Gyors eljárások a diszkrét Fourier-transzformáció számítására. I. rész

DR. KOCSIS FERENC
Budapesti Műszaki Egyetem
Híradástechnikai Elektronika Intézet

ÖSSZEFOGLALAS

A digitális jelfeldolgozás egyik legfontosabb műveletének, a diszkrét Fourier-transzformációnak (DFT) az elvégzésére szolgáló gyors eljárások keresését tűztük ki célul egy gyakorlatban is felmerülő feladat kapcsán. A DFT számítási bonyolultságának a mérésére a szükséges valós szorzások számát választva, összefüggést származtattunk a kötött idejű (real-time) jelfeldolgozás maximális frekvenciájának a valós szorzások száma függvényében történő meghatározására. A közvetlen kiértékelés kötött idejű feladatok megoldására csak korlátozott mértékben alkalmas, műveletigénye $O(N^2)$. A jelfeldolgozási frekvencia növelése érdekében a szorzások számát algoritmikus eszközökkel próbáltuk csökkenteni. A fokozatos részekre osztással a műveletigény $O(N \log N)$-re csökkent. A transzformáció pontszámára tett bizonyos feltevések esetén (egymáshoz relatív prím számok szorzatára bontható) az egydimenziós DFT többdimenziós transzformációvá alakítható. A fokozatos részekre osztás lehetőségeinek kimerítése után a DFT számítását ekvivalens módon más feladat megoldására próbáltuk visszavezetni. Ha a pontszám $N=2, 4, p, p^k$ vagy $2p^k$ (p páratlan prím) alakú, akkor a DFT számítása ekvivalens a periodikus konvolúció meghatározásával. A lineáris és a periodikus konvolúció kiértékelésére a polinomokra vonatkozó kínai maradéktétel felhasználásával $O(N)$ szorzásigényű algoritmus származtatható. Röviden összefoglaljuk a periodikus konvolúcióra való visszavezetéssel számítható kis pontszámú, optimális ($N=2, 3, 4, 5, 7, 8, 9$ és 16) DFT modulokat. Nagyobb pontszámú transzformációk számításához a kis pontszámú modulok összekombinálhatók a Good-algoritmussal, ill. a Winograd-eljárással (WFTA). Utóbbi szorzásigénye $O(N)$. Végül elemezzük az egyes algoritmusokat a gyakorlati megvalósítás szempontjából.

DR. KOCSIS FERENC

1975-ben szerzett villamosmérnöki diplomát a BME Villamosmérnöki Karán, majd a Távközlési Kutató Intézetben kezdett dolgozni. Egyetemi doktori értekezését 1978-ban védte meg. 1983 szeptembere óta a BME HEI-ben dolgozik tudományos ösztöndíjasként, ahol a digitális jelfeldolgozás és jelszintézis algoritmikus kérdéseivel foglalkozik. Szakmai érdeklődési köre: rendszertechnika, digitális jelfeldolgozás, számítástechnika, algoritmusok elmélete.

1. Bevezetés

A digitális jelfeldolgozásban központi szerepet játszik az N elemmel leírható (véges vagy periodikus) számsorozathoz rendelt Fourier-spektrumot N egymástól egyenlő távolságú pontban vett mintákkal megadó diszkrét Fourier-transzformált (DFT). Definíció szerint véges, vagy periodikus sorozat egydimenziós diszkrét Fourier-transzformáltján (1–D DFT) az

$$(1\text{-}1)\quad X(k)=\sum_{i=0}^{N-1} x(i)w^{ik} \quad 0\leq k\leq N-1 \quad w_N^{ik}=e^{-j\left(\frac{2\pi}{N}\right)ik}$$

sorozatot értjük. $\{x(i)\}$ a szóban forgó számsorozat, vagy annak egy periódusa, $\{X(k)\}$ a transzformált értékek sorozata. Periodikus esetben N a jel egy periódusra eső pontjainak a száma, míg véges sorozatnál N a sorozat pontjainak a száma.

A DFT lineáris transzformáció, alapvető tulajdonságainak elemzése megtalálható pl. a [9], [10], [21] művekben. A transzformáció pontsorozatot képez le azonos számú pontot tartalmazó pontsorozatba: $\{x(i)\}\rightarrow\{X(k)\}$.

Mátrixalakban:

$$(1\text{-}2)\qquad \mathbf{X}=\mathbf{W}_N\cdot\mathbf{x}, \qquad \text{ahol} \qquad \mathbf{W}_N=$$

$$=\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w_N^1 & w_N^2 & \dots & w_N^{N-1} \\ 1 & w_N^2 & w_N^4 & \dots & w_N^{2(n-1)} \\ \vdots & & & & \\ 1 & w_N^{N-1} & w_N^{2(N-1)} & \dots & w_N^{(N-1)(N-1)} \end{bmatrix} \quad a$$

transzformációs mátrix, $\mathbf{X}^T=[X(0), X(1), \dots, X(N-1)]$ *a* transzformált értékek vektora és $\mathbf{x}^T=[x(0), x(1) \dots, x(n-1)]$ a kiindulási ponthalmazt leíró vektor. $\mathbf{X}^T$ és $\mathbf{x}^T$ elemei általános esetben komplexek is lehetnek.

Egy, a műszaki gyakorlatban felmerülő feladat kapcsán egységes keretbe foglaljuk a DFT gyors meghatározására kidolgozott eljárásokat ([5], [8], [11], [13], [23], [28]). A lényeges pontokat a mérnöki gyakorlat követelményeinek figyelembe vételével rövid levezetésekkel támasztjuk alá. Kifejezést származtatunk DFT felhasználásával történő, kötött idejű jelfeldolgozási feladatok (pl. spektrumanalízis) maximális sávszélessége (a maximális jelfeldolgozási frekvencia) és a DFT kiszámításához szükséges valós szorzások száma közti összefüggés leírására, majd annak felhasználásával elvégezzük az egyes DFT algoritmusok összehasonlító értékelését.

A továbbiakban szükségünk lesz az egyes DFT számítási eljárások értékelésénél valamilyen mértékre, amely alapján az egyes algoritmusok egységes alapon összehasonlíthatók, értékelhetők. A gyakorlati alkalmazások szempontjából egyik döntő tényező a DFT kiszámításához szükséges idő: sok feladatban ez korlátozza az elérhető maximális jelfeldolgozási frekvenciát. A jelenlegi áramköri technológiák és ismeretek mellett a DFT kiértékelése során a szorzás a legidőigényesebb művelet. Feltételezve, hogy a szükséges szorzások ideje mellett az egyéb műveletek (összeadások, kivonások, léptetések, adatmozgatások stb.) ideje elhanyagolható, a következőkben egy DFT eljárás számítási bonyolultságán a szükséges valós szorzások $f(N)$ számát értjük.

Beérkezett: 1984. VI. 6. (↔)

Tekintsük a következő problémát (amely összetett digitális jelfeldolgozási feladatok része is lehet): valamely folytonos jelből egymást követően t_s időintervallumonként N pontot tartalmazó mintákat veszünk, és képezzük ezen sorozatok diszkrét Fourier-transzformáltjait. A kapott spektrumon esetleg további jelfeldolgozási műveleteket kívánunk végezni. A kötött idejű (real-time) feldolgozás feltétele, hogy az N ponthoz rendelt spektrumot legfeljebb t_s idő alatt elő tudjuk állítani (különben az egyes sorozatok feldolgozásának már az első lépése is átlapolódna a következő részsorozat feldolgozási intervallumára). Ha egyetlen valós szorzás elvégzéséhez $t_{szorzás}$ időre van szükség, akkor az átlapolódás határhelyzetében (a DFT számításához szükséges idő éppen a következő mintasorozat vételéhez szükséges t_s idő):

(1-3) $$t_s = f(N) \cdot t_{szorzás}.$$

Az egy pontra eső t_s/N átlagos feldolgozási idő értékéből a mintavételi frekvenciára vonatkozó Nyquist-kritérium ($f_{mintavétel} = 2 \cdot f_{maxjel}$) felhasználásával a maximális, még feldolgozható jelfrekvencia:

(1-4) $$f_{maxjel} = \frac{N}{2 \cdot t_{szorzás} \cdot f(N)}.$$

Az összefüggésből látható, hogy az f_{maxjel} növelésének két útja kínálkozik:

- a $t_{szorzás}$ értékének csökkentése, ami új áramkörök, technológiák kidolgozását jelenti;
- az $N/f(N)$ hányados értékének növelése. Az egyik lehetőség N (a pontszám) csökkentése. Sok esetben azonban a pontszám értékét más követelmények (pl. felbontóképesség) határozzák meg. Emiatt járható útként elsősorban $f(N)$ értékének a csökkentése marad.

A továbbiakban algoritmikus eszközökkel próbáljuk $f(N)$ értékét csökkenteni. A vizsgálatok során azonban meghatározzuk a DFT számításához szükséges összeadások számát is.

A dolgozat három részből áll, irodalomjegyzék azonban csak az elsőhöz kapcsolódik. A többi részben levő hivatkozások az első részbeli irodalomjegyzék sorszámai szerintiek.

A jelen első részben először az (1-1) definíció szerinti összeg közvetlen kiértékelésével próbálkozunk. Ezt követi az algoritmuselméletben gyakran használt fokozatos részekre osztás (divide and conquer) elvének alkalmazása. A részekre osztás korlátainak felmérése után a második részben más úton kísérletezünk: az eredetileg kitűzött feladatot olyan más feladattá kíséreljük meg átalakítani ekvivalens módon, amelynek megoldására már ismert hatékony eljárás. Kimutatható, hogy bizonyos feltételek teljesülése esetén az egydimenziós (1—D) DFT többdimenziós (n—D) DFT-be alakítható át. A számelmélet eredményeinek felhasználásával egyes esetekben a DFT számítása periodikus konvolúció kiértékelésére vezethető vissza, amely elvégzésére a polinomok elméletének eredményei alapján adható igen hatékony algoritmus. A harmadik rész foglalkozik a DFT gyors konvolúciós eljárások felhasználásával történő meghatározásával. Az 1—D és n—D DFT együtthatómátrixai közti összefüggés felhasználásával kétféle módszert (a Good-eljárás és a Winograd-algoritmus) mutatunk be nagyobb pontszámú DFT kisebb pontszámú transzformáltakból történő előállítására. Végül a szükséges műveletszám alapján röviden összehasonlítjuk az ismertetett algoritmusokat, értékeljük az eredményeket.

2. A DFT számítása közvetlen kiértékeléssel

A közvetlen kiértékelés műveletigényének becsléséhez alakítsuk át az alapkifejezést (általános esetben $x(i) = a(i) + jb(i)$ alakú komplex szám (a és b valósak):

$$X(0) = \sum_{i=0}^{N-1} x(i) \qquad 1 \le k \le N-1$$

(2-1) $$X(k) = x(0) + \sum_{i=1}^{N-1} x(i) e^{-j\left(\frac{2\pi}{N}\right)ki} =$$

$$= x(0) + \sum_{i=1}^{N-1} (a(i) + jb(i)) \left(\cos\left(\frac{2\pi}{N}\right) ki - j \sin\left(\frac{2\pi}{N}\right) ki \right).$$

Két komplex szám szorzása elvégezhető 4 valós szorzás és 2 valós összeadás felhasználásával. A (2-2) azonosságok alapján azonban elegendő 3 valós szorzás is, aminek ára az összeadások számának 5-re történő növekedése:

(2-2) $$(a + jb)(c + jd) = p + jq \quad \left.\begin{aligned} w_1 &= a(c+d) \\ w_2 &= d(a+b) \\ w_3 &= c(b-a) \end{aligned}\right\} \begin{aligned} p &= w_1 - w_2 \\ q &= w_1 + w_2 \end{aligned}$$

Ekkor a közvetlen számítás műveletigénye $f(N) = 3 \cdot (N-1)^2$ valós szorzás és $A(N) = (5N-3) \cdot (N-1)$ valós összeadás. A bonyolultságvizsgálatok során szokásos jelölésmóddal: $f(N) = O(N^2)$ és $A(N) = O(N^2)$. Általában azt mondjuk, hogy egy $f(N)$ függvény $O(g(N))$ rendű, ha létezik olyan c állandó, amelyre az $f(N) \le c \cdot g(N)$ összefüggés legfeljebb véges számú érték kivételével minden egész, nem negatív N értékre teljesül.

N nagyobb értékeire a DFT konkrét meghatározása kötött időben igen nagy problémát jelenthet. Az (1-4) kifejezés alapján, gyors szorzóáramkört használva ($t_{szorzás} = 200$ ns) egy tipikus alkalmazásban ($N = 10^3$) az f_{maxjel} maximális jelfeldolgozási frekvencia értéke: $f_{maxjel} = 833$ Hz, ami a gyakorlati esetek többségében túl kicsi.

Az $\{x(i)\}$ jelsorozat speciális tulajdonságaira vonatkozó előzetes információ ismeretében a közvetlen kiértékelés műveletszáma jelentősen csökkenthető. Ilyen információ lehet pl. az adatok valós volta, az $\{x(i)\}$ sorozat szimmetrikus stb. Kimutatható azonban, hogy a szükséges szorzások $f(N)$ száma a lehetséges egyszerűsítések elvégzése után is $O(N^2)$ rendű, csupán a c állandó értéke csökken. Ez azt jelenti, hogy az f_{maxjel} frekvencia értéke csak azonos nagyságrenden belül növelhető.

3. A DFT meghatározása fokozatos részekre osztással

3.1. *Számítás a részekre osztás elvének felhasználásával*

Az algoritmuselmélet gyakran alkalmazott fogása a kiindulási probléma kisebb méretű részekre történő felbontása, az ezekhez tartozó megoldások megkeresése, majd a kapott eredmények összekombinálásával az eredeti feladat megoldásának összeállítása. A módszer gyakran vezet a közvetlen megoldásnál gyorsabb (kisebb műveletigényű) megoldásra különösen akkor, ha a részekre osztásnál az eredeti feladat kisebb méretű változatai állnak elő. Ezen megoldástípusokat „divide-and-conquer" (fokozatos részekre osztás) típusú algoritmusoknak szokás nevezni. Próbáljuk meg a módszert nagy méretű (nagy pontszámú) DFT számítására alkalmazni, azaz több kisebb méretű DFT meghatározására visszavezetni.

Feltéve, hogy az eredeti DFT N pontszáma öszszetett szám, a tényezőkre bontást elvégezve: $N=N_1 \cdot N_2 \cdot \ldots \cdot N_n$, ahol az egyes N_j $(1 \leq j \leq n)$ tényezőkre azon kívül, hogy egészek, semmilyen megkötést nem teszünk. A $P=N_2 \cdot N_3 \cdot \ldots \cdot N_n$ választással az (1-1) definíciós összefüggésben az indexeket N_1 és P lineáris kombinációjaként kifejezve:

$$(3\text{-}1) \qquad \begin{array}{ll} k=k_2+k_1P & \begin{array}{l} 0 \leq k_1 \leq N_1-1 \\ 0 \leq k_2 \leq P-1 \end{array} \\ i=i_1+i_2N_1 & \begin{array}{l} 0 \leq i_1 \leq N_1-1 \\ 0 \leq i_2 \leq P-1. \end{array} \end{array}$$

Az indexek más felbontása (nemlineáris, nem algebrai stb.) is elképzelhető, azonban a gyakorlatban leginkább hasznos a (3-1) típusú lineáris összefüggés. A felbontás láthatóan kölcsönösen egyértelmű leképezést teremt $k \to (k_1, k_2)$ és $i \to (i_1, i_2)$ között. A definíciós összefüggésbe helyettesítve:

$$(3\text{-}2) \qquad X(k_2+k_1P)=X(k_1, \; k_2)=$$
$$=\sum_{i_1=0}^{N-1} \sum_{i_2=0}^{P-1} x(i_1+i_2N_1) \cdot e^{-\mathrm{j}\left(\frac{2\pi}{N}\right)(k_1P+k_2)(i_2+i_2N_1)}.$$

Azonos átalakítások után $\left(e^{-\mathrm{j}\left(\frac{2\pi}{N}\right) k_1 i_2 P N_1} \equiv 1\right)$:

$$(3\text{-}3) \qquad X(k_1, \; k_2)=\sum_{i_1=0}^{N_1-1} e^{-\mathrm{j}\left[\left(\frac{2\pi}{N_1}\right) i_1 k_1+\left(\frac{2\pi}{N}\right) i_1 k_2\right]}.$$
$$\cdot \sum_{i_2=0}^{P-1} x(i_1, \; i_2) e^{-\mathrm{j}\left(\frac{2\pi}{N_2}\right) i_2 k_2}$$

A belső összeg csupán k_2 és i_1 függvénye:

$$(3\text{-}4) \qquad X_1(k_1, \; i_2)=\sum_{i_2=0}^{P-1} x(i_1, \; i_2) e^{-\mathrm{j}\left(\frac{2\pi}{N_2}\right) i_2 k_2}.$$

X_1 értékkészlete $N_1 \cdot P=N$ db értékből áll (valójában N elemű kétdimenziós tömb az (i_1, k_2) indexekkel). Közvetlen kiértékelésnél a szükséges szorzások száma (eltekintve a $k_1=0$ esetben lehetséges egyszerűsítéstől) $M_1=3 \cdot P^2 \cdot N_1=3P \cdot N$ valós szorzás. A külső összeg belső összegből való számításának lépésszáma $M_2=3N^2 \cdot P=3N \cdot N_1$ valós szorzás. A teljes műveletigény: $f(N)=M_1+M_2=3N \cdot (N_1+P)$. Látható, hogy a belső összeg valójában egy P-pontos DFT. A tényezőkre bontást folytatva, és az előző gondolatmenetet követve az eredeti DFT végül kisebb méretű DFT-k számítására vezethető vissza. Az eljárás $(n-1)$ lépésben ér véget (n a tényezők száma N felbontásában). Az összes valós szorzások száma:

$$(3\text{-}5) \qquad \mathrm{f}(N)=3N \cdot (N_1+N_2+\ldots+N_n),$$

ahol

$$N=\prod_{i=1}^{n} N_i.$$

Ha $N_i=a_i b_i$ $(a_i, \; b_i>1)$, akkor $a_i+b_i \leq N_i$, így általában (de nem minden esetben, mint azt majd látni fogjuk) célszerű N értékét a lehető legtöbb tényezőre felbontani, azaz a törzstényezős felbontásból kiindulni. $N=\prod_{i=1}^{n} p_i^{\alpha_i}$ esetén (3-5) alakja:

$$(3\text{-}6) \qquad f(N)=3N \sum_{i=1}^{n} \alpha_i p_i.$$

Ha $N>4$, akkor $\sum_{i=1}^{n} \alpha_i p_i<\prod_{i=1}^{n} p_i^{\alpha_i}$, azaz $\mathrm{f}(N)$ rendje valóban csökkent az $\mathrm{f}(N)=\mathrm{O}(N^2)$-hez képest.

$N=\mathrm{p}^n$ esetben (p prím) $\mathrm{f}(N)=3 \cdot N \cdot \log_p N$, vagyis $f(N)=\mathrm{O}(N \cdot \log N)$. A gyakorlatban különösen fontos $N=2^n$ esetben $\mathrm{f}(N)=\mathrm{O}(N \cdot \log_2 N)$. A DFT ily módon történő számítását szokás gyors Fourier-transzformációnak (FFT) nevezni.

3.2. *A nem triviális szorzások számának meghatározása 2 és 4 szerinti faktorizáció esetén*

A pontos műveletszám meghatározásánál sok egyszerűsítő tényező is figyelembe vehető. $N=2^n$ esetén a w_N^{ik} exponenciális tényezővel történő szorzás sok esetben ± 1, $\pm \mathrm{j}$ értékkel való szorzásra redukálódik (triviális szorzások), amit a tényleges számítás során természetesen nem szorzásként veszünk figyelembe.

A szorzások nem triviális számának számításához írjuk fel az (1-1) definíciós összefüggésben szereplő i és k indexeket kettes számrendszerben (ez minden különösebb feltétel teljesülése nélkül megtehető).

$$(3\text{-}7) \qquad \begin{array}{lll} k=2^{n-1}k_{n-1}+\ldots+k_0 2^0 & k_l=0, \; 1 & 0 \leq l \leq \mathrm{n}-1 \\ i=2^{n-1}i_{n-1}+\ldots+i_0 2^0 & i_l=0, \; 1 & n=\log_2 N. \end{array}$$

Az alapösszefüggésbe helyettesítve:

$$X(k_0, \; k_1, \; \ldots, \; k_{n-1})=$$
$$=\sum_{i=0}^{1} \ldots \sum_{i_{n-1}=0}^{1} x(i_0, \; i_1, \; \ldots, \; i_{n-1}) e^{-\mathrm{j}\left(\frac{2\pi}{N}\right) ki}$$
$$(3\text{-}8)$$
$$k \cdot i=\left(\sum_{l=0}^{N-1} k_l 2^l\right) \cdot \left(\sum_{l=0}^{n-1} i_l 2^l\right).$$

Behelyettesítve, átrendezve és a minden c egészre érvényes $e^{-\mathrm{j}2\pi c} \equiv 1$ azonosságot figyelembe véve:

$$(3\text{-}9) \qquad X(k_0, \; k_1, \; \ldots, \; k_{n-1})= \qquad (1 \leq p \leq n)$$
$$=\sum_{i_0=0}^{1} e^{-\mathrm{j}\left(\frac{2\pi}{N}\right) i_0\left(\sum_{l=0}^{n-1} k_l 2^l\right)} \ldots \sum_{i_n-p=0}^{1} e^{-\mathrm{j}\left(\frac{2\pi}{N}\right) i_{n-p} 2^{n-p}\left(\sum_{l=0}^{p-1} k_l 2^l\right)} \ldots$$
$$\ldots \sum_{i_{n-1}=0}^{1} x(i_0, \; i_1, \; \ldots, \; i_{n-1}) e^{-\mathrm{j}\left(\frac{2\pi}{N}\right) i_{n-1} 2^{n-1} k_0}.$$

A legbelső összeg láthatóan $N/2$ db 2-pontos DFT. Kiértékeléséhez nem szükséges szorzás. Általában az (n – p)-ik összegben az előző fokozatban keletkezett N db értéket kell szorozni egy exponenciális taggal, amelynek kitevője:

(3-10) $$-j\left(\frac{2\pi}{N}\right) i_{n-p} 2^{n-p} \sum_{l=0}^{p-1} k_l 2^l.$$

A legmagasabb indexű k taggal való szorzás ismét 2-pontos DFT-t jelent, amelynek számításához — mint már említettük — nincs szükség szorzásra. Az összeg többi tagjával történő szorzás forgatást jelent. Az összeg második tagjával (k_{p-2}) való szorzások kiesnek, mivel $e^{-j\left(\frac{2\pi}{N}\right) i_{n-p}k_{p-2}}$ értéke i_{n-p} és k_{p-2} lehetséges értékeire +1, ill. –j. Azon kívül még nincs szükség valós szorzásokra, ha $i_{n-p}=0$, ill. a $\sum_{l=0}^{p-3} k_l 2^l$ összegben az összes k együttható értéke zérus. Az előző fokozatbeli számítások után adódó n-változós eredmény:

(3-11) $$x_{n-p-1}(k_{p-2}, k_{p-3}, \ldots, k_0, i_{n-p}, i_{n-p-1}, \ldots, i_0).$$

A többi i_{n-q} változó ($1 \le q \le p-1$) a korábbi összegezések során eltűnik, és helyükre a k_s értékek ($0 \le s \le p-1$) kerülnek. Ha $i_{n-p}=0$, akkor a többi változó értékétől függetlenül nincs szükség szorzásra, így az eredetileg N db szorzásból $N/2$ kiesik. Kiesnek még azok a szorzások, amelyeknél a $\sum_{l=0}^{p-3} k_l 2^l$ összeg zérus, azaz $(p-2)$ db k_l együttható zérus. Ezen feltételek mellett az x_{n-p-1} kifejezés lehetséges értékeinek száma 2^{n-p+1} ($n-p$ db i_l ($0 \le l \le n-p-1$) változó és a k_{p-2} változó értékei). Azaz az n – p-ik összegzésben a szükséges komplex szorzások száma:

(3-12) $$M_{n-p} = N - \frac{N}{2} - 2^{n-p+1} = \frac{N}{2}\left(1 - \frac{1}{2^{p-2}}\right).$$

A (3-9) összefüggés nyilván csak $p \ge 3$-ra érvényes, mivel az első két összefüggésben csak a DFT-k, ill. a triviális forgatás szerepel. Az egyes fokozatokbeli szorzások számát összegezve:

(3-13) $$M_N = \frac{N}{2}\sum_{p=3}^{n}\left(1-\frac{1}{2^{p-2}}\right) = \frac{N}{2}\log_2 N - \frac{3N}{2} + 2,$$

ill. komplex szorzásonként 3 valós szorzást és 5 valós összeadást számolva:

(3-14) $$f(N) = 3\cdot M_N = \frac{3N}{2}\log_2 N - \frac{9N}{2} + 6.$$

A szükséges összeadások száma: fokozatonként N, amihez jön a forgatások miatti szorzásokból adódó összeadások száma:

(3-15) $$A(N) = 2N\log_2 N + \frac{5f(N)}{3} = \frac{9N}{2}\log_2 N - \frac{15N}{2} + 10.$$

Hasonló gondolatmenetet alkalmazva $N=4^n$ alakú pontszámokra 4 szerinti faktorizációval (a 4-pontos DFT számírásához sincs szükség szorzásokra, azok csak a forgatásokat leíró exponenciális tagokkal kapcsolatosak), a szükséges komplex szorzások száma:

(3-16) $$M_N = \sum_{p=2}^{n}\left(\frac{9N}{16} - \frac{N}{4^p}\right) = \frac{9N}{16}\log_4 N - \frac{31N}{48} + \frac{1}{3},$$

amiből:

(3-17) $$f(N) = 3M_N = \frac{27N}{16}\log_4 N - \frac{31N}{16} + 1.$$

Az összeadások száma az előzőekhez hasonlóan adódik:

(3-18) $$A(N) = 2N\log_4 N + \frac{5}{3}f(N) = \frac{67N}{16}\log_4 N - \frac{155N}{48} + \frac{5}{3}.$$

A (3-17) és (3-18) kifejezések csak $N \ge 16$ értékekre érvényesek.

3.3. *A fokozatos részekre osztás eredményének értelmezése*

A fokozatos részekre osztás egyúttal azt is jelentette, hogy az (1-1) definíciós összefüggésben az i és k indexeket több indexváltozó lineáris kombinációjára bontottuk. Hatása: a kiindulási egydimenziós pontsorozatot több indexszel elérhető többdimenziós pontsorozattá alakítottuk át, aminek következtében a számításokhoz szükséges adatsorrend eltérhet az eredeti adatsorrendtől. Ugyanaz áll a transzformált értékeket tartalmazó sorozatra is. Következésképp a DFT meghatározásához a hagyományos aritmetikai műveleteken kívül (szorzás, összeadás, kivonás, léptetés) más műveletekre (adatmozgatás) is szükség lehet, amelyek a gyakorlati megvalósítás során esetleg nem elhanyagolhatóak. Léteznek olyan eljárások is, amelyek a fokozatos részekre osztást úgy szervezik, hogy a számításokhoz az adatok eredeti sorrendjére van szükség, és az eredmény is az eredeti sorrendben képződik. A továbbiakban az átrendezés problémáját azonban részletesen nem vizsgáljuk.

A (3 – 3) összefüggés másképp is értelmezhető. Mint arról már korábban szó volt, a belső összeg N_1 db p-pontos DFT számítását jelenti. Azonos átalakítás után (3 – 3) új formája:

(3-19) $$X(k_1, k_2) = \sum_{i_1=0}^{N_1-1} e^{-j\left(\frac{2\pi}{N_1}\right) i_1k_1} \cdot e^{-j\left(\frac{2\pi}{N}\right) i_1k_2} \cdot \sum_{i_2=0}^{P-1} x(i_1, i_2) e^{-j\left(\frac{2\pi}{p}\right) i_2k_2}.$$

Látható, hogy a külső összeg az $e^{-j\left(\frac{2\pi}{N}\right) i_2k_2}$ szorzótényezőtől eltekintve N_1-pontos DFT-t definiál, azaz a (3 – 19) kifejezés számításának algoritmusa:

1. N_1 db P-pontos DFT számítása;
2. a kapott $N_1 \cdot P = N$ érték szorzása a megfelelő $e^{-j\left(\frac{2\pi}{N}\right) i_1k_2}$ értékekkel ($0 \le i_1 \le N_1 - 1$, $0 \le k_2 \le P - 1$), azaz egy forgatást kell végrehajtani a komplex számsíkban;
3. a külső összeg által meghatározott P db N_1-pontos DFT meghatározása.

A számítások 2. lépésében szükséges szorzások nélkül az eredetileg $1-D$ DFT számítását $2-D$ DFT számítására lehetne visszavezetni. A szükséges szorzások száma:

$$(3\text{-}20) \quad f(N)=N_1 \cdot f(P)+P \cdot f(N_1)+N=\frac{N}{P} \cdot f(P)+ + \frac{N}{N_1} \cdot f(N_1)+N.$$

A kiinduló feltevések szerint $P=N_2 \cdot N_3 \cdot \ldots \cdot N_n$, vagyis a fokozatos részekre osztás tovább folytatható a (3-19) kifejezésben szereplő belső összeg (P-pontos DFT) felbontásával. A korábbiakban követett eljárás ismételt alkalmazásával adódik [24]:

$$(3\text{-}21) \quad f(N)=\sum_{j=1}^{n} \frac{N}{N_j} \cdot f(N_j)+(\mathrm{n}-1)N.$$

A felbontási módszerből következik, hogy a fokozatonként szükséges korrekciós tagtól eltekintve (2. lépés) lényegében az $1-D$ DFT $n-D$ DFT-be való átalakítását kaptuk, azaz az eredeti nagy méretű feladatot valóban több, kisebb méretű, azonos típusú feladattá sikerült felbontani. A részekre osztást általában célszerű prímtényezős alakig elvégezni, bár egyes esetekben a gyakorlati követelmények más felbontás alkalmazását tehetik szükségessé.

Elemezzük röviden a $(3-21)$ összefüggés jelentését. Az első tag jelenti az $N_1 \times N_2 \times \ldots \times N_n$ $\mathrm{n}-D$ DFT számításának műveletigényét, ha $f(N_j)$ az N_j-pontos $1-D$ DFT számításigénye. Ha $f(N_j)=0(N_j^2)$, akkor a $(3-5)$ összefüggés származtatásánál alkalmazott gondolatmenet kisebb műveletszámra vezet (nem bontjuk fel a (3-3) összefüggés külső összegében az exponenciális tagot két exponenciális tag szorzatára):

$$(3\text{-}22) \quad f(N_j)=3N_j^2 \rightarrow f(N)=\sum_{j=1}^{n} \frac{N}{N_j} \cdot 3N_j^2+(n-1)N= =3N(N_1+N_2+\ldots+N_n)+(n-\mathrm{i})N.$$

Ha azonban $f(N_j)<0(N_j^2)$, akkor remélhető, hogy a (3-21) a (3-5) egyenlőséggel meghatározott műveletszámnál kevesebbre vezet. Amennyiben $N=2^n$, akkor:

$$(3\text{-}23) \quad f(N)=\mathrm{n}2^{n-1}f(2)+(n-1)2^n.$$

Mivel $f(2)=0$ $(X(0)=x(0)+x(1),\ X(1)=x(0)-x(1))$,

$$(3\text{-}24) \quad f(N=2^n)=(n-1)2^n=N(\log_2 N-1)= =0(N \cdot \log_2 N),$$

mint azt az előzőekben meghatároztuk.

Kiindulva ismét az $N=\prod_{i=1}^{n} \mathrm{p}_i^{\alpha_i}$ törzstényezős alakból és N értékét a lehető legtöbb tényezőre bontva:

$$(3\text{-}25) \quad f(N)=N\sum_{j=1}^{n} \frac{\alpha_j}{p_j} f(p_j)+N\left(\sum_{j=1}^{1} \alpha_j-1\right).$$

Ha $N=\mathrm{p}$ (p prím), akkor $f(N)=f(\mathrm{p})$, azaz a fenti módon történő fokozatos részekre osztással nem érhetünk el eredményt (egyébként is kiindulási feltétel volt, hogy N összetett szám). $N=\mathrm{p}$ esetén;

$$(3-26) \quad f(N=\mathrm{p}^\alpha)=\alpha \mathrm{p}^{\alpha-1}f(\mathrm{p})+(\alpha-1)\mathrm{p}.$$

Egyszerű átalakításokkal kimutatható, hogy $N>4$ esetben $f(N)<0(N^2)$ még akkor is, ha $f(\mathrm{p})=0(\mathrm{p}^2)$.

(3-25)-ből az $f(N)$ bonyolultsági mérték (valós szorzások száma) csökkentésének két kézenfekvő módja adódik. Egyrészt visszavezetni az $1-D$ DFT számítását valóban $\mathrm{n}-D$ DFT meghatározására: ekkor ugyanis a (3-21) összegben az $(n-1)N$ alakú második tag elmarad. Ennek feltétele, hogy a számítási algoritmus 2. lépésében az exponenciális taggal való szorzásra (forgatásra) ne legyen szükség. A másik lehetőség az $f(\mathrm{p}_j)$ értékek csökkentése: prímszámokra heurisztikus vagy szisztematikus úton olyan algoritmusokat származtatni, amelyekre $f(\mathrm{p}_i)<$ $<0(p_i^2)$. A továbbiakban megvizsgáljuk, mi a feltétele, hogy az $1-D$ DFT $n-D$ DFT-be legyen átalakítható.

(A cikk ugyanezen folyóirat későbbi számaiban folytatódik.)

IRODALOM

[1] *Aho, A. V.—Hopcroft, J. E.—Ullmann, J. D.:* "The Design and Analysis of Computer Algorithms", Addison-Wesley Publishing Company, (1975).

[2] *Ahmed, N.—Rao, K. R.:* "Orthogonal Transforms for Digital Signal Processing", Springer-Verlag, Berlin and New York, (1975).

[3] *Agarwal, R. C.—Burrus, C. S.:* "Fast One-Dimensional Convolution by Multidimensional Techniques", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-22, pp. 1—10, (1974).

[4] *Agarwal, R. C.—Burrus, C. S.:* "Number Theoretic Transforms to Implement Fast Digital Convolution", Proceedings of the IEEE, vol. 63, pp. 550—560, (1975).

[5] *Agarwal, R. C.—Cooley, J. C.:* "New Algorithms for Digital Convolutions", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-25, pp. 392-410, (1977).

[6] *Agarwal, R. C.:* "An In-Place and In-Order WFTA", ICASSP '83, Boston, pp. 190—193, (1983).

[7] *Bellman, R.:* "Introduction to Matrix Analysis", McGraw-Hill, New York, (1960).

[8] *Burrus, C. S.:* "Index Mappings for Multidimensional Formulation of the DFT and Convolution", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-25, pp. 239—242, (1977).

[9] *Bringham, E. O.:* "The Fast Fourier Transform", Prentice-Hall, Englewood Cliffs, New Jersey, (1974).

[10] *Czebe, L.:* „A diszkrét és a gyors Fourier-transzformáció", Híradástechnika, vol. 32., no. 4., pp. 141—155, (1981).

[11] *Cooley, I. W.—Tukey, I. W.:* "An Algorithm for the Machine Calculation of Complex Fourier Series", Mathematics of Computation, vol. 19., pp. 297—301, (1965).

[12] *Good, I. J.:* "The Relationship Between Two Fast Fourier Transforms", IEEE Transactions on Computers, C-20, pp. 310—317, (1917).

[13] *Kolba, D. P.—Parks, I. W.:* "A Prime Factor FFT Algorithm Using High-Speed Convolution", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-25, pp. 281—294, (1977).

[14] *Knuth, D. E.:* "The Art of Computer Programming", vol. 1. "Fundamental Algorithms", és vol. 2. "Seminumerical Algorithms", Addison-Wesly, Reading, Massachusetts, (1969).

[15] *Nawab, H.—McClellan, J. H.:* "A Comparison of WFTA and FFT Programs", Proceedings of Asilomar Conference on Circuits, Systems and Computers, Pacific Grove, California, pp. 613—617, (1978).
[16] *Nawab, H.—McClellan, J. H.:* "Bounds on the Minimum Number of Data Transfers in WFTA and FFT Programs", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-27, pp. 394—398, (1979).
[17] *Niven, I.—Zuckerman, H. S.:* "Bevezetés a számelméletbe", Műszaki Könyvkiadó, Budapest, (1978).
[18] *Nussbaumer, H. J.—Quandalle, P.:* "Comparison of Convolutions and Discrete Fourier Transforms", IBM Journal of Research and Development, vol. 22, pp. 134—144, (1978).
[19] *Nussbaumer, H. J.—Quandalle, P.:* "Fast Compatition of Discrete Fourier Transforms Using Polynominal Transforms", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-27, pp. 169—181, (1979).
[20] *Nussbaumer, H. J.:* "Fast Polynominal Transform Algorithms for Digital Convolution", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-28, pp. 205—215, (1980).
[21] *Oppenheim, A. V.—Schafer, R. W.:* "Digital Signal Processing", Prentice-Hall, Englewood Cliffs, New Jersey, (1975).
[22] *Rabiner, L. R.—Gold, B.:* "Theory and Application of Digital Signal Processing", Prentice-Hall, Englewood Cliffs, New Jersey, (1975).
[23] *Rader, C. M.:* "Discrete Fourier Transforms When the Number of Data Samples is Prime", Proceedings of the IEEE, vol. 56., pp. 1107—1108, (1968).
[24] *Rader, C. M.—McClellan, J. H.:* "Number Theory in Digital Signal Processing", Prentice-Hall, Englewood Cliffs, New Jersey, (1979).
[25] *Silverman, H. F.:* "An Introduction to Programming the Winograd Fourier Transform Algorithm (WFTA)", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-25, pp. 152—165, (1977).
[26] *Toom, A. L.:* "The Complexity of a Scheme of Functional Elements Simulating the Multiplication of Integers", Dokladi Akademii Nauk SSSR, pp. 496—498, (1963).
[27] *Winograd, S.:* "Some Bilinear Forms Whose Multiplicative Complexity Depends on the Field of Constants", Mathematical Systems Theory, vol. 10., pp. 169—180, (1977).
[28] *Winograd, S.:* "On Computing the Discrete Fourier Transform", Mathematics of Computation, vol. 32., pp. 175—199, (1978).
[29] *Winograd, S.:* "Signal Processing and Complexity of Computation", Proceedings of International Conference on Acoustics, Speech, and Signal Processing, pp. 94—101, (1980).
[30] *Zohar, S.:* "A Prescription of Winograd's Discrete Fourier Transform Algorithm", IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-27, pp. 409—421, (1979).
[31] *Gordos, G.—Takács, Gy.:* „Digitális beszédfeldolgozás", Műszaki Könyvkiadó, Budapest, 1983, p. 345.