

ПРИМЕНЕНИЕ БЫСТРЫХ СВЕРТОЧНЫХ АЛГОРИТМОВ

Тимошенко Л.И.

Ставропольский филиал Краснодарского университета МВД России, Ставрополь,
e-mail: lit-545@yandex.ru

В современных и перспективных алгоритмах цифровой обработки сигналов широкое применение начали находить математические модели, использующие аппарат линейной алгебры, основными вычислительными процедурами являются операции типа перемножения векторов и матриц, обращение матриц, поиска собственных векторов и собственных значений матриц, решение систем линейных алгебраических уравнений.

Ключевые слова: цифровая обработка сигналов, реализация арифметических операций, показатели быстродействия, скорость обработки, алгоритмы ускоренного вычисления

APPLICATION OF FAST SVERTOCHNY ALGORITHMS

Timoshenko L.I.

Stavropol branch of the Ministry of Internal Affairs Krasnodar university of Russia, Stavropol,
e-mail: lit-545@yandex.ru

In modern and perspective algorithms of digital processing of signals the mathematical models using the device of linear algebra started finding broad application, the main computing procedures are operations like multiplication of vectors and matrixes, the address of matrixes, search of own vectors and own values of matrixes, the decision of systems of the linear algebraic equations.

Keywords: digital processing of signals, realization of arithmetic operations, speed indicators, processing speed, algorithms of the accelerated calculation

Для достижения желаемого эффекта по обеспечению вычислений при стандартной разрядной сетке процессора цифровой обработки сигналов (ЦОС) были предложены алгоритмы коротких сверток, основанные на специальных способах умножения номиналов, которые можно использовать в соответствии с китайской теоремой об остатках (КТО) для вычисления больших сверток, заменяя их на последовательность коротких. Данные методы составляют основу второй группы методов цифровой обработки сигналов [1, с. 60-97., 2].

Если исходные две последовательности дискретных отчетов $x(nT)$ и $u(nT)$ представить в виде полиномиальной формы, то операцию циклического свертывания можно свести к процедуре умножения этих полиномов.

При этом вычисления циклической свертки двух последовательностей $x(nT)$ и $u(nT)$ равной длины N , эквивалентного нахождению коэффициентов полинома $y(z)$ согласно выражения

$$y(z) \equiv x(z)u(z) \bmod z^N - 1, \quad (1)$$

где $x(z)$ и $u(z)$ – полиномиальная форма последовательностей $x(nT)$ и $u(nT)$ соответственно.

Если положить, что двучлен $z^N - 1$ может быть представлен в виде произведения S круговых неприводимых полиномов

$$z^N - 1 = \prod_{i=1}^S p_i(z), \quad (2)$$

где S – число делителей N , то это позволяет использовать для вычисления коротких сверток КТО. В этом случае вычисление свертки (1) выполняется согласно алгоритма:

Приведение полиномов $x(z)$ и $u(z)$ по модулю $p_i(z)$, $i=1, 2, \dots, S$, т.е.

$$\begin{aligned} x_i(z) &\equiv x(z) \bmod p_i(z), \\ u_i(z) &\equiv u(z) \bmod p_i(z). \end{aligned} \quad (3)$$

2. Определение произведения

$$y_i(z) = x_i(z)u_i(z) \bmod p_i(z).$$

Восстановление значений $y(z)$ с помощью КТО, т.е.

$$y(z) = \sum_{i=1}^S y_i(z)B_i(z) \bmod z^N - 1. \quad (4)$$

где $B_i(z)$ – ортогональный базис i -го основания.

Если в качестве коэффициентов полиномов $x(z)$ и $u(z)$ выбрать элементы поля рациональных чисел Q , то рассматриваемый выше алгоритм будет иметь минимальную сложность. Это обусловлено тем, что в поле рациональных чисел круговые полиномы

с индексами $N < 105$ имеют простые коэффициенты $\{-1, 0, 1\}$

В этом случае обобщенная форма алгоритмов вычисления коротких сверток с помощью КТО имеет вид

$$\overline{y} = C(\overline{Bx} \bullet \overline{Au}), \quad (5)$$

где $\overline{y}, \overline{x}, \overline{u}$ – векторная форма выходной, входной и сворачивающей последовательностей соответственно; C, B, A , – матрицы; « $\bullet$ » – поточечное произведение векторов.

Значения элементов матриц C, B, A для коротких сверток длиной $N < 9$, обеспечивающих минимальное количество операций умножения, приведены в работе [7, с. 76, 8, с. 38-39, 9, с. 77]. Альтернативным решением проблемы сокращения времени вычисления циклических сверток является гнездовой алгоритм.

Вычисление циклической свертки непосредственно по выражению (5) требует разработки специальных алгоритмов, минимизирующих число операций умножения для каждого значения N . Лучшие оценки по бы-

стродействию вычислительных устройств получаются при использовании гнездового алгоритма [4, с. 60-97].

В основу данного алгоритма положена возможность сведения вычисления одномерной свертки к многомерной. Отправной точкой такого преобразования является изоморфизм, порождаемый китайской теоремой об остатках [5, с. 73-74, 14].

Если длина циклической свертки N может быть представлена в виде произведения двух чисел n_1 и n_2 , т. е. $N = n_1 n_2$ причем НОД $(n_1, n_2) = 1$, то любое число из интервала $0, 1 \dots N-1$ можно записать в виде (m_1, m_2) , где $m_1 = N \bmod n_1, m_2 = N \bmod n_2$. Тем самым устанавливается взаимнооднозначное соответствие между элементами кольца N и целыми числами отрезка $\Omega = [0, N-1]$. При этом вычет m_1 рассматривается как первая координата числа N , а вычет m_2 – как вторая координата.

Рассмотрим матричную форму представления операции циклической свертки. В этом случае выражение (1) имеет вид

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} u_0 & u_1 & u_2 & \dots & u_{N-1} \\ u_1 & u_2 & u_3 & \dots & u_0 \\ u_2 & u_3 & u_4 & \dots & u_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{N-1} & u_0 & u_1 & \dots & u_{N-2} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}. \quad (6)$$

Тогда циклическая свертка определяется в виде произведения $N \times N$ циклической матрицы свертки на сворачиваемый вектор. Если строки матрицы упорядочить по первой координате m_1 , а столбцы – по второй, то получим разбиение по блокам, каждый из которых также образует циклическую матрицу для свертки меньшей размерности.

$$\begin{bmatrix} y_{0,0} \\ y_{0,1} \\ \vdots \\ y_{0,n_2-1} \\ \vdots \\ y_{n_1-1,0} \\ y_{n_1-1,1} \\ \vdots \\ y_{n_1-1,n_2-1} \end{bmatrix} = \begin{bmatrix} u_{0,0} & u_{0,1} & \dots & u_{0,n_2-1} & \dots & u_{n_1-1,0} & u_{n_1-1,1} & \dots & u_{n_1-1,n_2-1} \\ u_{0,1} & u_{0,2} & \dots & u_{0,0} & \dots & u_{n_1-1,1} & u_{n_1-1,2} & \dots & u_{n_1-1,0} \\ \vdots & \vdots & & & & \vdots & & & \\ u_{0,n_2-1} & u_{0,0} & \dots & u_{0,n_2-2} & \dots & u_{n_1-1,n_2-1} & u_{n_1-1,0} & \dots & u_{n_1-1,n_2-2} \\ \vdots & \vdots & & & & & & & \\ u_{n_1-1,0} & u_{n_1-1,1} & & u_{n_1-1,n_2-1} & \dots & u_{0,0} & u_{0,1} & \dots & u_{0,n_2-1} \\ u_{n_1-1,1} & u_{n_1-1,2} & & u_{n_1-1,0} & & u_{0,1} & u_{0,2} & \dots & u_{0,0} \\ \vdots & \vdots & & & & \vdots & & & \\ u_{n_1-1,n_2-1} & u_{n_1-1,0} & & u_{n_1-1,n_2-2} & & u_{0,n_2-1} & u_{0,0} & \dots & u_{0,n_2-2} \end{bmatrix} \begin{bmatrix} x_{0,0} \\ x_{0,1} \\ \vdots \\ x_{0,n_2-1} \\ \vdots \\ x_{n_1-1,0} \\ x_{n_1-1,1} \\ \vdots \\ x_{n_1-1,n_2-1} \end{bmatrix}$$

Перестановки строк и столбцов циклической матрицы осуществляются в следующем порядке: первыми ставятся те, у которых первая координата равна 0, а вторая – в возрастающем порядке, затем те, у которых первая координата равна 1 и вторая в возрастающем порядке. В результате получается блочно-циклическая структура.

Обозначив через $\overline{y_0} = (y_{0,0}, y_{0,1}, \dots, y_{0,n_2-1}), \dots, \overline{y_{n_1-1}} = (y_{n_1-1,0}, y_{n_1-1,1}, \dots, y_{n_1-1,n_2-1})$;

$\overline{x_0} = (x_{0,0}, x_{0,1}, \dots, x_{0,n_2-1}), \dots, \overline{x_{n_1-1}} = (x_{n_1-1,0}, x_{n_1-1,1}, \dots, x_{n_1-1,n_2-1})$, а также

$$u_0 = \begin{bmatrix} u_{0,0} & u_{0,1} & \dots & u_{0,n_2-1} \\ u_{0,1} & u_{0,2} & \dots & u_{0,0} \\ \vdots & & & \\ u_{0,n_2-1} & u_{0,0} & \dots & u_{0,n_2-2} \end{bmatrix} \quad u_{n_1-1} = \begin{bmatrix} u_{n_1-1,0} & u_{n_1-1,1} & \dots & u_{n_1-1,n_2-1} \\ u_{n_1-1,1} & u_{n_1-1,2} & \dots & u_{n_1-1,0} \\ \vdots & & & \\ u_{n_1-1,n_2-1} & u_{n_1-1,0} & \dots & u_{n_1-1,n_2-2} \end{bmatrix}$$

получаем преобразованную матрицу в более компактном виде

$$\begin{bmatrix} \overline{y_0} \\ \vdots \\ \overline{y_{n_1-1}} \end{bmatrix} = \begin{bmatrix} u_0 & \dots & u_{n_1-1} \\ \vdots & & \\ u_{n_1-1} & \dots & u_0 \end{bmatrix} \begin{bmatrix} \overline{x_0} \\ \vdots \\ \overline{x_{n_1-1}} \end{bmatrix}. \quad (7)$$

Для реализации гнездового алгоритма вычисления свертки представим выражение (7) в полиномиальной форме, воспользовавшись z -преобразованием. Тогда

$$\overline{y_{n_1-1}}z^{n_1-1} + \dots + \overline{y_0}z^0 = (\overline{u_{n_1-1}}z^{n_1-1} + \dots + \overline{u_0}z^0)(\overline{x_{n_1-1}}z^{n_1-1} + \dots + \overline{x_0}z^0) \bmod z^{n_2-1} - 1. \quad (8)$$

используя круговые полиномы $p_i(z)$ представим двучлен $z^{n_2-1} - 1$ в виде

$$z^{n_2-1} - 1 = \prod_{i=1}^S p_i(z). \quad (9)$$

Тогда на основе КТО, получаем (1.28)

$$\overline{y_j} = \sum_{i=1}^S M_i(z), \quad (10)$$

где

$$M_i(z) = \overline{y_{n_1-1}}z^{n_1-1} + \dots + \overline{y_0}z^0 \bmod p_i(z). \quad (11)$$

Известно [6, с. 98-100, 11, с. 59-60], что с помощью интерполяционной формулы Лагранжа может быть точно восстановлен полином степени $N-1$ по значениям N точках. В этом случае метод нахождения $Y(z) = U(z)X(z)$ заключается в определении констант $\langle Y(z_0), Y(z_1), \dots, Y(z_{N-1}) \rangle$, а затем в вычислении интерполяционной формулы

$$Y(z) = \sum_{i=0}^{N-1} Y(z_i) L_i(z), \quad (12)$$

в которой интерполяционные полиномы имеют вид

$$L_i(z) = \prod_{j \neq i} \frac{(z - z_j)}{(z_i - z_j)}. \quad (13)$$

В общем случае сложность гнездового алгоритма определяется следующим образом. Если M_1 и M_2 – количество умножений в алгоритмах вычисления n_1 и n_2 – точечной свертки соответственно, а A_1 и A_2 – количество сложений. Тогда, при НОД

$(n_1, n_2) = 1$, $n_1 n_2$ – точечная свертка вычисляется за $M = M_1 M_2$ операций умножений и $A = A_1 n_2 + M_1 A_2$ сложений.

Перестановка местами первой и второй координаты обработки приводит к изменению количества операций сложений, которые составят $A = A_2 n_1 + M_2 A_1$. Таким образом, при выборе гнездового алгоритма предпочтение отдается тому способу при котором число сложений меньше.

Если N разлагается в произведение более чем двух взаимно простых сомножителей $n_1, n_2, \dots, n_p$, то выражение для оценки временных затрат имеет вид

– число умножений

$$M = \prod_{j=1}^p M_j; \quad (14)$$

– число сложений

$$A = \sum_{j=1}^p A_j \prod_{l=j+1}^p n_l \prod_{i=0}^{j-1} M_j \quad (15)$$

при условии, что $M_0 = 1$.

В формуле (15) в каждом последующем слагаемом количество символов M_i увеличивается на единицу по сравнению с предыдущим, вследствие чего число операций сложений A гораздо сильнее зависит от числа операций умножения M_i , чем от числа сложений. Поэтому для больших сверток наиболее выгодны алгоритмы, вычисляющие короткие свертки с наименьшим числом умножений, даже если это достигается за счет увеличения числа операций сложения.

Применение гнездового алгоритма вычисления сверток позволяет осуществлять процедуры ортогональных преобразований сигналов, не используя тригонометрические функции. Отказ от быстрых алгоритмов ДПФ позволяет осуществлять переход от алгебраической системы, функционирующей в поле комплексных чисел в поле рациональных чисел $Q(R)$.

В работах [17, с. 22-25, 12, с. 76-78] приведена нижняя оценка эффективности различных сверточных алгоритмов. Известно, что вычисление линейной свертки двух последовательностей $x(n)$ и $u(n)$ длины N эквивалентно определению коэффициентов полинома

$$y(z) = x(z)u(z). \quad (16)$$

При этом $\text{ord } y(z) = \text{ord } x(z) + \text{ord } u(z) = 2N - 2$. Применение данного алгоритма позволяет найти значение $x(z_i)$ и $u(z_i)$ в точках z_i , $i = 0, 1, \dots, 2N - 2$, с последующим вычислением произведения

$$y(z_i) = x(z_i)u(z_i). \quad (17)$$

Искомый полином $y(z)$ полностью определяется значениями $y_i(z)$ заданных в $2N-1$ точках интерполяции z_i . Таким образом, если для восстановления используется быстрый интерполяционный полином по китайской теореме об остатках, то вычислительная сложность алгоритма линейной свертки определяется $2N-1$ умножениями вида (16).

В работах [13, с. 71-73, 18, с. 367-371, 19, с. 188-193] доказано, что минимальное число умножений, необходимых для вычисления циклической свертки длины N , равно $2N - k$, где k – число делителей N , включая 1 и N . Известно, что полином $z^N - 1$ разлагается на произведение круговых полиномов, число которых равно k . Если положить, что t_j – делитель N , то соответствующий этому делителю круговой полином C_{t_j} имеет степень $n_{t_j} = N/t_j$, а сумма степеней круговых полиномов равна N .

Применение КТО позволяет свести вычисление циклической свертки к выражению

$$y_j(z) = x_j(z)h_j(z) \bmod C_{t_j}(z), \quad j = 1, 2, \dots, k. \quad (18)$$

Известно, что для реализации равенства (1.36.18) требуется $2n_j - 1$ умножений. Проведя суммирование по всем j , получаем

$$\sum_{j=1}^k (2n_j - 1) = 2N - k. \quad (19)$$

Таким образом, очевидно, что для сверток малой длины граница мультипликативной сложности достигнута, и, следовательно, улучшение алгоритмов цифровой обработки сигналов возможно только за счет сокращения числа операций сложения [15, с. 23-24, 16, с. 22-23].

Однако, несмотря на то, что все алгоритмы с минимальной мультипликативной сложностью имеют структуру вида (5), где поточечное умножение векторов Bx и Au требует выполнения $2N-1$ итераций над операндами, практическая реализация их довольно сложна. Кроме того, изменение величины N входного вектора $x(n)$ влечет за собой изменение всей структуры вычислительного устройства.

Обращаясь к рассмотренным алгоритмам можно легко заметить, что наибольший выигрыш от их применения возможен только при реализации на основе классической фон-неймановской архитектуры построения вычислительного устройства. Таким образом, обобщая сказанное, можно сделать вывод о том, что рассмотренные выше ортогональные преобразования, реализованные в поле комплексных чисел, быстрые алгоритмы их вычисления, а так же гнездовые вычисления сверток не позволяют в полной мере реализовать все достоинства представляемые нейросетевым базисом.

Вместе с тем существуют математические модели ЦОС, обладающие свойством конечного кольца и поля, которые характеризуются простой реализацией, высоким уровнем параллелизма и возможностью построения с использованием нейросетевого базиса. Среди таких моделей особое место занимают теоретико-числовые и полиномиальные преобразования в полях Галуа.

Список литературы

1. Адошев А.И., Аникуев С.В., Гальвас А.В., Жданов В.Г., Ивашина А.В., Кобозев В.А., Логачева Е.А., Привалов Е.Е., Тимошенко Л.И., Шарипов И.К. Современные технологии в образовании // Развитие системы образования – обеспечение будущего. – Одесса, 2013. – С. 60-97.
2. Адошев А.И., Аникуев С.В., Тимошенко Л.И. и др. Развитие системы образования – обеспечение будущего. – Одесса, 2013. – Том 1. – Книга 2.
3. Аракелов О.Г., Тимошенко Л.И. Безопасность труда при работе с персональным компьютером // Культура и общество: история и современность: материалы III Всероссийской (с международным участием) научно-практической конференции / под ред. О.Ю. Колосовой, Т. В. Вергун,

- Р.Ф. Гударенко. – Ставрополь: АГРУС Ставропольского гос. аграрного ун-та. – 2014. – С. 114-117.
4. Земцев А.М., Тимошенко Л.И. Информационная составляющая безопасной эксплуатации электроустановок // Методы и средства повышения эффективности технологических процессов в АПК: Опыт, проблемы и перспективы. – Ставрополь, 2013. – С. 76-78.
5. Калмыков И.А., Тимошенко Л.И. Нейросетевые модели многоходовых сумматоров по модулю два // Фундаментальные исследования. – 2008. – № 3. – С. 73-74.
6. Калмыков И.А., Тимошенко Л.И. Систематическая матрица для цифровой фильтрации в модулярной арифметике // Современные наукоемкие технологии. – 2007. – № 11. – С. 98-100.
7. Калмыков И.А., Хайватов А.Б., Тимошенко Л.И., Гахов В.Р. Применение полиномиальной системы классов вычетов для повышения скорости функционирования спецпроцессора адаптивных средств защиты информации // Успехи современного естествознания. – 2007. – № 5. – С. 76.
8. Калмыков И.А., Зиновьев А.В., Тимошенко Л.И., Оленева Д.А. Математические модели цифровой обработки сигналов, используемые в современных информационных технологиях систем управления // Успехи современного естествознания. – 2009. – № 4. – С. 38-39.
9. Калмыков И.А., Емарлукова Я.В., Тимошенко Л.И., Гахов В.Р. Обобщенное дискретное преобразование Фурье для колец неприводимых полиномов // Успехи современного естествознания. – 2007. – № 5. – С. 77.
10. Калмыков И.А., Петлеваний С.В., Тимошенко Л.И., Лисицын А.В. Разработка преобразователя модулярного кода ПСКВ в позиционный код // Современные наукоемкие технологии. – 2006. – № 4. – С. 57-59.
11. Калмыков И.А., Тимошенко Л.И., Чипига А.А. Разработка преобразователя позиционного кода в полиномиальную систему классов вычетов // Современные наукоемкие технологии. – 2006. – № 4. – С. 59-60.
12. Кузьменко И.П., Тимошенко Л.И. Систематические принципы организации вычислений в спецпроцессоре цифровой обработки сигналов с параллельно-конвейерным распределением вычислительного процесса // Культура и общество: история и современность: материалы II Всероссийской (с международным участием) научно-практической конференции. – Ставрополь. – 2013. – С. 76-78.
13. Тимошенко Л.И. Нейросетевая реализация вычислений в полиномиальной системе классов вычетов // Фундаментальные исследования. – 2008. – № 3. – С. 71-73.
14. Тимошенко Л.И. Информатика. Учебное пособие. – Ставрополь: Издательство АГРУС. – 2014. – Т. 2.
15. Тимошенко Л.И. Анализ основных методов прямого преобразования из позиционной системы счисления в модулярный полиномиальный код // Современные наукоемкие технологии. – 2007. – № 9. – С. 23-24.
16. Тимошенко Л.И. Применение математической модели обладающей свойством кольца, для реализации цифровой обработки сигналов // Современные наукоемкие технологии. – 2007. – № 9. – С. 22-23.
17. Тимошенко Л.И. Реализация модульных операций в кольце полиномов с помощью нейронных сетей // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 1-1. – С. 22-25.
18. Тимошенко Л.И. Разработка нейросетевых реализаций базовых операций обобщенного дискретного преобразования Фурье в кольце полиномов // Международный журнал экспериментального образования. – 2015. – № 2-3. – С. 367-371.
19. Тимошенко Л.И. Дискретное преобразование Фурье и его быстрые алгоритмы // Современные наукоемкие технологии. – 2014. – № 12-2. – с. 188-193.