

УДК 004.031.6

Н.А. Петровский, А.В. Станкевич, А.А. Петровский

**CORDIC-ТЕХНИКА ДЛЯ ФИКСИРОВАННОГО УГЛА ВРАЩЕНИЯ
В ОПЕРАЦИИ УМНОЖЕНИЯ КВАТЕРНИОНОВ**

Предлагается ряд решений ключевого элемента параунитарного банка фильтров на основе алгебры кватернионов – умножителя кватернионов с использованием CORDIC (Coordinate Rotation Digital Computer)-техники для фиксированного угла вращения, в которых в отличие от известных решений параметры управления 4D-вращением представляются нелинейной функцией числа сдвигов входных операндов операции микровращения. Предложенный подход проектирования умножителя на кватернион-константу позволяет достигать максимальной производительности схемы умножителя при скромном использовании ресурсов, например FPGA.

Введение

Алгебра кватернионов \mathbb{H} является ассоциативной некоммутативной четырехмерной алгеброй [1] $\mathbb{H} = \{ \mathbf{q} = q_1 + q_2i + q_3j + q_4k \mid q_1, q_2, q_3, q_4 \in \mathbb{R} \}$, где ортогональные мнимые части подчиняются следующим законам умножения: $i^2 = j^2 = k^2 = ijk = -1$, $ij = -ji = k$, $jk = -kj = i$, $ki = -ik = j$. Операции сложения и умножения кватернионов могут быть реализованы с использованием векторно-матричной алгебры. Для этого кватернионы представляются в виде четырехмерных векторов. Умножение кватернионов в векторной форме производится по правилу умножения вектора на матрицу. В связи с тем что операция умножения некоммутативна, продукт умножения определяется операторами умножения «левый» $\mathbf{M}^+(q)$ и «правый» $\mathbf{M}^-(q)$ [2]:

$$qx \Leftrightarrow \underbrace{\begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix}}_{\mathbf{M}^+(q)} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}}, \quad xq \Leftrightarrow \underbrace{\begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & q_4 & -q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ q_4 & q_3 & -q_2 & q_1 \end{bmatrix}}_{\mathbf{M}^-(q)} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}}. \quad (1)$$

Обе матрицы (1) ортогональны до масштабного коэффициента, заданного нормой кватернионов $|q| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}$: $\mathbf{M}^\pm(q)^{-1} = (1/|q|)\mathbf{M}^\pm(q)^T$, где T – знак транспонирования матрицы. Как следует из структур матриц, их свойства не зависят от значений компонент кватерниона и, следовательно, не чувствительны к их модификации, например квантованию. Матрицы $\mathbf{M}^+(q)$ и $\mathbf{M}^-(q)$ соотносятся между собой как

$$\mathbf{M}^-(q) = \mathbf{D}_C \mathbf{M}^+(q)^T \mathbf{D}_C, \quad (2)$$

где $\mathbf{D}_C = \text{diag}(1, -\mathbf{I}_3)$ описывает в матричной нотации оператор гиперкомплексного сопряжения. Определив сопряженный кватернион $\bar{q} = q_1 - q_2i - q_3j - q_4k$ в векторно-матричном виде как $\bar{\mathbf{q}} = \mathbf{D}_C \mathbf{q}$, можно получить

$$\mathbf{M}^\pm(\bar{q}) = \mathbf{M}^\pm(q)^T, \quad (3)$$

тогда выражение (2) переопределяется следующим образом:

$$\mathbf{M}^-(\bar{q}) = \mathbf{D}_C \mathbf{M}^+(q) \mathbf{D}_C. \quad (4)$$

Последнее равенство показывает, что результаты, полученные для умножения «левого» qx или «правого» xq , могут быть применимы к умножению на сопряженный кватернион. В частности, это доказывает, что вычислительная сложность операторов умножений кватернионов одинакова. Данное обстоятельство позволяет сосредоточить внимание на организации вычисления продукта qx оператора «левого» умножения $\mathbf{M}^+(q)$.

Последние 20 лет в цифровой обработке сигналов алгебра кватернионов рассматривается как новая парадигма, позволяющая осуществлять обработку сигналов непосредственно в многомерном домене. Кватернионы уже с успехом использовались в цифровой обработке сигналов, например адаптивных фильтрах Калмана [3], при оценке спектра по методу MUSIC [4], в сингулярной декомпозиции матриц [5], параунитарных банках фильтров (ПУБФ) – Q-ПУБФ [6–8] и других приложениях [9, 10]. ПУБФ могут рассматриваться как наиболее значимое преобразование среди многоскоростных систем цифровой обработки сигналов [11] для систем мультимедиа. Это обусловлено тем фактом, что подобные банки являются преобразованиями без потерь в дополнение к гарантированной перфективной реконструкции сигнала. Точное соотношение между энергиями во всем частотном диапазоне и в субполосах сильно упрощает теоретические выкладки и делает ПУБФ полезными в применении к компрессии (кодированию) изображений. В работе [8] утверждается, что для каждой ортогональной матрицы размерностью 4×4 существует уникальная (с точностью до знака) пара единичных кватернионов P и Q ($|P|=1$, $|Q|=1$), таких, что $\mathbf{M}^+(P) \cdot \mathbf{M}^-(Q) = \mathbf{M}^-(Q) \cdot \mathbf{M}^+(P)$. Факторизации полифазных передаточных матриц анализа $\mathbf{E}(z)$ и синтеза $\mathbf{D}(z)$ решетчатой структуры Q-ПУБФ в алгебре кватернионов будут соответственно равны [8]:

$$\mathbf{E}(z) = \left(\prod_{i=N-1}^1 \mathbf{M}^+(Q_i) \Lambda(z) \right) \mathbf{M}^+(Q_0) \mathbf{M}^-(P_0); \quad (5)$$

$$\mathbf{D}(z) = \mathbf{M}^-(\bar{P}_0) \mathbf{M}^+(\bar{Q}_0) \left(\prod_{i=1}^{N-1} \hat{\Lambda}(z) \mathbf{M}^+(\bar{Q}_i) \right), \quad (6)$$

где \bar{P}_0 и \bar{Q}_0 – сопряженные кватернионы; $\Lambda(z) = \text{diag}(z^{-1}, \mathbf{I}_{M-1})$ и $\hat{\Lambda}(z) = \text{diag}(1, z^{-1} \mathbf{I}_{M-1})$ – матрицы задержек; \mathbf{I}_{M-1} – $M \times M$ единичная матрица. Данная факторизация всегда соответствует ортогональному преобразованию сигналов, даже если ее компоненты квантованы. Это происходит потому, что столбцы каждой матрицы операторов умножения кватернионов (1) составлены из одних и тех же элементов с точностью до знака. Таким образом, множитель кватернионов является альтернативным модулем построения Q-ПУБФ и может рассматриваться как обобщение планарных вращений Гивенса на 4D-пространство [7, 8]. Данные решетчатые структуры Q-ПУБФ на кватернионах являются системами без потерь, несмотря на квантование коэффициентов. Более того, предложенный подход дает возможность непосредственного выражения условия регулярности первого порядка, которое зафиксировано в значениях коэффициентов решетчатой структуры Q-ПУБФ и легко удовлетворяется даже для арифметики с фиксированной запятой [7, 8].

Следовательно, базовым элементарным преобразованием Q-ПУБФ является умножение кватернионов, где один из сомножителей – кватернион-константа, т. е. кватернион с постоянными параметрами. При этом умножение кватернионов – ключевая операция, от эффективной реализации которой зависят характеристики всего преобразования. Однако прямое умножение матрицы на вектор потребует 16 умножений действительных чисел и 12 алгебраических сложений. В [12] показано, как можно вычислить продукт умножения кватернионов на основе восьми действительных умножений, но данная техника не подходит для случая, когда один операнд есть кватернион-константа. Алгоритм выполнения умножения кватернионов на основе лестничной структуры в арифметике с фиксированной запятой представлен в работе [13] (12 операций умножения на действительные числа), где основной компонент схемы – множитель-накопитель. Реализация множителя в этом случае требует больших затрат компонент структуры ПЛИС типа FPGA. Представляют интерес вычислительные схемы множителя кватернионов без использования устройств действительных умножений. Известно решение реали-

зации умножителя кватернионов на распределенной арифметике [14], которое позволяет получать сбалансированные схемные решения как по производительности, так и по потребляемой мощности. Однако для достижения высокой скорости и точности вычислений здесь необходим большой объем памяти.

В настоящее время в области систем мультимедиа достигнут большой прогресс в развитии алгоритмов и архитектурных решений для высокопроизводительных вычислений с малыми аппаратными затратами – CORDIC-алгоритмов [15]. Преимущество CORDIC заключается в том, что с помощью простых операций сдвига и сложения $a \pm b2^{-i}$ можно вычислять ряд задач во многих областях, таких как обработка сигналов и изображений, системы телекоммуникаций, робототехника, 3D-графика. Параллельные и конвейерные структуры CORDIC-процессоров позволяют достигать высокой скорости вычислений [15].

Целью настоящей статьи является представление ряда решений ключевого элемента Q-ПУБФ – умножителя кватернионов на базе CORDIC-алгоритмов.

1. Базовая CORDIC-техника

1.1. 2-D-CORDIC-алгоритм

Вычислительный CORDIC-алгоритм, предложенный Волдером [16] и позднее развитый Вальтером [17], представляет собой алгоритм, оперирующий 2D-векторами для вычисления вращений с помощью простых арифметических примитивов: сдвига и сложения. Поворот вектора $\mathbf{x} = [x_1 \ x_2]^T$ на угол φ может быть описан произведением данного вектора на соответствующую матрицу вращения $\mathbf{R}_2(\varphi)$:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}}_{\mathbf{R}_2(\varphi)} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{x}}. \quad (7)$$

Вместе с тем выражение (7) описывает произведение комплексного числа $(x_1 + jx_2)$ на комплексную экспоненту $e^{-j\varphi}$. CORDIC-алгоритм аппроксимирует матрицу вращения $\mathbf{R}_2(\varphi)$ произведением N элементарных вращений $\mathbf{R}_2(i)$:

$$\mathbf{R}_2(\varphi) = \prod_{i=0}^{N-1} \mathbf{R}_2(i); \quad (8)$$

$$\mathbf{R}_2(i) = (1 + \delta t_i^2)^{-1/2} \mathbf{U}_2(i), \quad \mathbf{U}_2(i) = \begin{bmatrix} 1 & -\delta \cdot \sigma(i) t_i \\ \delta \cdot \sigma(i) t_i & 1 \end{bmatrix}, \quad (9)$$

где $\mathbf{U}_2(i)$ – ненормализованная часть $\mathbf{R}_2(i)$; $\delta = 1$, если вращение осуществляется в евклидовом пространстве; $\delta = -1$, если вращение осуществляется в псевдоевклидовом пространстве (гиперболическое вращение); $t_i = 2^{-i}$. Параметры N , $\sigma(i)$, i подбираются таким образом, что результат произведения (8) аппроксимирует вращение на угол

$$\varphi \approx \sum_{i=0}^{N-1} \sigma(i) \cdot 2^{-i}, \quad (10)$$

где N – число вращений, а параметр $\sigma(i) \in \{-1, 1\}$ определяет направление вращения для оставшейся части угла. CORDIC-алгоритм состоит из итерационного процесса вращения и процесса масштабирования. На каждой итерации входной вектор \mathbf{x} поворачивается на угол 2^{-i} , но результат вращения не соответствует фиксированному радиусу: длина вектора увеличивается на величину $(1 + 2^{-2i})^{1/2}$, т. е. необходима нормализация размера вектора, которая выполняется в

процессе масштабирования. Таким образом, на i -й итерации для соответствующего входного вектора $\mathbf{x}(i) = [x_1(i) \quad x_2(i)]^T$ выходной вектор $\mathbf{x}(i+1) = [x_1(i+1) \quad x_2(i+1)]^T$ вычисляется следующим образом:

$$\mathbf{x}(i+1) = \mathbf{U}_2(i) \cdot \mathbf{x}(i); \quad (11)$$

$$\varphi(i+1) = \varphi(i) - \sigma(i)2^{-i} \quad \text{для } i = \overline{0, N-1}, \quad (12)$$

где $\mathbf{U}_2(i)$ – оператор микроповорота на i -й итерации CORDIC-алгоритма; $\varphi(i)$ – оставшийся угол после $(i-1)$ -й итерации. Амплитуда результата нормализуется на этапе масштабирования после окончания итерационного этапа: выходной вектор $\mathbf{x}(N)$ делится на масштабный фактор $T = \prod_{i=0}^{N-1} (1 + 2^{-2i})^{1/2}$. Итак, окончательный результат $\mathbf{x}_{out}(N)$ CORDIC-алгоритма может быть представлен как

$$\mathbf{x}_{out}(N) = \frac{1}{T} \mathbf{x}(N) = \left(\prod_{i=0}^{N-1} (1 + 2^{-2i})^{1/2} \right)^{-1} \mathbf{x}(N). \quad (13)$$

Таким образом, масштабирование требует дополнительной вычислительной операции, но если масштабный фактор представить в виде

$$\frac{1}{T} = \sum_{s=0}^{S-1} \sigma(s)2^{-i(s)}, \quad (14)$$

где $\sigma(s) = \pm 1$ и $0 \leq i(s) \leq N-1$, то процесс масштабирования может быть описан рекурсивным выражением:

пусть $\mathbf{x}(0) = 0$ и $\mathbf{x}(S) = \mathbf{x}_{out}(N)$, тогда

$$\mathbf{x}(s+1) = \mathbf{x}(s) + \sigma(s)2^{-i(s)} \mathbf{x}(N) \quad \text{для } s = 0, 1, \dots, S-1. \quad (15)$$

В стандартной CORDIC-технике параметры алгоритма следующие: $\sigma(i) \in \{-1, 1\}$, $i = \overline{0, N-1}$, и $N = B$, где B – число разрядов в двоичном представлении компонент векторов, т. е. сдвиг и знак оператора направления вращения для обоих процессов (итерационного и процесса масштабирования) определяются в онлайн-режиме на основе входных данных. Выбор соответствующих параметров алгоритма определяет фиксированное число микроповоротов (итераций).

1.2. Многомерный CORDIC-алгоритм

В работе [18] были предложены быстрые 3D- и 4D-CORDIC-алгоритмы, число итераций которых немного больше, чем для одного 2D-CORDIC-поворота. Данные алгоритмы основываются на фундаментальной теореме Кэли [19]. Формула Кэли устанавливает связь между ортогональными и кососимметрическими операторами в евклидовом пространстве и позволяет любое n D евклидово вращение \mathbf{R}_n , для которого характеристическое число равно -1 , представить как

$$\mathbf{R}_n = (\mathbf{I}_n - \mathbf{T}_n)(\mathbf{I}_n + \mathbf{T}_n)^{-1}, \quad (16)$$

где \mathbf{I}_n – единичная матрица $n \times n$; \mathbf{T}_n – кососимметрическая матрица (оператор), т. е. $\mathbf{T}_n^T = -\mathbf{T}_n$. Например, для i -го элементарного 3D-поворота в [18] предлагается матрицу $\mathbf{T}_{3,i}$ конструировать следующим образом:

$$\mathbf{T}_3(i) = \begin{bmatrix} 0 & -\sigma_1(i)t & \sigma_2(i)t \\ \sigma_1(i)t & 0 & -\sigma_1(i)t \\ -\sigma_2(i)t & \sigma_1(i)t & 0 \end{bmatrix}, \quad \sigma_1(i), \sigma_2(i) \in \{1, -1\}, \quad (17)$$

где $t = 2^{-i}$. Соответствующая элементарная нормализованная матрица вращения согласно (16) определится как

$$\mathbf{R}_3(i) = (\mathbf{I}_3 - \mathbf{T}_{3,i})(\mathbf{I}_3 + \mathbf{T}_{3,i})^{-1} = \frac{1}{\sqrt{1+3 \cdot t^2}} \begin{bmatrix} 1-t^2 & 2(\sigma_1(i)t+t^2) & -2(\sigma_2(i)t+t^2) \\ -2(\sigma_1(i)t+t^2) & 1-t^2 & 2(\sigma_1(i)t+t^2) \\ 2(\sigma_2(i)t+t^2) & -2(\sigma_1(i)t+t^2) & 1-t^2 \end{bmatrix}. \quad (18)$$

Данные вращения выполняются относительно оси $[\sigma_1(i) \ \sigma_2(i) \ \sigma_1(i)]^T$, что соответствует направляющему вектору либо $[1 \ 1 \ 1]^T$, либо $[1 \ -1 \ 1]^T$. Когда вектор $[x_1 \ x_2 \ x_3]^T$ следует привести к направлению $[1 \ 0 \ 0]^T$, т. е. первой координатной оси, знаки направления вращения на i -й итерации $\sigma_1(i)$ и $\sigma_2(i)$ выбираются согласно следующему закону управления: $\sigma_1(i) = \text{sign}(x_{1,i-1} \cdot x_{2,i-1})$, $\sigma_2(i) = -\text{sign}(x_{1,i-1} \cdot x_{2,i-1})$, где $x_{j,i-1}$ определяет j -ю компоненту вектора в начале i -й итерации. Данный метод не является систематическим подходом конструирования nD CORDIC элементарных матриц вращения \mathbf{R}_n для произвольных n . Согласно [18] это его главный недостаток.

CORDIC-алгоритм может быть расширен на многомерный случай с использованием преобразования Хаусхолдера [18], матрица отражения которого определяется следующим образом:

$$\mathbf{H}_n = \mathbf{I}_n - 2 \frac{\mathbf{u} \cdot \mathbf{u}^T}{\mathbf{u}^T \cdot \mathbf{u}}, \quad (19)$$

где \mathbf{u} – n -мерный вектор. Результат преобразования $(\mathbf{H}_n \cdot \mathbf{v})$ есть отражение n -мерного вектора \mathbf{v} по отношению к гиперплоскости с нормалью \mathbf{u} , которая проходит через начало координат. Таким образом, CORDIC-алгоритм на основе отражения Хаусхолдера выполняет операцию векторизации n -мерного вектора к одной из осей. Ненормализованная матрица вращения для n -мерного вектора соответствующей i -й итерации $\mathbf{R}_{Hn}(i)$ определяется как произведение двух простых отражений Хаусхолдера [5]:

$$\mathbf{R}_{Hn}(i) = \left(\mathbf{I}_n - 2 \frac{\mathbf{e}_1 \mathbf{e}_1^T}{\mathbf{e}_1^T \mathbf{e}_1} \right) \left(\mathbf{I}_n - 2 \frac{\mathbf{u}_i \mathbf{u}_i^T}{\mathbf{u}_i^T \mathbf{u}_i} \right), \quad (20)$$

где $\mathbf{e}_1 = [1 \ 0 \ \dots \ 0]^T$ и $\mathbf{u}_i = [1 \ \sigma_1(i) \cdot t_i \ \dots \ \sigma_{n-1}(i) \cdot t_i]^T$, $t_i = \tan \varphi_i = 2^{-i}$, а управляющие сигналы направления вращения выбираются как

$$\sigma_j(i) = \text{sign}(x_{1,i-1} \cdot x_{j+1,i-1}), \quad 1 \leq j \leq n-1. \quad (21)$$

Для $n=4$ нормализованная матрица вращения $\mathbf{R}_{H4}(i)$ согласно (20) задается как функция сдвигов и управляющих сигналов направления вращения [5]:

$$\mathbf{R}_{H4}(i) = \frac{1}{\sqrt{1+3 \cdot t_i^2}} \begin{bmatrix} 1 & -\sigma_1(i)t_i & -\sigma_2(i)t_i & -\sigma_3(i)t_i \\ \sigma_1(i)t_i & 1 & \sigma_3(i)t_i & -\sigma_2(i)t_i \\ \sigma_2(i)t_i & -\sigma_3(i)t_i & 1 & \sigma_1(i)t_i \\ \sigma_3(i)t_i & \sigma_2(i)t_i & -\sigma_1(i)t_i & 1 \end{bmatrix}. \quad (22)$$

Следовательно, CORDIC-алгоритм Хаусхолдера представляет собой обобщение CORDIC-алгоритма на nD -мерный случай для осуществления вращения в евклидовом и псевдо-евклидовом пространствах. Следует также отметить, что nD CORDIC-алгоритм хорошо отображается на параллельные структуры процессоров [18].

В [20] описан 4D-CORDIC-алгоритм умножения кватернионов для сингулярного разложения матриц на основе преобразования Хаусхолдера. Авторы показывают, что матрицы умножения (1) можно записать в виде факторизации:

$$\mathbf{M}^\pm(q) \approx \prod_{n=0}^{N-1} \frac{1}{|\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))|} \mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))), \quad (23)$$

где кватернион \mathbf{q} на i -й итерации 4D-CORDIC-алгоритма аппроксимируется как $\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i)) = 1 + \sigma_1(i)2^{\tau(i)}i + \sigma_2(i)2^{\tau(i)}j + \sigma_3(i)2^{\tau(i)}k$, а его матрица умножения, которая описывает i -ю итерацию 4D-CORDIC-алгоритма, представляется следующим образом:

$$\mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))) \approx \begin{bmatrix} 1 & -\sigma_1(i)2^{\tau(i)} & -\sigma_2(i)2^{\tau(i)} & -\sigma_3(i)2^{\tau(i)} \\ \sigma_1(i)2^{\tau(i)} & 1 & \mp\sigma_3(i)2^{\tau(i)} & \pm\sigma_2(i)2^{\tau(i)} \\ \sigma_2(i)2^{\tau(i)} & \pm\sigma_3(i)2^{\tau(i)} & 1 & \mp\sigma_1(i)2^{\tau(i)} \\ \sigma_3(i)2^{\tau(i)} & \mp\sigma_2(i)2^{\tau(i)} & \pm\sigma_1(i)2^{\tau(i)} & 1 \end{bmatrix}. \quad (24)$$

Параметр количества двоичных сдвигов $\tau(i)$ зависит от индекса итераций i , т. е. $\tau(i) = -i$, а параметры управления направлением вращения $\sigma_1(i), \sigma_2(i), \sigma_3(i)$ принимают значения 1 или -1 .

Таким образом, 4D-CORDIC-алгоритм умножения кватернионов с матрицей умножения (24), предложенный в [20], определяется как

$$\mathbf{x}_{i+1} = \mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))) \cdot \mathbf{x}_i \quad (i = \overline{0, N-1}; \mathbf{x}_0 = \mathbf{x}). \quad (25)$$

Для того чтобы 4D-вектор \mathbf{X} привести к направлению первой канонической оси, сигналы управления направлением вращения на i -й итерации определяются кватернионом \mathbf{X}_i :

$$\sigma_1(i) = -\text{sign}(x_{1,i}) \cdot \text{sign}(x_{2,i}), \quad \sigma_2(i) = -\text{sign}(x_{1,i}) \cdot \text{sign}(x_{3,i}), \quad \sigma_3(i) = -\text{sign}(x_{1,i}) \cdot \text{sign}(x_{4,i}). \quad (26)$$

Норма кватерниона \mathbf{X}_{i+1} , или длина вектора на i -й итерации, согласно (22) равна $|\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))| = \sqrt{1 + 3 \cdot 2^{-2\tau(i)}}$. Следовательно, выходной результат итерационного процесса \mathbf{X}_N нормализуется на коэффициент $1/K = \prod_{i=0}^{N-1} 1/\sqrt{1 + 3 \cdot 2^{-2\tau(i)}}$, который может быть

аппроксимирован в соответствии с (14) и тогда умножение на данный коэффициент реализуется отдельно от итерационного процесса как серия сдвигов и сложений (процесс масштабирования). Окончательный результат 4D-CORDIC-алгоритма равен $\mathbf{x}_{out,N} = (1/K)\mathbf{x}_N$. Анализ матрицы (24) показывает, что при увеличении числа итераций i матрица $\mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i)))$ сходится к единичной матрице 4×4 . Схема вычисления итерации (микровращений) 4D-CORDIC-алгоритма может быть реализована на четырех регистрах сдвига и на четырех четырехходовых сумматорах. Это обуславливает необходимость дополнительной логики контроля переполнения, а при реализации на ПЛИС типа FPGA такие сумматоры требуют повышенного аппаратного ресурса. Схема микровращений 4D-CORDIC-алгоритма не может быть использована на этапе масштабирования результата итерационного процесса, 4D-итерации (25), (26) не всегда сходятся [20]. Как показали экспериментальные исследования, число кватернионов на множестве 10^4 случайно сгенерированных единичных кватернионов, для которых заданная точность недостижима (алгоритм не сходится), изменялось от 6 до 18 % для точности CORDIC-процесса соответственно 10^{-2} и 10^{-6} .

2. 4D-CORDIC-алгоритм умножения на кватернион-константу

2.1. Постановка задачи

Имеется множество алгоритмов, например таких, как дискретное косинусное преобразование или быстрое преобразование Фурье, где параметры преобразования известны априори. Для большинства алгоритмов на основе алгебры кватернионов гиперкомплексные переменные умножаются на постоянные коэффициенты, например Q-ПУБФ [7, 8]. Как было показано выше, в решетчатой структуре Q-ПУБФ (5) и (6) в операции умножения кватернионов (1) операнд x является входной переменной, а коэффициент q – кватернион-константой, компоненты которого определяют характеристики Q-ПУБФ и задают матрицу умножения. Оба операнда нормированы (норма равна единице), и их матрицы умножения различаются от заданных только масштабным коэффициентом. Для данных кватернионов операция умножения на сопряженный кватернион \bar{q} эквивалентна умножению на обратный кватернион:

$$q \cdot \bar{q} = q \cdot q^{-1} = 1 + (0i + 0j + 0k) \Leftrightarrow \mathbf{M}^{\pm}(q) \underbrace{\begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \end{bmatrix}^T}_{\bar{\mathbf{q}}^T} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T}_{\mathbf{e}_1^T}. \quad (27)$$

Матричное выражение (27) может интерпретироваться с различных точек зрения: результат умножения на \bar{q} есть кватернион, у которого мнимые части обнулены, что эквивалентно расположению вектора $\bar{\mathbf{q}}$ вдоль первой канонической оси при умножении на $\mathbf{M}^{\pm}(q)$. Таким образом, единичные кватернионы, у которых матрицы умножения ортонормальны, могут быть использованы для вывода CORDIC-алгоритма умножения кватернионов на основе факторизаций $\mathbf{M}^{\pm}(q)$. Факторизации $\mathbf{M}^{\pm}(q)$ имеют эффект, аналогичный умножению на вектор $\bar{\mathbf{q}}$, при следующих ограничениях:

– все сомножители в факторизации должны иметь структуру, аналогичную матрице $\mathbf{M}^{\pm}(q)$, т. е. произведение любого их множества представляет матрицу умножения кватернионов соответствующего типа:

$$\begin{aligned} \mathbf{M}^+(q_{k-1} \dots q_0) &= \mathbf{M}^+(q_{k-1}) \cdot \dots \cdot \mathbf{M}^+(q_0); \\ \mathbf{M}^-(q_0 \dots q_{k-1}) &= \mathbf{M}^-(q_{k-1}) \cdot \dots \cdot \mathbf{M}^-(q_0); \end{aligned} \quad (28)$$

– сомножители в факторизации (28) должны сходиться к единичной матрице 4×4 , \mathbf{I}_4 , т. е. возможная бесконечная серия сомножителей в факторизации может быть усечена до некоторой аппроксимации исходной матрицы при допустимой ошибке;

– сомножители в факторизации должны быть «простыми» матрицами, умножение на которые можно реализовать с использованием нескольких операций сдвига и сложения.

2.2. 4D-CORDIC-алгоритм умножения кватернионов (4D-CONST-Q-CORDIC) на базе 2D-вращений в 4D-гиперплоскости

Для уменьшения вычислительных ресурсов необходим поиск альтернативных Q-CORDIC-алгоритмов, которые просто проектируются. Предлагается конструировать 4D-CORDIC-алгоритм умножения кватернионов на базе 2D-вращений в 4D-гиперплоскости [21, 22], основываясь на факторизации (28) матрицы умножения кватернионов $\mathbf{M}^{\pm}(q)$, когда единичный кватернион q представляется как произведение $q = d_k d_j d_i$ трех выродившихся единичных кватернионов d_i, d_j, d_k , у которых соответственно одна мнимая часть i, j, k :

$$\begin{aligned} \mathbf{M}^+(q) &= \mathbf{M}^+(d_k) \cdot \mathbf{M}^+(d_j) \cdot \mathbf{M}^+(d_i); \\ \mathbf{M}^-(q) &= \mathbf{M}^-(d_i) \cdot \mathbf{M}^-(d_j) \cdot \mathbf{M}^-(d_k). \end{aligned} \quad (29)$$

Матрицы умножения вырожденных кватернионов $\mathbf{M}^\pm(d_i)$, $\mathbf{M}^\pm(d_j)$ и $\mathbf{M}^\pm(d_k)$ разреженные, каждая из них описывает 2D-вращение в различных плоскостях 4D-гиперплоскости на одинаковый угол, но, возможно, в противоположных направлениях:

$$\begin{aligned}\mathbf{M}^\pm(d_i) &= \text{diag}(\mathbf{R}_2(\varphi_i), \mathbf{R}_2(\pm\varphi_i)); \\ \mathbf{M}^\pm(d_j) &= \mathbf{P}_{1324} \text{diag}(\mathbf{R}_2(\varphi_j), \mathbf{R}_2(\mp\varphi_j))\mathbf{P}_{1324}^T; \\ \mathbf{M}^\pm(d_k) &= \mathbf{P}_{1423} \text{diag}(\mathbf{R}_2(\varphi_k), \mathbf{R}_2(\pm\varphi_k))\mathbf{P}_{1423}^T,\end{aligned}\quad (30)$$

где $\mathbf{R}_2(\varphi)$ – матрица вращения Гивенса (7); \mathbf{P}_{1324} и \mathbf{P}_{1423} – матрицы перестановок строк и столбцов в матрице умножения кватернионов в соответствии с индексами. Умножение на каждую разреженную матрицу умножения вырожденного кватерниона равносильно выполнению вращения на одинаковые углы, которое может осуществляться параллельно на двух 2D-CORDIC-блоках для фиксированного угла вращения φ . Поэтому итерации плотно связанных 2D-CORDIC-алгоритмов идентифицируются с одной итерацией 4D-CORDIC-алгоритма. Изменение плоскостей 2D-вращений или переназначение входов и выходов 2D-CORDIC-блоков эквивалентно переключению 4D-CORDIC-алгоритмов среди трех итерационных процессов, описываемых следующими разреженными матрицами умножения вырожденных кватернионов:

$$\mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), 0, 0)) \approx \begin{bmatrix} 1 & -\sigma_1(i)2^{-\tau(i)} & 0 & 0 \\ \sigma_1(i)2^{-\tau(i)} & 1 & 0 & 0 \\ 0 & 0 & 1 & \pm\sigma_1(i)2^{-\tau(i)} \\ 0 & 0 & \pm\sigma_1(i)2^{-\tau(i)} & 1 \end{bmatrix}; \quad (31)$$

$$\mathbf{M}^\pm(\zeta(\tau(i), 0, \sigma_2(i), 0)) \approx \begin{bmatrix} 1 & 0 & -\sigma_2(i)2^{-\tau(i)} & 0 \\ 0 & 1 & 0 & \pm\sigma_2(i)2^{-\tau(i)} \\ \sigma_2(i)2^{-\tau(i)} & 0 & 1 & 0 \\ 0 & \pm\sigma_2(i)2^{-\tau(i)} & 0 & 1 \end{bmatrix}; \quad (32)$$

$$\mathbf{M}^\pm(\zeta(\tau(i), 0, 0, \sigma_3(i))) \approx \begin{bmatrix} 1 & 0 & 0 & -\sigma_3(i)2^{-\tau(i)} \\ 0 & 1 & \pm\sigma_3(i)2^{-\tau(i)} & 0 \\ 0 & \pm\sigma_3(i)2^{-\tau(i)} & 1 & 0 \\ \sigma_3(i)2^{-\tau(i)} & 0 & 0 & 1 \end{bmatrix}. \quad (33)$$

Переходя от размерности пространства 4D к 2D и обратно, можно аппроксимировать умножение кватернионов дискретными вырожденными кватернионами, у которых действительные части равны единице, а ненулевые мнимые части представляются отрицательной степенью по основанию два.

В факторизациях (29), (30) выбор углов $\varphi_i, \varphi_j, \varphi_k$ определяет соответствующую тройку вырожденных единичных кватернионов d_i, d_j, d_k , умножение которых на сопряженный кватернион \bar{q} обнуляет мнимые части результирующего произведения, т. е. $d_i \cdot d_j \cdot d_k \cdot \bar{q} = q \cdot \bar{q} = 1 + 0i + 0j + 0k$. Обнуление мнимых частей произведения на основе CORDIC-алгоритма предполагает решение задачи аппроксимации данного процесса с помощью дискретных вырожденных кватернионов (31), (33), каждый из которых выбирается следующим образом:

$$\begin{aligned}d_i &= \cos(\varphi_i) + i \sin(\varphi_i), \quad \varphi_i = \arctg(q_2/q_1) \rightarrow d_i \bar{q} = 0i + \dots; \\ d_j &= \cos(\varphi_j) + j \sin(\varphi_j), \quad \varphi_j = \arctg(q_3/q_1) \rightarrow d_j \bar{q} = \dots + 0j + \dots; \\ d_k &= \cos(\varphi_k) + j \sin(\varphi_k), \quad \varphi_k = \arctg(q_4/q_1) \rightarrow d_k \bar{q} = \dots + 0k.\end{aligned}$$

После умножения на сопряженный кватернион мнимые части могут отличаться от нуля, тогда данный процесс повторяется для другой тройки дискретных вырожденных кватернионов и так до тех пор, пока мнимые части не будут обнулены или по крайней мере их значения не будут меньше некоторой заданной величины. Это объясняется тем, что преобразование ортогональное и не модифицирует норму входных кватернионов. Если одна из мнимых частей увеличивается, то другая мнимая часть должна уменьшаться, так как сумма квадратов мнимых частей единичного кватерниона равна единице минус квадрат действительной части. Таким образом, последовательность умножений на дискретные вырожденные кватернионы путем увеличения реальной части неявно уменьшает значения мнимых частей, даже если их максимальная амплитуда временно увеличивается. Обнуление мнимых частей может осуществляться последовательно, начиная с мнимой части по i , однако фактически необходимо на каждой итерации выбирать для обнуления мнимую часть, у которой наибольшая величина. Хотя оба подхода сходятся, но у второго сходимость процесса выше [21]. Максимум точности операции умножения на кватернион-константу достигается за некоторое число итераций на полученном наборе управляющих параметров $\tau(i)$ и $\sigma(i)$. Значения мнимых компонент по модулю вектора $\mathbf{x}(i)$ операции CORDIC на сопряженный кватернион-константу \bar{q} близки к нулю. Псевдокод программы алгоритма расчета операторов сдвига $\tau(i)$ и направления вращения $\sigma(i)$ для 4D CONST Q-CORDIC-алгоритма умножения кватернионов показан ниже.

Алгоритм 1 расчета параметров управления схемы 4D-CONST-Q-CORDIC умножения кватернионов (кватернион q – константа, т. е. угол вращения фиксированный)

Вход: \bar{q} – сопряженный кватернион-константа;

ε – точность CORDIC-операции.

Выход: $\tau(i)$ – неотрицательное целое число сдвигов на i -й итерации;

$\sigma(i) \in \pm 1$ – направление вращения на i -й итерации.

Начало:

1. Установить: $i = 0$ и $\mathbf{x}(i) = \bar{q}$.

2. Определить индекс $m(i)$ наибольшей мнимой части i -й итерации: $x_{m(i)}(i) = \max_{2 \leq l \leq 4} |x_l(i)|$.

3. Определить значение параметра направления вращения: $\sigma(i) = -\text{sign}(x_{m(i)}(i))$.

4. Вычислить параметр числа сдвигов: $\tau(i) = \text{round}(\log_2 |x_{m(i)}(i)|)$.

5. Сохранить $m(i)$, $\sigma(i)$ и $\tau(i)$ как параметры i -й итерации.

6. Установить $\zeta = 1 + 0i + 0j + 0k$ и $\zeta_{m(i)} = \sigma(i)2^{\tau(i)}$.

7. Выполнить итерацию операции CORDIC, чтобы вычислить $\mathbf{x}(i+1) = \mathbf{M}^+(\zeta)\mathbf{x}(i)$.

8. Нормализовать результат итерации: $\mathbf{x}(i+1) = \mathbf{x}(i+1) / |\zeta|$.

9. Если $\forall_{l=2,4} |x_l(i+1)| \leq \varepsilon$, то переход на *Конец*, иначе $i = i + 1$ и переход к п. 2.

Конец.

В табл. 1 приведен пример расчета параметров управления схемы 4D CONST Q-CORDIC-умножения на кватернион-константу $q = [4, -1, 3, -2] / \sqrt{30}$ для точности CORDIC-операции $\varepsilon < 10^{-3}$.

Таблица 1

Пример расчета коэффициентов для кватерниона $q = [4, -1, 3, -2] / \sqrt{30}$

i	$m(i)$	$\tau(i)$	$\sigma(i)$	β_2	β_1	Значение вектора $\mathbf{x}(i)$			
0	–	–	–	–	–	0,730297	0,182574	–0,547723	0,365148
1	2	1	+1	1	0	0,903696	0,387298	0,129099	0,129099
2	1	2	–1	0	1	0,981495	–0,057735	0,173205	0,057735
3	2	3	–1	1	0	0,994199	–0,070014	–0,070014	0,042008
4	2	4	+1	1	0	0,995206	0,053842	–0,074684	0,033000

Окончание табл. 1

i	$m(i)$	$\tau(i)$	$\sigma(i)$	β_2	β_1	Значение вектора $\mathbf{x}(i)$			
5	1	4	+1	0	1	0,996784	0,057519	0,049333	0,026067
6	1	5	-1	0	1	0,998431	-0,004770	0,050863	0,022939
7	3	5	-1	1	1	0,999659	-0,006192	-0,011517	0,022596
8	2	5	-1	1	0	0,999877	-0,006549	-0,011318	-0,008639
9	2	7	+1	1	0	0,999932	-0,006683	0,004305	-0,008535
10	3	7	+1	1	1	0,999943	-0,006749	0,004200	0,007088
11	3	8	-1	1	1	0,999968	-0,006716	0,004253	-0,000724
12	2	8	-1	1	0	0,999990	0,001096	0,004258	-0,000691
13	2	9	+1	1	0	0,999993	0,001101	-0,003554	-0,000682
14	1	10	-1	0	1	0,999999	0,001099	0,000352	-0,000687
15	1	11	+1	0	1	0,999999	-0,000855	0,000351	-0,000687

В предложенном многомерном CORDIC-алгоритме умножение кватернионов микровращения соответствует умножению вырожденных кватернионов, у которых только одна мнимая часть отличается от нуля и может быть разной для разных CORDIC-итераций. Вместе с тем параметры знака и сдвига итераций выбираются таким образом, чтобы итерации могли пропускаться или повторяться. Из табл. 1 видно, что сходимость алгоритма достигается. Исследования проводились на множестве 10^6 случайно сгенерированных единичных кватернионов, точность CORDIC-операции задавалась в диапазоне от 10^{-2} до 10^{-6} . Среднее число итераций колебалось от 9 до 28, а максимальное число итераций – от 15 до 41. Масштабный множитель здесь меньше, чем у 4D-Q-CORDIC-алгоритма, и при равном количестве итераций для каждой разреженной матрицы определяется как $1/K = \left[\prod_{i=0}^{N-1} 1/\sqrt{1+3 \cdot 2^{-2\tau(i)}} \right]^3$.

2.3. Схема микровращений 4D-CONST-Q-CORDIC-алгоритма

Схема микровращений 4D-CONST-Q-CORDIC-алгоритма умножения кватернионов (рис. 1, а) может быть реализована с помощью четырех двухвходовых сумматоров и схемы коммутации ICN (interconnection network). Схема коммутации ICN применяется для выбора ненулевой мнимой части для текущей итерации. Центральное место схемы 4D-CONST-Q-CORDIC-алгоритма занимает схема коммутации ICN (рис. 1, б). Исходя из структуры разреженных матриц умножения вырожденных кватернионов (31)–(33), достаточно четырех вариантов перестановок данных и можно воспользоваться упрощенной сетью Бенесса [23]. Выбирая режимы работы переключающих схем 2×2 между «прямо» и «накрест» (рис. 1, в), можно получить четыре возможных потока перестановки данных (рис. 2).

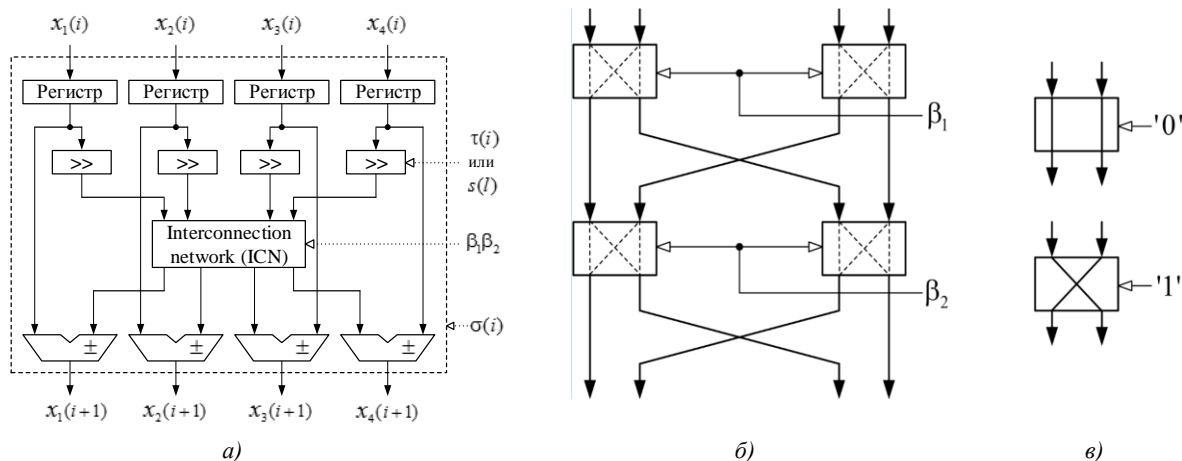


Рис. 1. 4D-CONST-Q-CORDIC-умножения кватернионов: а) схема микровращений 4D-CONST-Q-CORDIC-алгоритма; б) схема коммутации ICN для перестановки данных на основе сети Бенесса; в) состояния переключающих схем 2×2

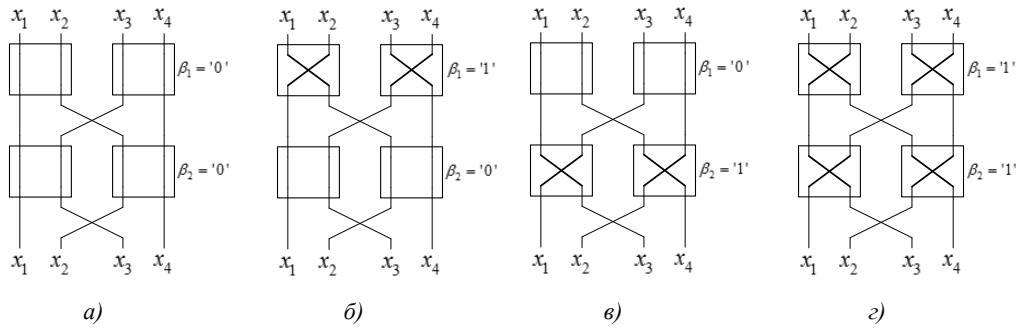


Рис. 2. Возможные варианты перестановки компонентов входного вектора $\mathbf{x}(i)$ на схеме коммутации ICN для перестановки данных: а) путь передачи данных без изменения их порядка; б) операции с мнимой частью i ; в) операции с мнимой частью j ; з) операции с мнимой частью k

Экспериментальные исследования 4D-CONST-Q-CORDIC-алгоритма проводились для аппаратной реализации схемы микровращений на FPGA XC6VLX240T (Xilinx Virtex-6). Для 16-разрядных данных задействовано следующее количество ресурсов FPGA: LUT6-FF – 189, LUT – 105, FF – 81, максимальная частота задающего генератора составляет 315 МГц. Кроме того, если в ICN выбирать путь передачи данных без изменения их порядка (см. рис. 2, а), то схема микровращений 4D-CONST-Q-CORDIC-алгоритма умножения кватернионов может быть использована для реализации этапа масштабирования, который представляется как умножение на вырожденный кватернион с ненулевой действительной компонентой:

$$\mathbf{M}^{\pm}(1 - 2^{-s(l)} + 0i + 0j + 0k) = \begin{bmatrix} 1 - 2^{-s(l)} & 0 & 0 & 0 \\ 0 & 1 - 2^{-s(l)} & 0 & 0 \\ 0 & 0 & 1 - 2^{-s(l)} & 0 \\ 0 & 0 & 0 & 1 - 2^{-s(l)} \end{bmatrix}.$$

Здесь $1 - 2^{-s(l)}$ – аппроксимация масштабного множителя l -й итерации процесса масштабирования, расчет которого приведен в подразд. 3.3. Цикл работы (рис. 3) состоит из последовательно выполняемых шагов итерационного (16 шагов) и масштабирующего (7 шагов) процессов, выбор которых определяется настройкой схемы ICN(β_1, β_2) (значение кода $i_beta(1:0)$). Направление поворота определяет управляющий сигнал $i_sigma(\sigma(i))$, а количество сдвигов – $i_shift(\tau(i))$. Переключение схемы коммутации ICN для выбора соответствующей разреженной матрицы умножения вырожденного кватерниона задает значение кода $i_beta(\beta_1, \beta_2)$. Результат операции микровращения формируется на выходах $i_data0(x_1(i+1))$, $i_data1(x_2(i+1))$, $i_data2(x_3(i+1))$, $i_data3(x_4(i+1))$. Как следует из временной диаграммы, скорость выполнения операции умножения кватернионов обратно пропорциональна суммарному времени итерационного и масштабирующего процессов.

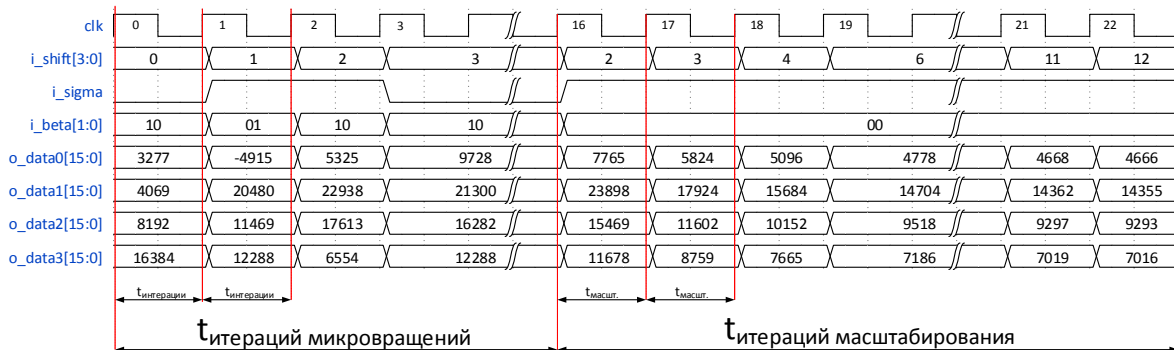


Рис. 3. Временные диаграммы работы схемы микровращений 4D-CONST-Q-CORDIC-алгоритма умножения на кватернион $q = [4, -1, 3, -2] / \sqrt{30}$

3. Целочисленный умножитель на кватернион-константу на основе встроенного 2D-CORDIC-модуля

3.1. Блочная лестничная схемная параметризация оператора умножения кватернионов

Ограничение динамического диапазона арифметики с фиксированной запятой приводит к тому, что в (5) результаты умножений округляются и, следовательно, реализовать обратное преобразование (6) – умножение на сопряженный кватернион – в арифметике с фиксированной запятой не представляется возможным. В результате получается только аппроксимация входной величины. Это является недопустимым для таких приложений, например, как компрессия изображений по схеме L2L (lossless-to-lossy, сжатие и восстановление изображений как без потерь, так и с контролируемым внесением артефактов). Данный недостаток может быть исправлен, если перейти к лестничной схемной параметризации оператора умножения кватернионов, которая позволяет выполнять обратимые «целое к целому» отображения, используя арифметику с фиксированной запятой [13]. Вычислительная сложность операции составит 12 умножений на действительные числа. Однако в настоящей работе не берется во внимание количество операций округления результатов умножений в лестничной структуре, рассматриваемых как белый шум. Число операций округления влияет на компактность энергии в субполосах Q-ПУБФ. Следовательно, количество операций округления должно быть уменьшено в максимально возможной степени для приложений кодирования без потерь. Структура обратимого преобразования на основе блочной лестничной схемной параметризации (рис. 4) представляет собой специальный класс структур обратимых преобразователей с лестничной схемной параметризацией. Количество источников шума здесь сокращается за счет объединения многих операций округления [24].

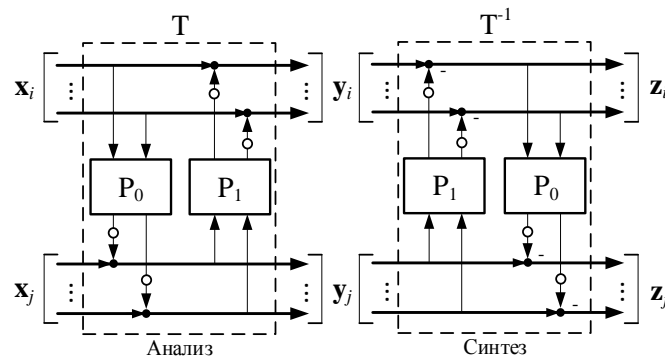


Рис. 4. Структура обратимого преобразователя на основе блочной лестничной схемной параметризации (черные и белые кружки обозначают операции суммирования и округления соответственно)

Векторы \mathbf{x}_i и \mathbf{x}_j на входе и на выходе \mathbf{y}_i и \mathbf{y}_j прямого преобразования, а также векторы \mathbf{z}_i и \mathbf{z}_j на выходе обратного преобразования и блоки преобразования – матрицы \mathbf{P}_0 и \mathbf{P}_1 – соотносятся следующим образом: $\mathbf{y}_j = \mathbf{x}_j + \text{round}(\mathbf{P}_0 \mathbf{x}_i)$, $\mathbf{y}_i = \mathbf{x}_j + \text{round}(\mathbf{P}_1 \mathbf{y}_j)$; $\mathbf{z}_i = \mathbf{y}_i - \text{round}(\mathbf{P}_1 \mathbf{y}_j) = \mathbf{x}_i$, $\mathbf{z}_j = \mathbf{y}_j - \text{round}(\mathbf{P}_0 \mathbf{y}_i) = \mathbf{x}_j$, где $\text{round}(\cdot)$ – оператор округления. В данном случае матрицы преобразования и их инверсные матрицы определяются как

$$\begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}, \begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_j \end{bmatrix} = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix},$$

$$\text{где } \mathbf{T} = \mathbf{P}_U \mathbf{P}_L, \mathbf{T}^{-1} = \mathbf{P}_L^{-1} \mathbf{P}_U^{-1}, \mathbf{P}_U = \begin{bmatrix} \mathbf{I} & \mathbf{P}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{P}_U^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{P}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{P}_L = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{P}_0 & \mathbf{I} \end{bmatrix}, \mathbf{P}_L^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{P}_0 & \mathbf{I} \end{bmatrix}.$$

Таким образом, следуя философии блочной лестничной схемной параметризации, матрицу оператора левого умножения кватернионов $\mathbf{M}^+(q)$ можно представить в виде блочной матрицы:

$$\mathbf{M}^+(q) = \begin{bmatrix} \mathbf{C}(q) & -\mathbf{S}(q) \\ \mathbf{S}(q) & \mathbf{C}(q) \end{bmatrix}, \quad \mathbf{C}(q) = \begin{bmatrix} q_1 & -q_2 \\ q_2 & q_1 \end{bmatrix}, \quad \mathbf{S}(q) = \begin{bmatrix} q_3 & q_4 \\ q_4 & -q_3 \end{bmatrix}.$$

Далее на основе известной [25] лестничной схемной факторизации двумерной матрицы вращения

$$\mathbf{R}_2(\varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} = \begin{bmatrix} 1 & \cos \varphi - 1 / \sin \varphi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin \varphi & 1 \end{bmatrix} \begin{bmatrix} 1 & \cos \varphi - 1 / \sin \varphi \\ 0 & 1 \end{bmatrix}$$

можно получить факторизацию матрицы умножения кватернионов

$$\mathbf{M}^+(q) = \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{F}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G}(q) & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{H}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}(q)}. \quad (34)$$

Для заданного коэффициента q и матрицы умножения $\mathbf{M}^+(q)$ определяется набор матричных выражений, которые могут быть решены однозначно для $\mathbf{F}(q)$, $\mathbf{G}(q)$ и $\mathbf{H}(q)$ при условии, что матрица $\mathbf{S}(q)$ является несингулярной или, более определенно, ненулевой:

$$\mathbf{F}(q) = (\mathbf{C}(q) - \mathbf{I}_2)\mathbf{S}(q)^{-1}, \quad \mathbf{G}(q) = \mathbf{S}(q), \quad \mathbf{H}(q) = \mathbf{S}(q)^{-1}(\mathbf{C}(q) - \mathbf{I}_2). \quad (35)$$

Элементы данных матриц являются вещественными коэффициентами лестничной схемной параметризации.

В случае если в прямом преобразовании Q-ПУБФ используется умножение на нормированный кватернион ($|q|=1$), обратное преобразование основывается на умножении на сопряженный кватернион \bar{q} . Из равенства $\mathbf{M}^\pm(\bar{q}) = \mathbf{M}^\mp(q)^T$ следует, что $\mathbf{M}^+(\bar{q}) = \mathbf{V}(q)^T \mathbf{L}(q)^T \mathbf{U}(q)^T$, но здесь в отличие от факторизации (34) нижние треугольные матрицы заменяются на верхние треугольные матрицы, и наоборот. Однако, принимая во внимание, что $\mathbf{M}^+(\bar{q}) = -\mathbf{P}\mathbf{M}^+(q)\mathbf{P}$, где $\mathbf{P} = \Gamma_4 \mathbf{J}_4$ и $\mathbf{P}^2 = -\mathbf{I}_4$, факторизация оператора умножения на сопряженный кватернион имеет вид

$$\mathbf{M}^+(\bar{q}) = \underbrace{(-\mathbf{P})\mathbf{V}(q)^T \mathbf{P}}_{\mathbf{U}(\bar{q})} \underbrace{(-\mathbf{P})\mathbf{L}(q)^T \mathbf{P}}_{\mathbf{L}(\bar{q})} \underbrace{(-\mathbf{P})\mathbf{U}(q)^T \mathbf{P}}_{\mathbf{V}(\bar{q})}$$

и упрощается:

$$\mathbf{M}^+(\bar{q}) = \pm \underbrace{\begin{bmatrix} \mathbf{I}_2 & -\mathbf{H}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}(\bar{q})} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ -\mathbf{G}(q) & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}(\bar{q})} \underbrace{\begin{bmatrix} \mathbf{I}_2 & -\mathbf{F}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}(\bar{q})}. \quad (36)$$

Из уравнений (5) и (6) факторизации полифазных передаточных матриц анализа $\mathbf{E}(z)$ и синтеза $\mathbf{D}(z)$ решетчатой структуры Q-ПУБФ следует, что необходимо переключение между операторами левого и правого умножений, а не только между оригинальным и сопряженным кватернионами. Принимая во внимание тот факт, что оператор сопряжения кватернионов есть собственно оператор получения обратного кватерниона, можно записать

$$\mathbf{M}^{-}(q) = \underbrace{\mathbf{D}_C \mathbf{U}(\bar{q}) \mathbf{D}_C}_{\mathbf{U}^{-}(q)} \underbrace{\mathbf{D}_C \mathbf{L}(\bar{q}) \mathbf{D}_C}_{\mathbf{L}^{-}(q)} \underbrace{\mathbf{D}_C \mathbf{V}(\bar{q}) \mathbf{D}_C}_{\mathbf{V}^{-}(q)}$$

или в форме блочной лестничной схемной параметризации

$$\mathbf{M}^{-}(q) = \underbrace{\begin{bmatrix} \mathbf{I}_2 & \Gamma_2 \mathbf{H}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}^{-}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G}(q) \Gamma_2 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}^{-}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & \Gamma_2 \mathbf{F}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}^{-}(q)}.$$

Далее, представляя матрицу $\mathbf{M}^{-}(q)$ в виде блочной матрицы

$$\mathbf{M}^{-}(q) = \begin{bmatrix} \mathbf{C}(q) & -\bar{\mathbf{S}}(q) \\ \bar{\mathbf{S}}(q) & \mathbf{C}(q) \end{bmatrix}, \quad \bar{\mathbf{S}}(q) = \mathbf{S}(q) \text{diag}(1, -1) = \begin{bmatrix} q_3 & -q_4 \\ q_4 & q_3 \end{bmatrix},$$

можно получить ту же схему умножителя кватернионов для правого оператора умножения, что и для левого оператора.

Структура обратимого оператора умножения кватернионов на основе блочной лестничной схемной параметризации (рис. 5) представляет собой каскадное соединение блочных матриц $\mathbf{H}(q)$, $\mathbf{G}(q)$ и $\mathbf{F}(q)$ (прямое преобразование) и их инверсное включение с отрицательным знаком (обратное преобразование). Здесь белые кружки обозначают выходы, результат которых округлен.

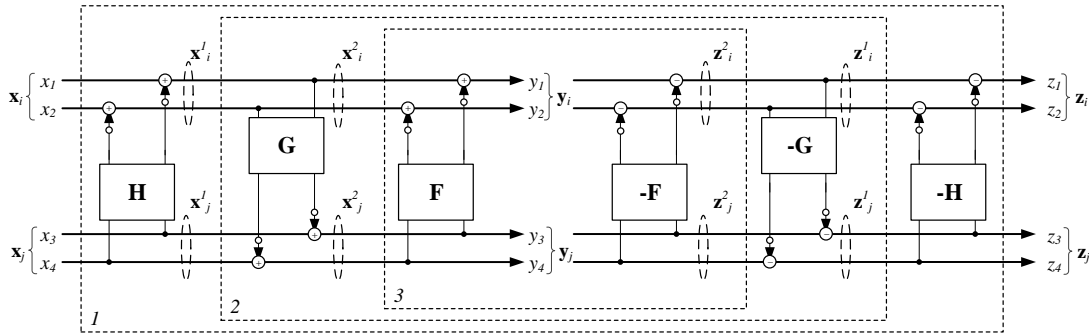


Рис. 5. Структура обратимого оператора умножения кватернионов

В результате прямого преобразования входной вектор $\mathbf{x} = [x_i, x_j]^T$ проходит три стадии обработки для формирования результата преобразования (умножения вектора \mathbf{x} на коэффициент q) $\mathbf{y} = [y_i, y_j]^T$:

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{H} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_i + \text{round}[\mathbf{H}x_j] \\ x_j \end{bmatrix} = \begin{bmatrix} x_i^1 \\ x_j^1 \end{bmatrix};$$

$$\mathbf{x}_2 = \begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G} & \mathbf{I}_2 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} x_i^1 \\ \text{round}[\mathbf{G}x_i^1] + x_j^1 \end{bmatrix} = \begin{bmatrix} x_i^2 \\ x_j^2 \end{bmatrix};$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{F} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{x}_2 = \begin{bmatrix} x_i^2 + \text{round}[\mathbf{F}x_j^2] \\ x_j^2 \end{bmatrix} = \begin{bmatrix} y_i \\ y_j \end{bmatrix}.$$

Выходной результат обратного преобразования $\mathbf{z} = [z_i z_j]^T$ получается также после трех ступенной обработки результата прямого преобразования $\mathbf{y} = [y_i y_j]^T$:

$$\begin{aligned} \mathbf{z}_2 &= \begin{bmatrix} \mathbf{I}_2 & -\mathbf{F} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{y} = \begin{bmatrix} x_i^1 + \text{round}[\mathbf{F}x_j^2] - \text{round}[\mathbf{F}x_j^2] \\ x_j^2 \end{bmatrix} = \begin{bmatrix} x_i^1 \\ x_j^2 \end{bmatrix} = \mathbf{x}_2; \\ \mathbf{z}_1 &= \begin{bmatrix} \mathbf{I}_2 & 0 \\ -\mathbf{G} & \mathbf{I}_2 \end{bmatrix} \mathbf{z}_2 = \begin{bmatrix} x_i^1 \\ -\text{round}[\mathbf{G}x_i^1] + \text{round}[\mathbf{G}x_i^1] + x_j^2 \end{bmatrix} = \begin{bmatrix} x_i^1 \\ x_j^2 \end{bmatrix} = \mathbf{x}_1; \\ \mathbf{z} &= \begin{bmatrix} \mathbf{I}_2 & -\mathbf{H} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{z}_1 = \begin{bmatrix} x_i + \text{round}[\mathbf{H}x_j] - \text{round}[\mathbf{H}x_j] \\ x_j \end{bmatrix} = \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \mathbf{x}. \end{aligned}$$

Таким образом, умножение на $1/q$ или на эквивалентный сопряженный кватернион \bar{q} реализуется переупорядочиванием коэффициентов в обратном порядке и изменением их знака. Следует отметить, что округление лестничных коэффициентов и результата соответствующего умножения не влияет на обратимость преобразования. Следовательно, выполнение обратимого «целое к целому» отображения на основе оператора умножения возможно для арифметики с фиксированной запятой.

3.2. Прямая и обратная схемы множителя кватернионов на основе блочной лестничной схемной параметризации со встроенным 2D-CORDIC-модулем

Величины углов в пределах $-\pi \leq \varphi < \pi$, $-\pi/2 \leq \psi \leq \pi/2$ и $-\pi/2 \leq \chi \leq \pi/2$ достаточно, чтобы описать любой кватернион заданного модуля $|q|$ и трех углов φ , ψ , χ в полярной форме: $q = |q|e^{i\varphi}e^{j\psi}e^{k\chi}$, где $q_1 = |q|\cos\varphi$, $q_2 = |q|\sin\varphi\cos\psi$, $q_3 = |q|\sin\varphi\sin\psi\cos\chi$, $q_4 = |q|\sin\varphi\sin\psi\sin\chi$. Для $|q|=1$ расчетные выражения блочной лестничной схемной параметризации (35) оператора умножения кватернионов могут быть представлены следующим образом:

$$\begin{aligned} f_{11}(q) &= -f_{22}(q) = [\cos\chi(\cos\varphi - 1) - \sin\varphi\cos\psi\sin\chi]/d, \\ f_{12}(q) &= f_{21}(q) = [\sin\chi(\cos\varphi - 1) + \sin\varphi\cos\psi\cos\chi]/d, \\ g_{11}(q) &= -g_{22}(q) = d\cos\chi, \\ g_{12}(q) &= g_{21}(q) = d\sin\chi, \\ h_{11}(q) &= -h_{22}(q) = [\cos\chi(\cos\varphi + 1) + \sin\varphi\cos\psi\sin\chi]/d, \\ h_{12}(q) &= h_{21}(q) = [\sin\chi(\cos\varphi + 1) - \sin\varphi\cos\psi\cos\chi]/d, \\ d &= \sin\varphi\sin\psi. \end{aligned} \tag{37}$$

Анализ (37) показывает, что в компонентах $\mathbf{F}(q)$, $\mathbf{G}(q)$ и $\mathbf{H}(q)$ факторизации матрицы умножения кватернионов $\mathbf{M}^+(q)$ (34) есть пары коэффициентов с одинаковым модулем, что не очевидно из (35) и не принималось во внимание, когда создавалась факторизация (34). Следовательно, структуры матриц $\mathbf{F}(q)$, $\mathbf{G}(q)$ и $\mathbf{H}(q)$ имеют структуру, близкую к матрице $\mathbf{R}_2(\varphi)$ вращения Гивенса $\begin{bmatrix} s & c \\ c & -s \end{bmatrix} = \mathbf{J}_2 \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, где через c и s обозначены косинус и синус соответствующего угла. Например, матрица $\mathbf{F}(q)$ имеет вид

$$\begin{bmatrix} f_{11}(q) & f_{12}(q) \\ f_{12}(q) & -f_{11}(q) \end{bmatrix} = \mathbf{J}_2 \begin{bmatrix} f_{12}(q) & -f_{11}(q) \\ f_{11}(q) & f_{12}(q) \end{bmatrix}.$$

Таким образом, данные матрицы могут быть аппроксимированы с использованием 2D-CORDIC-алгоритма, а зависимости (37) являются расчетными выражениями блочной лестничной схемной параметризации оператора умножения кватернионов и формируют матрицы вращения Гивенса для соответствующих этапов разложения $\mathbf{U}(q)$, $\mathbf{L}(q)$, $\mathbf{V}(q)$.

Динамический диапазон схемы умножителя кватернионов (рис. 6) ограничен диапазоном $-1..1$, что обусловлено форматом фиксированной запятой. Однако коэффициенты, полученные с помощью расчетных выражений блочной лестничной схемной параметризации (37) оператора умножения кватернионов, могут иметь абсолютные значения больше единицы. Приведение параметров умножителя к требуемому динамическому диапазону может быть достигнуто заменой умножения кватерниона q на кватернион \tilde{q} , который представляет собой версию кватерниона q с переставленными и (или) измененными знаками компонент. Матрицы оператора умножения кватернионов $\mathbf{M}^{\pm}(q)$ и $\mathbf{M}^{\pm}(\tilde{q})$ будут соотноситься согласно следующему выражению:

$$\mathbf{M}^{\pm}(q) = \mathbf{P}_{\text{post}} \mathbf{M}^{\pm}(\tilde{q}) \mathbf{P}_{\text{pre}}, \quad (38)$$

где матрицы \mathbf{P}_{pre} и \mathbf{P}_{post} являются матрицами перестановок исходного кватерниона q и кватерниона \tilde{q} с переставленными и (или) измененными знаками компонент кватерниона q . На практике это соответствует лишь выполнению пред- и постобработки входного операнда \mathbf{x} и выходного результата соответственно: $qx = \mathbf{M}^{\pm}(q) \mathbf{x} = \mathbf{P}_{\text{post}} \mathbf{M}^{\pm}(\tilde{q}) \mathbf{P}_{\text{pre}} \mathbf{x}$.

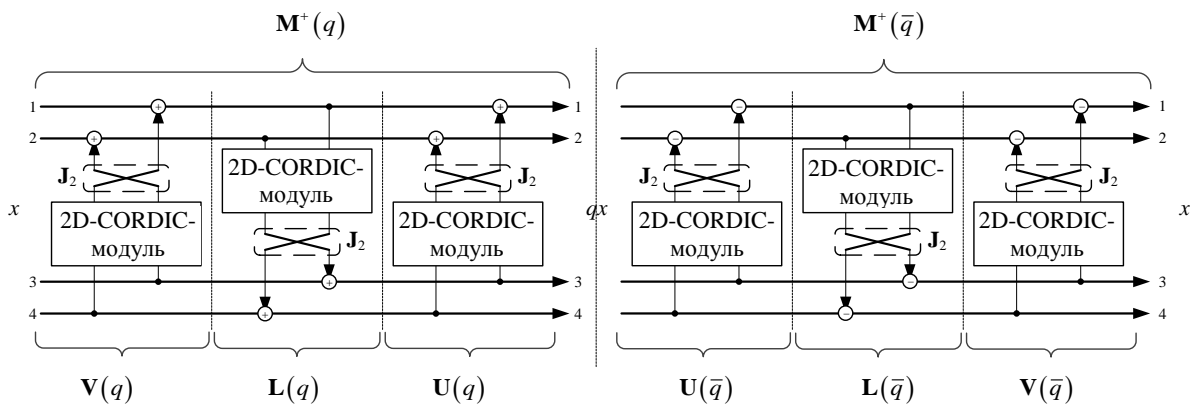


Рис. 6. Прямая и обратная схемы умножителя кватернионов на основе блочной лестничной параметризации со встроенным 2D-CORDIC-модулем

Таким образом, гибридная схема «CORDIC – лестничная параметризация» позволяет интегрировать CORDIC-алгоритм «внутри» лестничной схемы умножителя, заменив действительные умножения на микроповороты CORDIC-алгоритма: сложение и сдвиг, а также получить совершенное преобразование (ПУБФ) и в аппаратной реализации [26]. Неотъемлемым свойством блочных лестничных схем является то, что все результаты между этапами факторизации могут быть вычислены сразу, без использования вспомогательной памяти. Небольшая разница в схемах для прямого и обратного умножения (умножения на сопряженный кватернион) дает возможность проектировать универсальный умножитель. Отличие лишь в знаке для коэффициентов прямого и обратного преобразования, что позволяет иметь одну схему с переключаемыми режимами. Еще одно преимущество предложенной схемы умножителя кватернионов заключается в том, что число операций округления уменьшилось почти в два раза по сравнению со структурой умножителя на основе лестничной схемной параметризации [27].

3.3. Оптимизация параметров встроенного 2D-CONST-CORDIC-модуля

Задача выбора параметров схемы умножителя кватернионов со встроенным 2D-CORDIC-алгоритмом – наиболее сложная часть проекта. Как отмечалось выше, при фиксированном угле вращения φ с целью достижения заданной точности за минимальное число итераций M параметры управления для итерационного и масштабирующего процессов CORDIC-алгоритма могут быть определены и оптимизированы заранее во время проектирования схемы CORDIC-алгоритма:

– фиксированный угол φ предопределяет некоторое множество базисных углов CORDIC-алгоритма $\{\varphi(i) = \arctan(2^{-\tau(i)}) \text{ для } 0 \leq i \leq M-1\}$. Для того чтобы повернуть вектор $[x_1, x_2]^T$ на угол φ для получения вектора $[x_1^{out}, x_2^{out}]^T$, CORDIC-алгоритм (8)–(15) может быть модифицирован следующим образом:

$$\begin{bmatrix} x_1(i+1) \\ x_2(i+1) \end{bmatrix} = \begin{bmatrix} 1 & -\sigma(i)2^{-\tau(i)} \\ \sigma(i)2^{-\tau(i)} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(i) \\ x_2(i) \end{bmatrix}, \quad \begin{bmatrix} x_1^{out} \\ x_2^{out} \end{bmatrix} \leftarrow \frac{1}{K} \begin{bmatrix} x_1(M) \\ x_2(M) \end{bmatrix}; \quad (39)$$

– масштабный фактор $1/K$ сейчас зависит от множества базисных углов $\{\varphi(i)\}$, а точность алгоритма определяет, как близко фактический угол вращения φ_A , полученный за M микровращений, аппроксимирует заданный угол вращения φ . Это, в свою очередь, определяет отклонение фактического вектора вращения от требуемого.

Произвольное значение масштабирующего фактора $1/K = \prod_{i=0}^{M-1} (1 + 2^{-2\tau(i)})^{-1/2}$ при $1/K < 1$ может быть аппроксимировано выражением [14]

$$\frac{1}{K} \approx \prod_{l=0}^{L-1} [1 - 2^{-s(l)}], \quad (40)$$

где L и $s(l) \in \mathbb{N} \setminus \{0\}$ выбраны как продукт аппроксимации $1/K$, а процесс масштабирования описывается рекурсивным выражением

$$\mathbf{x}(l+1) = \mathbf{x}(l) - 2^{-s(l)} \mathbf{x}(l), \quad l = 0, 1, \dots, L-1, \quad \mathbf{x}(0) = \mathbf{x}(M), \quad \mathbf{x}(L) = \mathbf{x}_{out}(M). \quad (41)$$

Таким образом, для фиксированного угла вращения параметры управления для итерационного и масштабирующего процессов CORDIC-алгоритма могут быть оптимизированы за два шага:

1. Приведение угла к заданным ограничениям. Преобразования (8), (9) или (39), (40) применимы только для вращений на углы, для которых $\cos \varphi \geq 0$ и $|\cos \varphi| \geq |\sin \varphi|$, т. е. на углы $|\varphi| \leq \frac{\pi}{4}$ [27]. Так, для осуществления вращения на произвольный угол φ , не удовлетворяющий данным условиям, всегда могут быть реализованы простой пред- и постобработкой матрицы вращений \mathbf{R}_2 с подходящим углом $\tilde{\varphi}$ [14]:

$$\mathbf{R}_2(\varphi) = \mathbf{P}_{post}^{CORDIC} \mathbf{R}_2(\tilde{\varphi}) \mathbf{P}_{pre}^{CORDIC}, \quad (42)$$

в частности

$$\mathbf{R}_2(\varphi) = \mathbf{\Gamma}_2 \mathbf{R}_2(-\varphi) \mathbf{\Gamma}_2, \quad \mathbf{R}_2(\varphi) = \mathbf{J}_2 \mathbf{R}_2(-\varphi) \mathbf{\Gamma}_2, \quad \mathbf{R}_2(\varphi) = -\mathbf{J}_2 \mathbf{R}_2(\pi - \varphi) \mathbf{J}_2, \quad (43)$$

т. е. достаточно поменять местами величины и (или) их знаки. Здесь \mathbf{J}_2 – обратная единичная матрица 2×2 , а $\mathbf{\Gamma}_2 = \text{diag}(1, -1)$.

2. Минимизация числа базисных углов в множестве $\{\varphi(i)\}$ для достижения требуемой точности вычисления преобразования: определяются параметры $\tau(i)$, M и $s(l)$, L .

Простой псевдокод программы алгоритма расчета операторов сдвига $\tau(i)$ и направления вращения $\sigma(i)$ для схемы 2D-CONST-Q-CORDIC при фиксированном угле вращения показан

ниже. Заданная максимальная точность ε_φ определяется как максимальная ошибка Δ_φ между матрицей вращения \mathbf{M} , определенной заданным углом – входным параметром $\mathbf{q} = [q_1, q_2]^T$, и ее аппроксимацией $\mathbf{IR}_2(i)$. Алгоритм оптимизации поиска параметров $\tau(i)$ и $\sigma(i)$ базируется на минимизации целевой функции $\max|\Delta_\varphi|$. Последовательность матриц приближения $\mathbf{IR}_2(i)$, $i = 0, 1, 2, \dots$, определяется CORDIC-операцией, для которой матрица вращения формируется из параметров управления $\tau(i)$ и $\sigma(i)$, рассчитанных на i -й итерации. Для $i = 0$ существует единичная матрица вращения вектора, лежащего на оси абсцисс, вокруг центра декартовых координат на угол, определенный вектором $\mathbf{q} = [q_1, q_2]^T$ ($\text{Re } q = q_1, \text{Im } q = q_2$). При этом входной вектор $\mathbf{x}(i) = [x_1(i), x_2(i)]^T$ равен комплексно-сопряженному вектору $\bar{\mathbf{q}} = [q_1, -q_2]^T$ ($|\mathbf{q}| = |\bar{\mathbf{q}}|$).

Алгоритм 2 расчета параметров управления $\tau(i)$ и $\sigma(i)$ 2D-CONST-CORDIC-схемы

Вход: $\mathbf{q} = [q_1, q_2]^T$ ($\text{Re } q = q_1, \text{Im } q = q_2$) – комплексное число-константа, т. е. угол вращения фиксированный;

ε_φ – точность аппроксимации заданного угла вращения.

Выход: $\tau(i)$ – неотрицательное целое число сдвигов на i -й итерации;

$\sigma(i) \in \pm 1$ – направление вращения на i -й итерации.

Начало:

1. Сформировать матрицы вращения $\mathbf{M} = \begin{bmatrix} q_1 & -q_2 \\ q_2 & q_1 \end{bmatrix}$ и единичную $\mathbf{IR}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

2. Установить: $i = 0$, $\mathbf{x}(i) = \bar{\mathbf{q}}$, $\mathbf{x}(i) = [x_1(i), x_2(i)]^T$ и $\mathbf{IR}_2(i) = \mathbf{IR}_2$.

3. Определить значение параметра направления вращения $\sigma(i) = -\text{sign}[x_2(i)]$.

4. Вычислить параметр числа сдвигов $\tau(i) = \text{round}(\log_2 |x_2(i)/x_1(i)|)$.

5. Сохранить $\sigma(i)$ и $\tau(i)$ как параметры i -й итерации.

6. Сформировать $\chi = 1 + j\sigma(i)2^{\tau(i)}$, $j = \sqrt{-1}$.

7. Выполнить итерацию аппроксимации: $\mathbf{IR}_2(i+1) = \mathbf{R}_2(\chi)\mathbf{IR}_2(i)$, где $\mathbf{R}_2(\chi)$ – матрица вращения (9).

8. Нормализовать результат итерации: $\mathbf{IR}_2(i+1) = \mathbf{IR}_2(i+1)/|\chi|$.

9. Вычислить ошибку аппроксимации $\Delta_\varphi = \mathbf{IR}_2(i+1) - \mathbf{M}$.

10. Если $\max|\Delta_\varphi| \leq \varepsilon_\varphi$, то $M = i$, иначе выполнить итерацию CORDIC-операции $\mathbf{x}(i+1) = \mathbf{R}_2(\chi)\mathbf{x}(i)$ и $i = i + 1$, перейти к п. 3.

Конец.

Алгоритм 3 описывает задачу оптимизации поиска параметров L и $s(l)$ процесса масштабирования 2D-CONST-CORDIC-схемы. Масштабный множитель, обусловленный выполнением итерационного процесса с параметрами $\tau(i)$ и $\sigma(i)$, которые получены в результате выполнения алгоритма 2, вычисляется как $1/K = \prod_{i=0}^{M-1} (1 + 2^{-2\tau(i)})^{-1/2}$. Целевая функция ΔK определяет, на сколько отношение масштабного множителя $1/K$ к его аппроксимированному значению $(1/K)_A$ отличается от единицы, т. е. $\Delta K = |(1/K)/(1/K)_A - 1|$. Алгоритм начинает работу с вычисления аппроксимированного значения масштабного множителя для одного сдвига, а затем число масштабирующих сдвигов увеличивается на единицу до тех пор, пока ΔK не станет больше заданной точности ε_K операции масштабирования. Величина ε_K равна величине

не ε_ϕ в алгоритме 2, потому что ошибки ΔK и Δ_ϕ вносят равный вклад в общую ошибку 2D-CONST-CORDIC-схемы.

Алгоритм 3 расчета параметров L и $s(l)$ процесса масштабирования

Вход: $1/K = \prod_{i=0}^{M-1} (1 + 2^{-2\tau(i)})^{-1/2}$ – масштабный множитель;

ε_K – точность операции масштабирования.

Выход: L и $s(l) \in \mathbb{N} \setminus \{0\}$ – неотрицательное целое число сдвигов на l -й итерации процесса масштабирования.

Начало:

1. Установить $A=1, k=1, l=0$.
2. Вычислить $(1/K)_A = A(1-2^{-k})$.
3. Если $(1/K)_A \geq (1/K)$, то сохранить число сдвигов на l -й итерации процесса масштабирования $s(l) = k$ и $A = (1/K)_A$, иначе $k = k+1$ и переход к п. 2.
4. Оценить точность аппроксимации $\Delta K = |(1/K)/(1/K)_A - 1|$.
5. Если $\Delta K \leq \varepsilon_K$, то $L = k$, иначе переход к п. 2.

Конец.

В отличие от стандартного CORDIC-алгоритма в операции масштабирования параметры $\tau(i)$, M и L представляются произвольными значениями. Например, выбор количества сдвигов $\tau(i) \in \mathbb{N}$ не зависит от номера итерации i , но при этом может быть получена требуемая точность за меньшее число микровращений [28].

Анализ схем на рис. 7 показывает, что итерационный и масштабирующий процессы могут быть реализованы на одной коммутируемой во времени CORDIC-схеме. При этом затраты оборудования будут минимальны, а производительность данного процессора будет обратно пропорциональна сумме числа микровращений и итераций процесса масштабирования. С целью увеличения скорости работы 2D-CORDIC-процессора вычисления (39) и (41) могут осуществляться по конвейерной CORDIC-схеме [15].

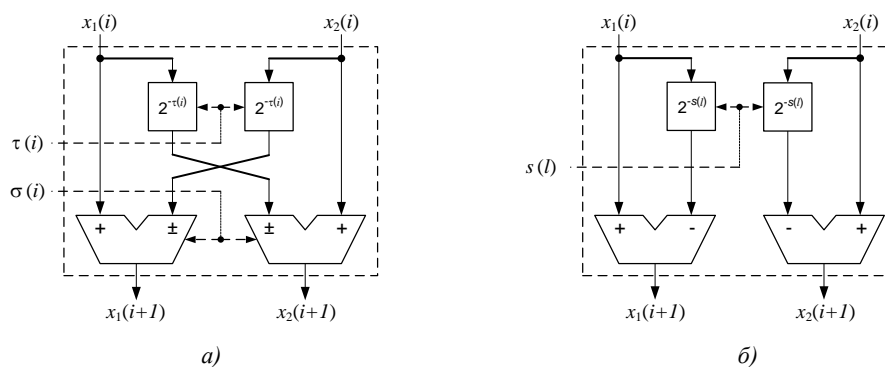


Рис. 7. Схемы для реализации: а) микровращений; б) итераций масштабирования 2D-CORDIC-алгоритма для априори известного угла

Процесс оптимизации параметров встроенной 2D-CONST-CORDIC-схемы выполняется в три этапа:

- 1) для кватерниона q рассматриваются все модификации гиперкомплексного числа, потенциально определяющие вычислительную схему умножителя, в которой все коэффициенты лестничной схемы находятся в диапазоне $-1...1$;
- 2) для каждого такого модифицированного кватерниона \tilde{q} осуществляется формирование матриц $\mathbf{P}_{pre}^{CORDIC}$ и $\mathbf{P}_{post}^{CORDIC}$ (42)–(43), рассчитываются параметры и оценивается число итераций 2D-CONST-CORDIC-алгоритма (число микровращений итерационного процесса и количе-

ство итераций процесса масштабирования), гарантирующее заданную точность операции умножения на кватернион-константу;

3) выбирается соответствующий кандидат из множества модифицированных кватернионов \tilde{q} , для которого параметры схемы наиболее близко согласуются с принятой платформой реализации (число итераций должно быть минимальным). Это обеспечивает максимальную скорость выполнения операции умножения.

Например, для кватерниона $q = q_1 + q_2i + q_3j + q_4k = 1/\sqrt{30}(4 - i + 3j - 2k)$ процесс настройки параметров умножителя на основе 2D-CONST-CORDIC-модуля выглядит следующим образом. На первом этапе для обеспечения заданного динамического диапазона из 24 возможных модификаций исходного кватерниона $q = q_1 + q_2i + q_3j + q_4k$ только у семи кватернионов \tilde{q} , представляющих собой версии кватерниона q с переставленными и (или) измененными знаками компонент, коэффициенты блочной лестничной схемной параметризации (37) попадают в динамический диапазон $-1 \dots 1$. У модифицированного кватерниона $\tilde{q} = q_3 + q_2i + q_1j + q_4k$ коэффициенты блочной лестничной факторизации (37) следующие: $f_{11}(q) = -f_{22}(q) = -0,395$, $f_{12}(q) = f_{21}(q) = 0,488$, $g_{11}(q) = -g_{22}(q) = 0,730$, $g_{12}(q) = g_{21}(q) = 0,365$, $h_{11}(q) = -h_{22}(q) = -0,595$, $h_{12}(q) = h_{21}(q) = 0,048$. Далее, на втором и третьем этапах процесса оптимизации параметров встроенного 2D-CONST-CORDIC-алгоритма расчет числа итераций (алгоритмы 2 и 3) 2D-CORDIC-модуля для каждого этапа разложения $U(q)$, $L(q)$, $V(q)$ схемы умножителя кватернионов (см. рис. 6) показал, что у данного модифицированного кватерниона $\tilde{q} = q_3 + q_2i + q_1j + q_4k$ для точности ε_ϕ и ε_K порядка 10^{-5} требуется минимальное число микровращений итерационного процесса 7 ($M_u = 3$, $M_L = 1$, $M_V = 3$) и итераций процесса масштабирования 17 ($L_u = 7$, $L_L = 5$, $L_v = 5$). В табл. 2 для кватерниона \tilde{q} приведены параметры управления $\tau(i)$ и $\sigma(i)$, а также L и $s(l)$ процесса масштабирования для каждой компоненты разложения $U(q)$, $L(q)$, $V(q)$. Для реализации умножения на кватернион q через умножение на кватернион \tilde{q} должна быть выполнена следующая пред- и постобработка (38), представленная в виде матриц перестановок:

$$\mathbf{P}_{\text{pre}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{P}_{\text{post}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Таблица 2

Параметры управления 2D-CONST-CORDIC-модулями схемы умножения кватернионов на основе блочной лестничной параметризации

Параметр управления	Компоненты разложения		
	U $M=3, L=7$	L $M=1, L=5$	V $M=3, L=5$
$\sigma(0)$	1	-1	1
$\tau(0)$	0	1	4
$\sigma(1)$	-1		1
$\tau(1)$	4		6
$\sigma(2)$	1		1
$\tau(2)$	11		9
$s(0)$	1	2	2
$s(1)$	3	6	3
$s(2)$	5	7	4

$s(3)$	8	9	6
$s(4)$	10	10	6
$s(5)$	12		
$s(6)$	13		
$\mathbf{P}_{pre}^{CORDIC}$	$-\Gamma_2$	\mathbf{I}_2	$-\mathbf{I}_2$
$\mathbf{P}_{post}^{CORDIC}$	\mathbf{J}_2	\mathbf{I}_2	\mathbf{I}_2

Рассмотрим структурную схему операции микровращения 2D-CONST-CORDIC-алгоритма (рис. 8). Ядро схемы образуют два сумматора, работающих в режиме суммирования и вычитания, два регистра сдвига вправо и модуль коммутации ICN, основное назначение которого заключается в переключении выходов регистров сдвига в соответствии с режимами работы 2D-CONST-CORDIC-схемы: итерационным процессом (выходы коммутируются накрест) и масштабирующим процессом (выходы коммутируются прямо). Компоненты $x_1(i)$ и $x_2(i)$ загружаются во входные регистры и передаются в сумматоры согласно схеме. Содержимое сумматоров $x_1(i+1)$, $x_2(i+1)$ загружается во входные регистры для следующей CORDIC-итерации. Для повышения производительности 2D-CORDIC-модуля в схеме умножителя кватернионов на основе блочной лестничной параметризации можно использовать конвейерное включение 2D-CONST-CORDIC-схем микровращения. При этом латентность операции будет определяться числом схем микровращения в конвейере и производительность модуля возрастет во столько же раз. Прототипирование схемы осуществлялось на ПЛИС с архитектурой FPGA XC6VLX240T (Xilinx Virtex-6).

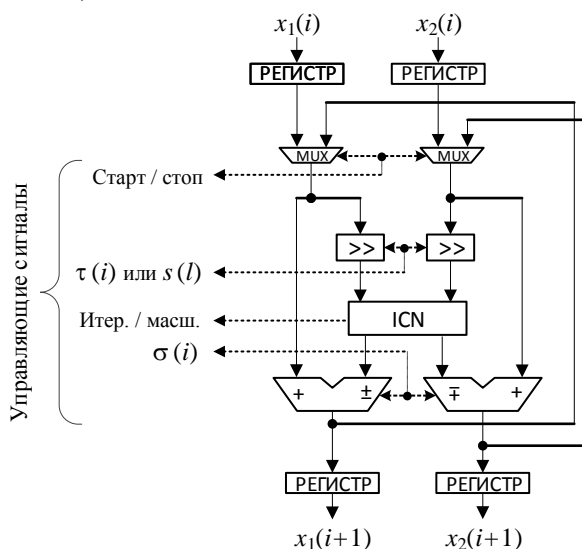


Рис. 8. Структурная 2D-CONST-CORDIC-схема операции микровращения

Аппаратные затраты умножителя кватернионов 4D-CONST-Q-CORDIC (см. рис. 2) приблизительно на 50 % меньше, чем у умножителя кватернионов с блочной лестничной структурой на основе 2D CONST CORDIC (см. рис. 6). Это обусловлено дополнительными аппаратными затратами на встраивание 2D-CONST-CORDIC-модуля в схему умножителя для динамического диапазона обработки данных в пределах диапазона $-1...1$ и вращения на произвольный угол (реализации матриц перестановок (38) и (43)). Однако только схема умножения кватернионов на основе блочной лестничной параметризации со встроенными 2D-CONST-CORDIC-модулями дает возможность реализации умножителя кватернионов как обратимого оператора умножения кватернионов в целочисленной арифметике. Данную схему можно рассматривать как структурное решение для цифровой обработки медиаданных без потерь, когда данные и алгоритмы представляются в алгебре кватернионов. В работе [28] предложен кон-

вейерный процессор умножителя кватернионов с блочной лестничной параметризацией на основе 2D-CONST-CORDIC-процессора, производительность которого выше, чем у 4D-CONST-Q-CORDIC-процессора, который по определению не допускает организации поточной обработки данных. Сбалансированного решения между производительностью умножителя, ядром которого является 2D-CONST-CORDIC, и аппаратными затратами, как показано в [29], можно достигнуть в схеме рекурсивного процессора умножителя кватернионов.

Моделирование в программной среде MATLAB с помощью инструментального модуля Fixed Point системы анализа-синтеза с линейной фазочастотной характеристикой четырехканального Q-ПУБФ на основе умножителя кватернионов из конвейера 2D-CONST-CORDIC-модулей осуществлялось для следующих параметров банка фильтров: эффективность субполосного кодирования $CG = 8,41$ дБ; ослабление в полосе непропускания более 21дБ; уровень постоянной составляющей не менее -331 дБ, т. е. отсутствует эффект DC-leakage; 2D-CORDIC-арифметика 16-разрядная, погрешность сходимости менее 10^{-5} .

Анализ результатов моделирования показывает, что амплитудно-частотная характеристика системы анализа-синтеза Q-ПУБФ представляет собой прямолинейную функцию, т. е. анализ-синтез Q-ПУБФ со встроенными 2D-CONST-CORDIC-модулями обладает свойством совершенной реконструкции входных данных и вносит только задержку. Например, у реализации решетчатой структуры Q-ПУБФ на основе планарных оборотов Гивенса амплитудно-частотная характеристика системы анализа-синтеза соответствует нелинейной функции с максимальной неравномерностью $\pm 0,2$ дБ. Таким образом, Q-ПУБФ на основе 2D-CONST-CORDIC-схем представляет собой преобразователь «целое к целому» и является системой без потерь.

Заключение

В работе сформулирована задача оптимизации схемных решений CORDIC-алгоритма для фиксированного угла поворота как для 2D-пространства, так и многомерного (4D) евклидова вращения. Результатом решения задачи является оптимизированное множество микровращений для фиксированного и известного углов поворота. При этом итерации микровращения и масштабирования выполняются последовательно друг за другом на одном аппаратном модуле. Эффективные CORDIC-схемы получаются благодаря тому, что параметры сдвига представляются нелинейной функцией числа сдвигов входных операндов операции микровращения в отличие от стандартного CORDIC-алгоритма, где параметры сдвига являются последовательностью однобитных сдвигов. В предложенной структуре параметры схемы подбираются каждый раз под соответствующий кватернион-константу, на который выполняется умножение, т. е. для каждого постоянного сомножителя-кватерниона константы получается разное количество сдвигов и разные масштабные множители, а также свой набор операторов направления вращения, которые должны храниться в памяти. Это цена за оптимизацию итерационного процесса и общая проблема к подходам построения более гибких CORDIC-алгоритмов [14, 28, 30].

Список литературы

1. Кантор, И.Л. Гиперкомплексные числа / И.Л. Кантор, А.С. Солодовников. – М. : Наука, 1973. – 145 с.
2. Parfieniuk, M. Quaternionic building block for paraunitary filter banks / M. Parfieniuk, A. Petrovsky // Proc. 12th European Signal Processing Conf. (EUSIPCO). – Vienna, Austria, 2004. – P. 1237–1240.
3. Choukroun, D. Novel quaternion kalman filter / D. Choukroun, I. Bar-Itzhack, Y. Oshman // IEEE Trans. Aerosp. Electron. Syst. – 2006. – Vol. 42, № 1. – P. 174–190.
4. Miron, S. Quaternion-music for vector-sensor array processing / S. Miron, N. Le Bihan, J. Mars // IEEE Trans. Signal Process. – 2006. – Vol. 54, № 4. – P. 1218–1229.
5. Hsiao, S.F. Parallel singular value decomposition of complex matrices using multidimensional CORDIC algorithms / S.F. Hsiao, J.M. Delosme // IEEE Trans. Signal Process. – 1996. – Vol. 44, № 3. – P. 685–697.

6. Sercov, V.V. Digital hypercomplex all-pass filters: A novel filters bank building block / V.V. Sercov, A.A. Petrovsky, D.V. Lushtyk // Proc. 6th Int. Workshop on Systems, Signals and Image Proc. (IWSSIP). – Bratislava, Slovakia, 1999. – P. 181–184.
7. Парфенюк, М. Параунитарные банки фильтров на основе алгебры кватернионов: теория и применение / М. Парфенюк, А.А. Петровский // Цифровая обработка сигналов. – 2008. – № 1. – С. 22–36.
8. Parfieniuk, M. Inherently lossless structures for eight- and six-channel linear-phase paraunitary filter banks based on quaternion multipliers / M. Parfieniuk, A. Petrovsky // Signal Process. – 2010. – Vol. 90. – P. 1755–1767.
9. Alexiadis, D. Estimation of motions in color image sequences using hypercomplex Fourier transforms / D. Alexiadis, G. Sergiadis // IEEE Trans. Image Process. – 2009. – Vol. 18, № 1. – P. 168–187.
10. Karney, C. Quaternions in molecular modeling / C. Karney // J. Molecular Graphics and Modelling. – 2007. – Vol. 25, № 5. – P. 595–604.
11. Vaidyanathan, P.P. Multirate Systems and Filter Banks / P.P. Vaidyanathan. – Englewood Cliffs : Prentice-Hall, 1993. – 911 p.
12. Howell, T.D. The complexity of the quaternion product: Tech. Rep. TR 75-245 / T.D. Howell, J.C. Lafon [Электронный ресурс]. – 1975. – Mode of access : <http://www.theworld.com/~sweetser/quaternions/ps/cornellcstr75-245.pdf>. – Date of access : 05.10.2015.
13. Parfieniuk, M. Quaternion multiplier inspired by the lifting implementation of plane rotations / M. Parfieniuk, A. Petrovsky // IEEE Trans. Circuits Syst. I. – 2010. – Vol. 57, № 10. – P. 2708–2717.
14. Parfieniuk, M. Rapid Prototyping of Quaternion Multiplier: From Matrix Notation to FPGA-Based Circuits / M. Parfieniuk, N. Petrovsky, A. Petrovsky // Rapid Prototyping Technology: Principles and Functional Requirements. – Vienna : InTech, 2011. – P. 227–246.
15. 50 years of CORDIC: Algorithms, architectures, and applications / P. Meher [et al] // IEEE Trans. Circuits Syst. I. – 2009. – Vol. 56, № 9. – P. 1893–1907.
16. Volder, J.E. The CORDIC trigonometric computing technique / J. E. Volder // IRE Trans. Electron. Comput. – 1959. – Vol. 8. – P. 330–334.
17. Walther, J.S. A unified algorithm for elementary functions / J.S. Walther // Proceedings of the Spring Joint Computer Conf. AFIPS '71 (Spring). – N. Y. : ACM, 1971. – P. 379–385.
18. Hsiao, S.-F. Householder CORDIC algorithms / S.-F. Hsiao, J.-M. Delosme // IEEE Trans. Comput. – 1995. – Vol. 44, № 8. – P. 990–1001.
19. Гантмахер, Ф.Р. Теория матриц / Ф.Р. Гантмахер; 5-е изд. – М. : Физматлит, 2004. – 560 с.
20. Hsiao, S.-F. Redundant constant-factor implementation of multi-dimensional CORDIC and its application to complex SVD / S.-F. Hsiao, C.-Y. Lau, J.-M. Delosme // IEEE Trans. VLSI Syst. – 2000. – Vol. 25, № 2. – P. 155–166.
21. Петровский, Н.А. Многомерный CORDIC алгоритм кватернионов с «разреженными» итерациями / Н.А. Петровский, М. Парфенюк // Труды 15-й Междунар. конф. «Цифровая обработка сигналов и ее применение» (DSPA'2013). – Т. 2. – М., 2013. – С. 206–210.
22. Петровский, Н.А. 4D-CORDIC арифметика для процессора параунитарного банка фильтров на основе алгебры кватернионов / Н.А. Петровский, М. Парфенюк, А.А. Петровский // Труды 40-й Междунар. науч. конф. «Вопросы оптимизации вычислений» (ВОВ-XL). – Ялта, 2013. – С. 214–215.
23. Benes, V.E. Mathematical theory of connecting networks and telephone traffic / V.E. Benes. – N. Y. : Academic Press, 1965. – 319 p.
24. Suzuki, T. Generalized block-lifting factorization of M-channel biorthogonal filter banks for lossy-to-lossless image coding / T. Suzuki, M. Ikehara, T. Nguyen // IEEE Trans. Image Process. – 2012. – Vol. 21, № 7. – P. 3220–3228.
25. Wavelet transforms that map integers to integers / A.R. Calderbank [et. al.] // Appl. Comput. Harmon. Anal. – 1998. – Vol. 5, № 3. – P. 332–369.

26. Petrovsky, N. Pipelined embedded processor of quaternionic m-band wavelets for image multiresolution analysis / N. Petrovsky, M. Parfieniuk, A. Petrovsky // 2nd Mediterranean Conf. on Embedded Computing (MECO). – Budva, Montenegro, 2013. – P. 196–199.
27. Оранский, А.М. Аппаратные методы в цифровой вычислительной технике / А.М. Оранский. – Минск : ЕСУ, 1977. – 208 с.
28. Petrovsky, N.A. The CORDIC-inside-lifting architecture for constant-coefficient hardware quaternion multipliers / N.A. Petrovsky, M. Parfieniuk // Proc. Intern. Conf. on Signals and Electronic Systems (ICSSES). – Wroclaw, 2012. – P. 1–6.
29. Петровский, Н.А. Рекурсивный процессор умножителя кватернионов со структурной CORDIC-лестничной параметризацией / Н.А. Петровский, А.В. Станкевич // Труды Междунар. науч. конф. «Информационные технологии и системы» (ИТС'2013). – Минск : БГУИР, 2013. – С. 196–197.
30. Meher, P. CORDIC designs for fixed angle of rotation / P. Meher, S. Park // IEEE Trans. VLSI Syst. – 2013. – Vol. 21, № 2. – P. 217–228.

Поступила 05.10.2015

*Белорусский государственный университет
информатики и радиоэлектроники,
Минск, П. Бровка, 6
e-mail: nick@petrovsky.eu,
stankevich@bsuir.by,
palex@bsuir.by*

N.A. Petrovsky, A.V. Stankevich, A.A. Petrovsky

CORDIC TECHNIQUES FOR FIXED ANGLE OF ROTATION IN MULTIPLYING OPERATION OF QUATERNIONS

The article contains a number of solutions for the key element of paraunitary filter banks based on quaternionic algebra (Q-PUBF) – the multiplier of quaternions with usage of CORDIC (Coordinate Rotation Digital Computer) techniques for the fixed angle of rotation where, unlike known solutions, 4D rotation control parameters are represented by nonlinear function of shifts number of input operands of the microrotation operation. Suggested approach of the multiplier designing on a quaternion-constant allows reaching the maximum performance of the multiplier scheme with low use of resources, for example, of FPGA.