ISSN 1392 – 1215 ELEKTRONIKA IR ELEKTROTECHNIKA, 2004. Nr. 2(58)

T 121 SIGNALŲ TECHNOLOGIJA

Lygiagrečiojo diskretinės Furjė transformacijos algoritmo analizė

K. Kazlauskas

Atpažinimo procesų skyrius, Matematikos ir informatikos institutas,
Akademijos g. 4, LT-08663 Vilnius, Lietuva, tel. +370 5 2109319, el. p. kazlausk@ktl.mii.lt

Įvadas

Diskretinė Furjė transformacija (DFT) yra vienas iš svarbiausių skaitmeninio signalų apdorojimo metodų. Tačiau šios transformacijos skaičiavimų apimtis gana didelė. Jei seka turi N atskaitų, tai DFT vienai atskaitai apskaičiuoti reikia atlikti N kompleksinių sandaugų ir N–1 kompleksinių sudėčių, o visoms N atskaitoms apskaičiuoti – maždaug po N^2 kompleksinių sandaugų ir sudėčių. Sandaugų ir sudėčių skaičius gana gerai atspindi apskaičiavimų sudėtingumą. Taigi tiesioginiu būdu skaičiuojant sekos iš N atskaitų DFT, reikia sugaišti laiką, proporcingą N^2. Esant dideliems N, skaičiavimų sudėtingumas dažnai yra pagrindinė kliūtis panaudoti DFT signalams apdoroti. Ekonomiškas DFT skaičiavimas yra sparčiosios Furjė transformacijos (SFT) algoritmas. Panaudojant SFT, skaičiavimų sudėtingumas įvertinamas ne N^2, bet $N\log_2 N$. Pavyzdžiui, kai N=1024, su SFT apskaičiavimai atliekami net šimtą kartų sparčiau [1].

Šiame straipsnyje analizuojamas lygiagretusis diskretinės Furjė transformacijos (LDFT) algoritmas, kuris gerokai sumažina operacijų skaičių bei padidina diskretinės Furjė transformacijos apskaičiavimo spartą.

Lygiagretusis DFT (LDFT) algoritmas

Sekos $x(n)$, n=0,1,…, N–1 DFT [2]

$$X(k)=\sum_{n=0}^{N-1} x(n)e^{-i\frac{2\pi}{N}nk},\ k=0,1,\ldots,N-1. \qquad (1)$$

Panaudokime blokinių algoritmų sudarymo metodą [3–5]. Tarkim, kad

$$k=k_1\frac{N}{M}+k_2,\ n=n_1M+n_2,$$
$$k_1,n_2=0,1,...,M-1;\qquad n_1,k_2=0,1,\ ...,\frac{N}{M}-1.$$

N, M ir $\frac{N}{M}$ – sveikieji skaičiai. Įrašykime k ir n vertes į (1) išraišką. Tada

$$X(k_1\frac{N}{M}+k_2)=$$
$$=\sum_{n_2=0}^{M-1}\sum_{n_1=0}^{\frac{N}{M}-1} x(n_1M+n_2)e^{-i\frac{2\pi}{N}(n_1M+n_2)(k_1\frac{N}{M}+k_2)}=$$
$$=\sum_{n_2=0}^{M-1}\sum_{n_1=0}^{\frac{N}{M}-1} x(n_1M+n_2)\ \times$$
$$\times\ e^{-i2\pi n_1k_1}e^{-i\frac{2\pi}{N/M}n_1k_2}e^{-i\frac{2\pi}{M}k_1n_2}e^{-i\frac{2\pi}{N}k_2n_2}.$$

Kadangi $e^{-i2\pi n_1k_1}=1$, todėl

$$X(k_1\frac{N}{M}+k_2)=$$
$$=\sum_{n_2=0}^{M-1}e^{-i\frac{2\pi}{N}k_2n_2}\times$$
$$\times\ (\sum_{n_1=0}^{\frac{N}{M}-1} x(n_1M+n_2)e^{-i\frac{2\pi}{N/M}n_1k_2})e^{-i\frac{2\pi}{M}k_1n_2}. \qquad (2)$$

DFT lygiagrečiai apskaičiuojama taip:

1. Iš sekos $x(n)$, n=0,1,…, N–1 atskaitų suformuojama $\left(\frac{N}{M}\times M\right)$- matė matrica $\boldsymbol{x}=\{x(n_1M+n_2),$ $n_1=0,1,...,\frac{N}{M}-1;\ n_2=0,1,...,M-1\}$, kurios pirmoji eilutė – $x(0)$, …, $x(M-1)$, antroji eilutė – $x(M)$, …, $x(2M-1)$ ir t. t.

2. M DFT arba SFT procesoriai lygiagrečiai apskaičiuoja matricos $\boldsymbol{x}$ M stulpelių diskretines Furjė transformacijas. Pirmasis procesorius apskaičiuoja pirmojo stulpelio $\frac{N}{M}$ atskaitų DFT, antrasis – antrojo stulpelio $\frac{N}{M}$ atskaitų DFT ir t. t. Gauname $\left(\frac{N}{M}\times M\right)$- matę matricą

$$\boldsymbol{y}=\{y(k_2,n_2),\ k_2=0,1,...,\frac{N}{M}-1,$$

$n_2 = 0,1,...,M-1$;

čia $y(k_2,n_2) = \sum_{n_1=0}^{\frac{N}{M}-1} x(n_1M+n_2)e^{-i\frac{2\pi}{N/M}n_1k_2}$.

3. Matricos $\boldsymbol{y}$ kiekvienas (k_2, n_2)-asis elementas dauginimo bloke dauginamas iš $e^{-i\frac{2\pi}{N}k_2n_2}$. Visos eilutės dauginamos tuo pat metu. Suformuojama $\left(\frac{N}{M}\times M\right)$-matė matrica $\boldsymbol{z} = \{z(k_2,n_2)\}$, kur

$$z(k_2,n_2) = e^{-i\frac{2\pi}{N}k_2n_2}\ y(k_2,n_2).$$

4. $\frac{N}{M}$ DFT arba SFT procesoriai lygiagrečiai apskaičiuoja matricos $\boldsymbol{z}$ $\frac{N}{M}$ eilučių diskretines Furjė transformacijas: pirmasis procesorius apskaičiuoja pirmosios eilutės M atskaitų DFT, antrasis – antrosios eilutės M atskaitų DFT ir t. t., o $\frac{N}{M}$-asis procesorius – $\frac{N}{M}$-osios eilutės M atskaitų DFT. Suformuojama $(M\times\frac{N}{M})$-matė matrica

$$X = \{\ X(k_1\frac{N}{M}+k_2),$$

$$k_1 = 0,1,...,M-1;\quad k_2 = 0,1,...,\frac{N}{M}-1\ \},$$

čia $X(k_1\frac{N}{M}+k_2) = \sum_{n_2=0}^{M-1} z(k_2,n_2)e^{-i\frac{2\pi}{M}k_1n_2}$,

kurios pirmojoje eilutėje yra DFT vertės $X(0),...,X(\frac{N}{M}-1)$, antrojoje eilutėje yra DFT vertės $X(\frac{N}{M}),...,X(\frac{2N}{M}-1)$ ir t. t.

5. Išskleidę matricą X eilutėmis, gauname sekos $x(n)$ DFT vertes $X(k)$, k=0,1,…, N–1.

Lygiagrečiojo DFT algoritmo analizė

Bendruoju atveju M gali būti bet koks sveikasis skaičius, tačiau turi tenkinti sąlygą, kad N/M taip pat būtų sveikasis skaičius. Nuo M priklauso LDFT algoritmo operacijų skaičius. Apskaičiuokime optimalų M, t. y. tokį, kad LDFT algoritmo operacijų skaičius būtų mažiausias.

Suraskime LDFT algoritmo operacijų skaičių. Pirmiausia apskaičiuojama matricos $\boldsymbol{x}$ M stulpelių $\frac{N}{M}$ atskaitų DFT. Tam reikia $\left(\frac{N}{M}\right)^2\cdot M$ kompleksinių daugybos ir sudėties operacijų. Po to kiekvienas matricos $\boldsymbol{y}$ elementas dauginamas iš $e^{-i\frac{2\pi}{N}k_2n_2}$. Tam prireiks N dauginimo operacijų. Pagaliau apskaičiuojamos matricos $\boldsymbol{z}$ $\frac{N}{M}$ eilučių M atskaitų diskretinės Furjė transformacijos. Tam reikės atlikti $M^2\cdot\frac{N}{M}$ operacijas. Visas LDFT algoritmo operacijų skaičius lygus

$$\frac{N^2}{M} + MN + N. \qquad (3)$$

Tarkime, kad M – tolydusis dydis. Tuomet, apskaičiavę (3) išraiškos išvestinę pagal M ir prilyginę gautą rezultatą nuliui, gauname optimalią M vertę $M_{opt} = \sqrt{N}$. Tačiau N yra sveikasis skaičius ir turi būti toks, kad M_{opt} būtų sveikasis skaičius. Be to, jei eilučių ir stulpelių DFT apskaičiuoti naudosime SFT algoritmą, M turi būti lygus 2^n. Todėl $N_{opt} = 2^{2n}$, kur n – sveikasis teigiamas skaičius. Matricos $\boldsymbol{x}$ eilučių skaičius lygus $\frac{N}{M}$, o optimalus eilučių skaičius lygus $\frac{N}{\sqrt{N}} = \sqrt{N}$.

Išvada. Matricos $\boldsymbol{x}$ optimalus dydis yra $\sqrt{N}$ eilutės ir $\sqrt{N}$ stulpeliai.

Įrašę vertę $M_{opt} = \sqrt{N}$ į (3) išraišką, gauname LDFT algoritmo optimalų operacijų skaičių $2N\sqrt{N} + N$.

Jei sekos $x(n)$, n=0,1,…,N – 1 DFT skaičiuotume lygiagrečiai, o matricos $\boldsymbol{x}$ eilučių ir stulpelių DFT apskaičiuoti naudotume SFT procesorius (LSFT algoritmas), tada optimalus LSFT algoritmo operacijų skaičius būtų lygus $N\log_2 N + N$.

Kai DFT realizuojama lygiagrečiai, naudojami $\sqrt{N}$ DFT (LDFT algoritmas) arba $\sqrt{N}$ SFT (LSFT algoritmas) procesoriai. Abiem atvejais vieną kartą skaičiuojama matricos $\boldsymbol{x}$ kiekvieno stulpelio DFT ir sugaištamas laikas yra proporcingas vieno stulpelio operacijų, kurios reikalingos DFT apskaičiuoti, skaičiui. LDFT algoritmo atveju matricos $\boldsymbol{x}$ vieno stulpelio DFT apskaičiuoti reikės N operacijų. Naudojant LSFT algoritmą, matricos $\boldsymbol{x}$ vieno stulpelio DFT apskaičiuoti reikės $\sqrt{N}\log_2\sqrt{N}$ operacijų. Antrą kartą kiekvienas DFT (arba SFT) procesorius apskaičiuoja matricos $\boldsymbol{z}$ vienos eilutės DFT ir atlieka tiek pat operacijų kaip ir pirmuoju atveju.

Be to, $(\sqrt{N}\times\sqrt{N})$–matės matricos $\boldsymbol{y}$ kiekvienas elementas dauginamas iš kompleksinio skaičiaus. Kai dauginimui naudojami $\sqrt{N}$ dauginimo įrenginiai, daugyba atliekama per laiką, proporcingą $\sqrt{N}$.

Laikas sugaištas DFT apskaičiuoti, jei naudojami $\sqrt{N}$ DFT procesoriai ir $\sqrt{N}$ dauginimo įrenginiai (LDFT algoritmas), proporcingas operacijų skaičiui $2N + \sqrt{N}$. Laikas, reikalingas DFT apskaičiuoti, jei naudojami $\sqrt{N}$ SFT procesoriai ir $\sqrt{N}$ dauginimo įrenginiai (LSFT algoritmas), proporcingas operacijų skaičiui $2\sqrt{N}\log_2\sqrt{N} + \sqrt{N}$. Pirmojoje lentelėje yra DFT, SFT, LDFT ir LSFT algoritmų kompleksinių sandaugų ir sudėčių skaičiai, kai M=$\sqrt{N}$.

1 lentelė. DFT, SFT, LDFT ir LSFT algoritmų kompleksinių sandaugų ir sudėčių skaičiai, kai $M=\sqrt{N}$

Sekos ilgis N	$\sqrt{N}$	DFT (N^2)	SFT ($N\log_2 N$)	LDFT ($2N\sqrt{N}+N$)	LSFT ($N\log_2 N+N$)
16	4	256	64	144	80
64	8	4096	384	1088	448
256	16	65536	2048	8448	2304
1024	32	1048576	10240	66560	11264
4096	64	16777216	49152	528384	53248
16384	128	268435456	229376	4210688	245760
65536	256	4.295×10^9	1048576	33619968	1114112

2 lentelė. DFT, SFT, LDFT ir LSFT algoritmų spartos palyginimas, kai $M=\sqrt{N}$

N	$\sqrt{N}$	SFT/LSFT	DFT/LDFT
16	4	3	7
64	8	7	30
256	16	14	124
1024	32	29	504
4096	64	59	2032
16384	128	119	8160
65536	256	241	32704
262144	512	485	130944

Antrojoje lentelėje pateikti algoritmų operacijų skaičių santykiai:

$$SFT / LSFT = \frac{N\log_2 N}{2\sqrt{N}\log_2\sqrt{N}+\sqrt{N}},$$

$$DFT / LDFT = \frac{N^2}{2N+\sqrt{N}}.$$

DFT, SFT, LDFT ir LFST algoritmų spartumas yra proporcingas atliekamų operacijų skaičiui: kuo operacijų mažiau, tuo algoritmas spartesnis.

Išvados

Apskaičiuojant sekos $x(n)$ diskretinę Furjė transformaciją LSFT algoritmu, kai eilučių ir stulpelių diskretinėms Furjė transformacijoms apskaičiuoti naudojami $\sqrt{N}$ SFT procesoriai, reikia atlikti N kompleksinių daugybos operacijų daugiau, palyginti su SFT algoritmu (1 lentelė), be to, reikalingi ne vienas, bet $\sqrt{N}$ SFT procesoriai. Tačiau šie procesoriai yra paprastesni, nes kiekvienas apskaičiuoja ne N, bet $\sqrt{N}$ atskaitų DFT. LSFT algoritmo aparatinė realizacija, palyginti su SFT algoritmo aparatine realizacija, yra gerokai spartesnė. Pavyzdžiui, kai N=1024 ir naudojami 32 SFT 32 atskaitų procesoriai, LSFT algoritmo realizacija yra 29 kartus spartesnė palyginti su SFT algoritmo realizacija (2 lentelė). Realizuojant LSFT algoritmą reikia, kad sekos $x(n)$ atskaitų skaičius $N=2^{2n}$, n=1,2,… . Jei N yra kitoks, tai seką $x(n)$ reikia papildyti nuliais, kad ši sąlyga būtų tenkinama.

Literatūra

1. **Krivickas R.** Skaitmeninis signalų apdorojimas. – Vilnius: Mokslas, 1984. – 126 p.
2. **DeFatta D. J., Lucas J. G. and Hodgkiss W. S.** Digital Signal Processing: A System Design Approach. – New York: John Wiley & Sons, 1988. – 661 p.
3. **Kung S. Y., Whitehouse H. J. and Kailath T.** VLSI and Modern Signal Processing. – New Jersey: Prentice- Hall, 1985. – 472 p.
4. **Казлаускас К., Пупейкис Р.** Цифровые системы обработки данных.–Вильнюс: Мокслас, 1991. – 218 с.
3. 5. **Foster I**. Designing and building parallel programs. Addison Wessley, 1995. – 382 p.

Pateikta spaudai 2004 12 22

K. Kazlauskas. Lygiagrečiojo diskretinės Furjė transformacijos algoritmo analizė // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2005. – Nr. 2(58). – P. 57–60.

Diskretinė Furjė transformacija labai dažnai naudojama sprendžiant įvairius skaitmeninio signalų apdorojimo uždavinius. Straipsnyje analizuojamas lygiagretusis diskretinės Furjė transformacijos algoritmas, kuris, palyginti su tiesioginiu diskretinės Furjė transformacijos skaičiavimo algoritmu, gerokai sumažina operacijų skaičių bei paspartina diskretinės Furjė transformacijos apskaičiavimą. Šį algoritmą galima realizuoti lygiagrečiai naudojant sparčiosios Furjė transformacijos procesorius. Išnagrinėtas lygiagrečiojo diskretinės Furjė transformacijos algoritmo sudėtingumas. Parodyta, koks turi būti optimalus sparčiosios Furjė transformacijos procesorių skaičius. Nustatyta, kad lygiagrečioji diskretinės Furjė transformacijos algoritmo aparatinė realizacija yra sudėtingesnė, nes naudojamas ne vienas, bet keli procesoriai, tačiau gerokai spartesnė, palyginti su sparčiosios Furjė transformacijos nelygiagrečiąja aparatine realizacija. Bibl. 5 (lietuvių kalba; santraukos lietuvių, anglų ir rusų k.).

K. Kazlauskas. Parallel Discrete Fourier Transform Algorithm Analysis // Electronics and Electrical Engineering. – Kaunas: Technologija, 2005. – No. 2(58). – P. 57–60.

Discrete Fourier transform is widely used for digital signal processing. In the paper, a parallel discrete Fourier transform algorithm is analysed. The algorithm, as compared with the direct discrete Fourier transform algorithm, has a considerably smaller number of operations and increased processing speed. The parallel algorithm can be realised in a parallel manner using fast Fourier transform processors. The complexity of the parallel discrete Fourier transform algorithm is discussed. The optimal number of the fast Fourier transform processors is found. It is shown that the hardware realization of the parallel discrete Fourier transform algorithm is more complicated, because it uses more than one processor, but considerably faster as compared with nonparallel hardware realization of the fast Fourier transform algorithm. Bibl. 5 (in Lithuanian; summaries in Lithuanian, English, Russian).

К. Казлаускас. Анализ параллельного алгоритма дискретной Фурье трансформации // Электроника и электротехника. – Каунас: Технология, 2005. – № 2(58). – С. 57–60.

Дискретная Фурье трансформация очень часто используется для цифровой обработки сигналов. В статье анализуется параллельный алгоритм вычисления дискретной Фурье трансформации, который по сравнению с прямым вычислением дискретной Фурье трансформации, заметно уменьшает число операций и увеличивает скорость вычислений. Этот алгоритм можно реализировать параллельно с помощью процессоров быстрого преобразования Фурье. Проанализирована сложность параллельного алгоритма дискретной Фурье трансформации. Показано, каким должно быть оптимальное число процессоров быстрого преобразования Фурье. Установлено, что аппаратная реализация параллельного алгоритма дискретной Фурье трансформации является более сложной, так как используется не один, а несколько процессоров, однако параллельная реализация несколько раз увеличивает скорость вычисления дискретной Фурье трансформации по сравнению с непараллельной аппаратной реализацией быстрого преобразования Фурье. Библ. 5 (на литовском языке; рефераты на литовском, английском и русском яз.).