M a_i c_{ik} = -c_{0k} \quad k=1, 2, \dots, M$$

$$c_{ik} = \sum_{n=n_0}^{n_1} x(n-i)x(n-k)$$

egyenlet megoldásával végezzük. A kovarianciamódszerben az n_0 és n_1 határokat úgy választjuk, hogy a $0 \leq n \leq N$ tartományon kívül eső $X(n)$ adatpontokra ne legyen szükség a fenti egyenlet értékelésekor, azaz $n_0 = M$, $n_1 = N-1$. Ebben az esetben a $\{C_{ik}\}$ együtthatóhalmaz szimmetrikus, pozitív szemidefinit mátrixot alkot. Ez a mátrix szingulárisává válik, ha az $\{x(n)\}$ sorozat M -edrendű vagy annál alacsonyabb rendű lineáris, homogén differenciálegyenletet elégít ki. Az autokorrelációs módszerben az adatsorozatot úgy tekintjük, hogy $x(n) = 0$ ha $n < 0$ és $n > N-1$. Ebben az esetben $n_0 = -\infty$ és $n_1 = +\infty$. A $\{C_{ik}\}$ halmaz szimmetrikus

pozitív Toeplitz-mátrixot alkotó és elemei a következőképpen fejezhetők ki:

$$c_{ik} = r(|i - k|)$$

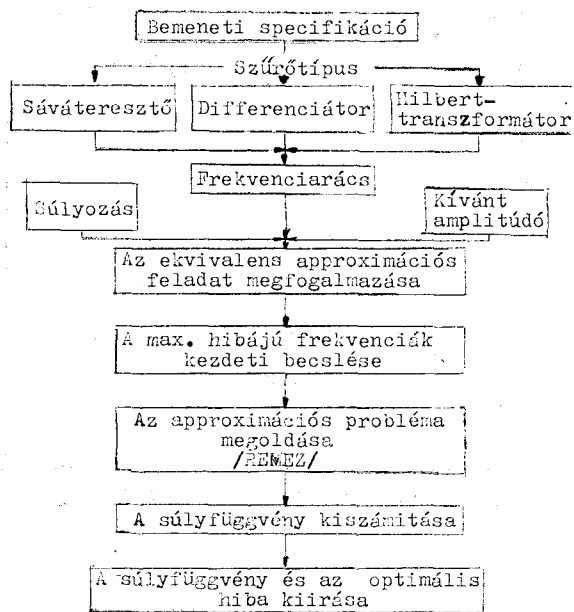
$$r_k = \sum_{n=0}^{N-1-k} x(n)x(n+k) \quad k=0, 1, \dots, N$$

A megoldáshoz $r(k)$ értékei csak a $k=0, 1, \dots, M$ esetben szükségesek. A bemutatott két módszer azonos eredményt ad, ha $x(n)=0$ $n < M$ és $n > N-1-M$ esetén, hasonló eredményeket adnak, ha $n \gg M$. A program a Cholesky-dekompozíciót és a Robinson-rekurziót alkalmazza. Az LPC programot egy beszédjel analízisére használtuk. A beszédjel 1 sec időtartamát 8kHz frekvenciával mintavételeztük és az így kapott diszkrét jelsorozatból határoztuk meg az első 20 lineáris predikciós együtthatót (3. ábra).

A lineáris predikció hatékony rácsmódszerét használja a LPCLAT jelű program. A rácsmódszer előnyei: nincs szükség a jel ablakozására, az eredményül kapott lineáris predikciós szűrő stabil, a stabilitás kevésbé érzékeny a szóhosszra. E rácsmódszerek hátránya, hogy költségesebbek a hagyományos autokorrelációs és kovariancia-módszerekénél. E hátrány kiküszöbölésére a rács paramétereit közvetlenül a jel kovarianciájából kell meghatározni, ezzel a számítás bonyolultsága megegyezik a hagyományos módszerekével.

k	a_k
0	1,000
1	-1,321
2	0,623
3	-0,463
4	0,494
5	-0,302
6	0,461
7	-0,250
8	-0,188
9	$0,628 \cdot 10^{-1}$
10	$0,850 \cdot 10^{-2}$
11	0,320
12	-0,311
13	0,113
14	-0,163
15	0,193
16	-0,112
17	0,118
18	$-0,248 \cdot 10^{-1}$
19	$0,138 \cdot 10^{-1}$
20	$0,333 \cdot 10^{-2}$

3. ábra. Egy beszédjel lineáris predikciós együtthatói az LPC-programmal számolva



H333-4

4. ábra. Lineáris fázisú digitális FIR-szűrők tervezésének blokkvázlata (EQFIR)

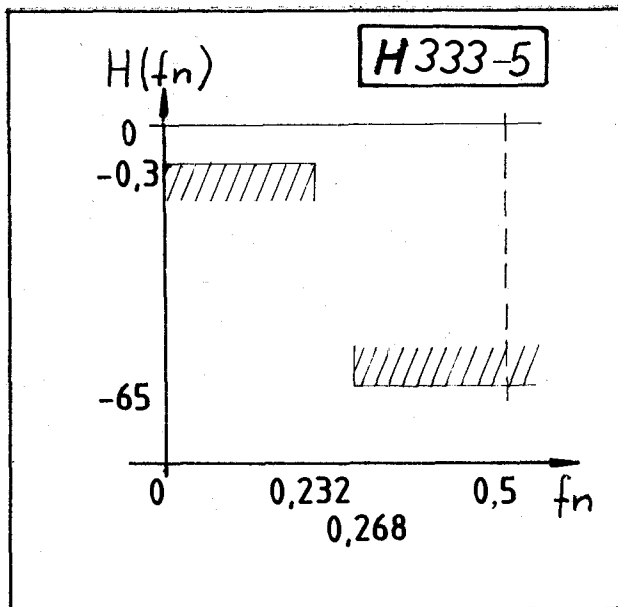
Az LPTRN jelű program lehetővé teszi, hogy a lineáris predikciós beszédanalízisben a $H(z)$ szintézis-szűrő megadására gyakran használt paraméterkészleteket egymásba transzformáljuk.

$$H(z) = \sqrt{\alpha} A(z)$$

$$A(z) = \sum_{k=0}^M a_k z^{-k} \quad \text{ahol } a_0 = 1$$

A bemenetként megadható paraméterkészletek a következők: szűrőegytáthatók és erősítés, kepsztromegytáthatók, autokorrelációs együtthatók, reflexiós együtthatók és bemeneti energia, logaritmikus area-viszonyok és bemeneti energia, area-függvények és bemeneti energia. Valamennyi esetben $M+1$ paraméter adható meg, amelyek egyértelműen meghatározzák a $H(z)$ szűrőt. A program a bemeneti paramétereiből valamennyi paramétert meghatározza.

Az EQFIR program lineáris fázisú digitális FIR szűrőket tervez a Remez-algoritmus segítségével [10]. A kívánt ideális átviteli függvény approximációja minimális súlyozott Csebisev-hibára törekszik. A program többszörös sáváteresztő és sávzáró szűrőket, differenciátorokat, Hilbert-transzformátorokat tervez, de tetszőleges átviteli függvények is approximálhatók a felhasználó által beépített szubrutinok alkalmazásával. A program működését a 4. ábrán mutatjuk be. A tervezés illusztrálásához egy szigorú követelményeket kielégítő, meredek csatornaszűrőt választottunk, amelynek specifikációja az 5. ábrán, az eredményül kapott FIR-szűrő együtthatóinak értéke a 6. ábrán látható. A szűrő hossza 68.



5. ábra. Egy meredek csatornaszűrő specifikációja az EQFIR-program számára

n	h/n/	n
1	$-1,735 \cdot 10^{-3}$	68
2	$-2,878 \cdot 10^{-3}$	67
3	$6,242 \cdot 10^{-4}$	66
4	$7,314 \cdot 10^{-3}$	65
5	$8,466 \cdot 10^{-3}$	64
6	$1,337 \cdot 10^{-3}$	63
7	$-3,492 \cdot 10^{-3}$	62
8	$1,519 \cdot 10^{-3}$	61
9	$6,234 \cdot 10^{-3}$	60
10	$5,563 \cdot 10^{-5}$	59
11	$-6,626 \cdot 10^{-3}$	58
12	$-7,760 \cdot 10^{-5}$	57
13	$8,423 \cdot 10^{-3}$	56
14	$1,049 \cdot 10^{-3}$	55
15	$-1,006 \cdot 10^{-2}$	54
16	$-2,040 \cdot 10^{-3}$	53
17	$1,219 \cdot 10^{-2}$	52
18	$3,523 \cdot 10^{-3}$	51
19	$-1,465 \cdot 10^{-2}$	50
20	$-5,471 \cdot 10^{-3}$	49
21	$1,767 \cdot 10^{-2}$	48
22	$8,128 \cdot 10^{-3}$	47
23	$-2,144 \cdot 10^{-2}$	46
24	$-1,182 \cdot 10^{-2}$	45
25	$2,646 \cdot 10^{-2}$	44
26	$1,722 \cdot 10^{-2}$	43
27	$-3,368 \cdot 10^{-2}$	42
28	$-2,582 \cdot 10^{-2}$	41
29	$4,547 \cdot 10^{-2}$	40
30	$4,175 \cdot 10^{-2}$	39
31	$-6,970 \cdot 10^{-2}$	38
32	$-8,238 \cdot 10^{-2}$	37
33	$1,563 \cdot 10^{-1}$	36
34	$4,432 \cdot 10^{-1}$	35

H333-6

6. ábra. Az 5. ábrán látható specifikációt kielégítő FIR-szűrő együtthatói

Az FWFIR jelű program FIR-szűrőket tervez az úgynevezett ablak-módszer alapján. A szűrők aluláteresztő, felüláteresztő, sáváteresztő és sávzáró típusúak, az ablak pedig négyzög, háromszög, Hamming-, Hanning-, Csebisev- és Kaiser-típusú lehet. Az ablakok $n=0$ -ra szimmetrikusak és a következő függvényekkel írhatók le:

— négyzög-ablak:

páratlan N esetén

$$w(n) = 1 \quad -(N-1)/2 \leq n \leq (N-1)/2$$

páros N esetén

$$w(n) = 1 \quad -N/2 \leq n \leq (N/2 - 1)$$

— háromszög-ablak:

páratlan N esetén $w(n) = 1 - |2n|/(n+1)$

$$-(N-1)/2 \leq n \leq (N-1)/2$$

páros N esetén $w(n) = 1 - |2n+1|/N$

$$-N/2 \leq n \leq (N/2 - 1)$$

— Hamming-ablak:

páratlan N esetén $w(n) = 0,54 +$

$$+ 0,46 \cos [2\pi n/(N-1)]$$

$$-(N-1)/2 \leq n \leq (N-1)/2$$

páros N esetén $w(n) = 0,54 +$

$$+ 0,46 \cos \left[\frac{2n(2n+1)}{2(N-1)} \right]$$

$$-N/2 \leq n \leq (N/2 - 1)$$

— Hanning-ablak:

páratlan N esetén $w(n) = 0,5 +$

$$+ 0,5 \cos [2\pi n/(N+1)]$$

$$-(N-1)/2 \leq n \leq (N-1)/2$$

páros N esetén $w(n) = 0,5 +$

$$+ 0,5 \cos \left[\frac{2\pi(2n+1)}{2(N+1)} \right]$$

$$-N/2 \leq n \leq (N/2 - 1)$$

— általánosított Hamming-ablak:

páratlan N esetén $w(n) = \alpha +$

$$+ (1-\alpha) \cos [2\pi n/(N-1)]$$

$$-(N-1)/2 \leq n \leq (N-1)/2$$

páros N esetén $w(n) = \alpha +$

$$+ (1-\alpha) \cos \left[\frac{2\pi(2n+1)}{2(N-1)} \right]$$

$$-N/2 \leq n \leq (N/2 - 1)$$

— Kaiser-ablak:

páratlan N esetén

$$w(n) = \frac{I_0 \left[\beta \sqrt{1 - \frac{4n^2}{(N-1)^2}} \right]}{I_0(\beta)}$$

$$-(N-1)/2 \leq n \leq (N-1)/2$$

páros N esetén

$$w(n) = \frac{I_0 \left[\beta \sqrt{1 - \frac{4(n+1/2)^2}{(N-1)^2}} \right]}{I_0(\beta)}$$

$$-N/2 \leq n \leq (N/2 - 1)$$

ahol β kívánt minimális zárósávi csillapításból adódó paraméter.

— Csebisev-ablak:

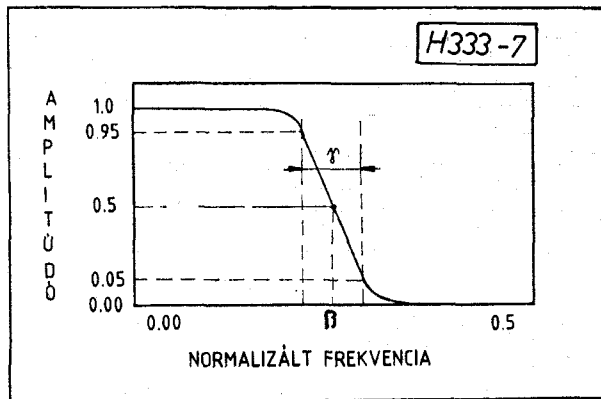
$w(n)$ az egységkör mentén egyenletesen elosztott N frekvencián értelmezett Csebisev-polinom inverz diszkrét Fourier-transzformáltja.

Az MXFLAT jelű program maximálisan lapos, szimmetrikus aluláteresztő FIR-szűrők együtt határozza meg a szűrő átmeneti sávja szélességének és középpontjának ismeretében. A specifikációt, azaz β és γ értékét a 7. ábrának megfelelően kell megadni.

Az FWFIREV jelű program lineáris fázisú FIR-szűrőket tervez minimális együtt határozható szóhosszal. Egy n_F hosszúságú FIR-szűrő átviteli függvénye

$$H(z) = \sum_{v=0}^{n_F-1} h_0(v)z^{-v}$$

ahol $h_0(v)$ a súlyfüggvény. Lineáris fázisú FIR-szűrők súlyfüggvénye szimmetrikus, azaz $h_0(v) = \pm h_0(n_F-1-v)$, így a súlyfüggvénynek $n_1 = \left\lfloor \frac{n_F+1}{2} \right\rfloor$ különböző eleme van. A program ezt az n_1 darab együtt határozható határozza meg a lehető legkisebb együtt határozható szóhosszal. A specifikáció a következő szűrőkre adható meg: aluláteresztő, felüláteresztő, szimmetrikus sáváteresztő, és sávzáró, valamint Hilbert-transzformátor.



7. ábra. Maximálisan lapos, szimmetrikus FIR-szűrő specifikációja az MXFLAT jelű program számára

Az LPIIR jelű program rekurzív digitális szűrőket tervez a Fletcher—Powell-féle függvény-minimalizálási módszer és a minimális p-hibakritérium alapján. Az algoritmus a hiba p-dik hatványát minimalizálja, ami $p=2$ esetén a legkisebb négyzetes hiba keresését, $p \rightarrow \infty$ esetén pedig a minimax kritériumot jelenti. Ugyanakkor a program fázis-kiegyenlítésre is lehetőséget ad.

Az IIR- és a FIR-szűrők tervezése több lényeges szempontból különböző. Az IIR-szűrők tervezésében felhasználhatjuk a klasszikus szűrőterve-

zés eredményeit. A FIR-szűrők együtt határozható meg határozása lineáris feladat, az IIR-szűrőké nemlineáris. Emiatt az IIR-tervező programok sokkal bonyolultabbak és több időt igényelnek, mint FIR megfelelőik. IIR-szűrők transzferfüggvényét optimalizálja az OPTIIR jelű program. A szűrők átviteli függvényének kaskád realizációjából kiindulva addig változtatja az együtt hatók értékét, amíg az abszolútértékkel szemben támasztott tetszőleges követelményeket a szűrő ki nem elégíti.

Az FWIIR jelű program abszolútérték-karakteristikájával specifikált véges szóhosszúságú IIR-szűrőket tervez. A Hooke—Jeeves-algoritmus általánosított változata, amely az optimális szóhosszat heurisztikus módszerrel határozza meg. Kiindulásként egy olyan nagy pontosságú szűrőt tervez, amely kielégíti a specifikációt. Ezután úgy optimalizálja az együtt hatókat, hogy szóhosszuk adott értéket ne haladjon meg, de a szűrő a kívánt specifikációt teljesítse.

A beszédanalízisben, a geofizikai adatfeldolgozásban és más alkalmazásokban használják a kepstrumszámítást. A komplex kepstrum egy jelsorozat Fourier-transzformáltja komplex logaritmusának inverz Fourier-transzformáltja. Ezt a komplex kepstrumot határozza meg a CEP jelű program [12], ahol $x(n)$ a bemeneti jelsorozat, $\hat{x}(n)$ a kepstrum:

$$\hat{x}(n) = \text{IDFT}\{\log[\text{DPT}(x[n])]\}$$

Az NPCEP jelű program tetszőleges fázisú, valós számsorozat valós kepstrumát és minimál-fázisú visszaállított a'akját állítja elő.

Az $x(n)$ jel minimál-fázisú rekonstrukciójához a jel valós kepstrumát a $w(n)$ ablakfüggvénnyel kell szorozni, ahol

$$w(n) = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ 2 & n > 0 \end{cases}$$

Ha $x(n)$ z-transzformáltja racionális, akkor ez azt jelenti, hogy a maximális fázisú pólusokat és zérusokat az egységkörre vett konjugált szimmetrikus megfelelőjükbe képezzük le. Végül a visszaállított minimál-fázisú jelet, $y(n)$ -t a komplex kepstrum inverzének kiszámításával kapjuk:

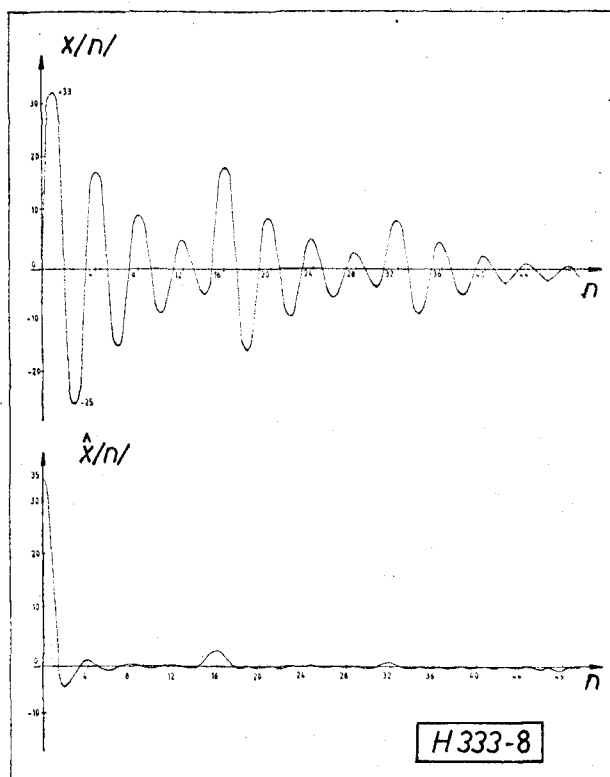
$$y(n) = \text{IDFT}\{\exp[\text{DFT}(\hat{y}[n])]\}$$

A 8. ábrán egy lecsengő sinus-jel és kepstruma látható az NPCEP jelű programmal számolva. Jól láthatók a két echo-jelnek megfelelő kepstrumvonalak.

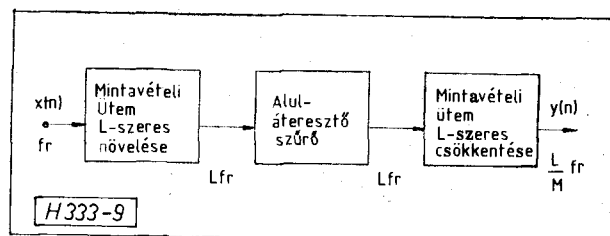
A digitális jelfeldolgozás a'apvető műveletei az interpoláció és a decimálás, amelyeket a mintavételi frekvencia változtatásakor kell alkalmazni. A keskenysávú FIR-szűrők és modulátorok tervezésében hasznos az interpoláció mind a'uláteresztő, mind fe'üláteresztő és sáváteresztő esetben. A két művelet másik alkalmazási területe a digitális fázistolás és a mintavételi idő tört részével történő késleltetés.

A decimáló vagy interpoláló szűrők tervezéséhez a szokásos módon (ablakozással, egyenletes előírással, stb.) tervezett aluláteresztő szűrőket használjuk fel. Speciális módszert kell használni, ha az interpoláló szűrőt az interpolációs hiba négyzetes átlagának minimalizálásával és az eredeti jel értékeinek megőrzésével tervezzük. Így dolgozik az INTFIL jelű program, amely a mintavételi ütemet R-szeresre növeli. Az eredményként kapott szűrő változatlanul átengedi a bemeneti mintákat és R-1 mintát helyez közéjük páronként úgy, hogy a minták és az elméleti értékük közötti hiba négyzete minimális legyen.

Az SRCONV jelű program egy adatsorozat mintavételi ütemét L/M-szeresére változtatja. L és M tetszőleges pozitív egész számok. Ha L=1, M-szeres decimálást, ha M=1, L-szeres interpolációt végez. Az algoritmus R. E. Crochiere és L. R. Rabiner módszerén alapul (9. ábra).



8. ábra. Lecsengő sinus-jel és kepsztuma



9. ábra. A mintavételi ütemet konvertáló (SRCONV) program blokkvázlata

A bemeneti mintavételi frekvencia f_r , ezt L-szeresére növeljük úgy, hogy a minták közé rendre L-1 darab zérus értékű mintát teszünk. Ez a jel aluláteresztő szűrőre kerül, melynek vágási frekvenciája az $f_r/2$ és $Lf_r/(2M)$ értékek közül a kisebb. A szűrő kimeneti jeléből minden M-dik mintát tartunk meg, a többit elhagyjuk.

Az MDECINT jelű program decimálja, interpolálja vagy keskenysávú szűrésnek veti alá a jelet. Ha a decimálás vagy az interpoláció aránya nagy, azaz 15 vagy nagyobb, akkor a műveleteket több, legfeljebb három lépésben végzi, ha az arány kisebb egész számok szorzataként felírható. Több lépésben végezve a műveleteket a lépésenként adódó számítási idő jelentősen csökken. Például ha 1:100 arányú interpolációt helyett 1:2, 1:5 és 1:10 arányút végzünk, akkor a számításigény közel 50%-ot csökken.

Köszönetnyilvánítás

Szeretnék köszönetet mondani Ákosfal Zoltánnak a mintapéldák futtatásában nyújtott segítségéért és Gáty Szilviának a rajzok elkészítéséért.

- [1] Programs for Digital Signal Processing, Edited by the Digital Signal Processing Committee, IEEE Acoustics, Speech and Signal Processing Society, IEEE Press, 1979.
- [2] *Simonyi Ernő*: Digitális szűrők, A jelfeldolgozás alapjai, Műszaki Könyvkiadó, Budapest, 1984.
- [3] *P. D. Welch*: The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms, IEEE, Vol. AU-15, No. 2, pp. 70-73, June, 1967.
- [4] *L. R. Rabiner, B. Gold*: Theory and Application of Digital Signal Processing, Chapter 6, pp. 399-419, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975.
- [5] *G. M. Rader*: An Improved Algorithm for High Speed Autocorrelation with Applications to Spectral Estimation, IEEE, Vol. AU-18, No. 4, pp. 439-442, 1970.
- [6] *T. G. Stockham*: High Speed Convolution and Correlation, Proc. AFIPS Spring Joint Computer Conference, Vol. 28, pp. 229-233, 1966.
- [7] *H. D. Helms*: Fast Fourier Transform Method of Computing Difference Equations and Simulating Filters, IEEE, Vol. AU-15, No. 2, pp. 85-90, June, 1967.
- [8] *J. L. Flanagan*: Speech Analysis, Synthesis and Perception, Springer Verlag, 1972.
- [9] *J. D. Markel, A. H. Gray Jr.*: Linear Prediction of Speech, Springer Verlag, New York, 1976.
- [10] *J. H. McClellan, T. W. Parks*: A Unified Approach to the Design of Optimum FIR Linear Phase Digital Filters, IEEE, Vol. CT-20, pp. 697-701, November, 1973.
- [11] *Szilvási M.*: Időmultiplex digitális IIR-szűrők hardware tervezése, Egyetemi doktori értekezés, Budapest, 1986.
- [12] *A. V. Oppenheim, R. W. Schaffer*: Digital Signal Processing, Chapter 10, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.
- [13] *L. R. Morris*: Automatic Generation of Time Efficient Digital Signal Processing Software, IEEE, Vol. ASSP-25, No. 1, pp. 74-79, February, 1977.
- [14] *Földváriné Grosz Julianna*: A SIPO-programcsomag felhasználói dokumentációja, TKI Intézeti Tanulmány, Budapest, 1982.