基于 N 点 DFT（FFT）计算 2N 点 MDCT 的方法

起草：季文骢，版本：4

第一部分：概述

2N 点实序列的 $x_w(n)$ 的 2N 点 MDCT（Modified Discrete Cosine Transform）的定义式如下：

$$X(k)=c\cdot\sum_{n=0}^{2N-1}x_w(n)\cos\left(\frac{\pi}{N}\cdot\left(n+\frac{1}{2}+\frac{N}{2}\right)\cdot\left(k+\frac{1}{2}\right)\right)\ (0\le k<N)$$

其中 $x_w(n)=x(n)w(n)$，$x(n)$ 序列为时域采样序列（输入序列），$w(n)$ 序列为窗序列，$X(k)$ 为输出序列，$x(n)$、$w(n)$、$x_w(n)$、$X(k)$ 序列均为实序列，c 是任意的实常数，$0\le n<2N$。

在实践中一般采用基于 2N 点 DFT（FFT）的算法来计算 2N 点 MDCT[1]。下面给出一种时间复杂度更低的基于 N 点 DFT（FFT）的计算 MDCT 输出序列 $X(k)$ 的方法及其正确性证明。

另外，文中所有的定义和推导过程都默认各种序列的自变量是整数，因此对于序列的自变量，本文仅给出其范围而不再强调其为整数这一事实。

第二部分：算法流程及正确性证明

将 $X(k)$ 展开：

$$X(k)=\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)\cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)+\frac{\pi}{2}\left(k+\frac{1}{2}\right)\right)$$

定义 $\alpha(n,k)=\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)$，$\varphi(k)=\frac{\pi}{2}\left(k+\frac{1}{2}\right)$（$0\le n<2N$，$0\le k<N$），那么：

$$X(k)=\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)\cos\left(\alpha(n,k)+\varphi(k)\right)$$

$$=\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)\left(\cos\left(\alpha(n,k)\right)\cos\left(\varphi(k)\right)-\sin\left(\alpha(n,k)\right)\sin\left(\varphi(k)\right)\right)$$

$$=\cos\left(\varphi(k)\right)\overbrace{\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)\cos\left(\alpha(n,k)\right)}^{T_1(k)}-\sin\left(\varphi(k)\right)\overbrace{\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)\sin\left(\alpha(n,k)\right)}^{T_2(k)}$$

定义序列 $T_1(k)$、$T_2(k)$（$0\le k<N$）：

$$T_1(k)=\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)\cos\left(\alpha(n,k)\right)$$

$$T_2(k)=\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)\sin\left(\alpha(n,k)\right)$$

显然有 $X(k)=\cos\left(\varphi(k)\right)T_1(k)-\sin\left(\varphi(k)\right)T_2(k)$，继续展开 $T_1(k)$、$T_2(k)$：

$$e^{j\alpha(n,k)} = \cos\big(\alpha(n,k)\big) + j \cdot \sin\big(\alpha(n,k)\big)$$
$$e^{-j\alpha(n,k)} = \cos\big(\alpha(n,k)\big) - j \cdot \sin\big(\alpha(n,k)\big)$$

$$T_1(k) = \sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) \frac{e^{j\alpha(n,k)} + e^{-j\alpha(n,k)}}{2}$$

$$= \frac{1}{2}\left(\overbrace{\sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) e^{j\alpha(n,k)}}^{A^*(k)} + \overbrace{\sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) e^{-j\alpha(n,k)}}^{A(k)} \right)$$

$$T_2(k) = \sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) \frac{e^{j\alpha(n,k)} - e^{-j\alpha(n,k)}}{2j}$$

$$= \frac{1}{2j}\left(\overbrace{\sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) e^{j\alpha(n,k)}}^{A^*(k)} - \overbrace{\sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) e^{-j\alpha(n,k)}}^{A(k)} \right)$$

定义序列 $A(k)$ 及其共轭序列 $A^*(k)$（$0 \le k < N$）：

$$A(k) = \sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) e^{-j\alpha(n,k)}$$

$$A^*(k) = \big(A(k)\big)^* = \sum_{n=0}^{2N-1} \big(c \cdot x_w(n)\big) e^{j\alpha(n,k)}$$

于是可用 $T_1(k)$、$T_2(k)$ 就可以用序列 $A(k)$ 及共轭序列 $A^*(k)$ 来表示：

$$T_1(k) = \frac{1}{2}\big(A^*(k) + A(k)\big)$$

$$T_2(k) = \frac{1}{2j}\big(A^*(k) - A(k)\big)$$

定义序列 $m(n)$ 及其共轭序列 $m^*(n)$（$0 \le n < N$）：

$$m(n) = \frac{1}{2}\big(c \cdot x_w(2n) + j \cdot c \cdot x_w(2n+1)\big) e^{-j\frac{\pi}{N}n}$$

$$m^*(n) = \big(m(n)\big)^* = \frac{1}{2}\big(c \cdot x_w(2n) - j \cdot c \cdot x_w(2n+1)\big) e^{j\frac{\pi}{N}n}$$

于是则有：

$$m(n) + m^*(n) e^{-j\frac{2\pi}{N}n} = c \cdot x_w(2n) \cdot e^{-j\frac{\pi}{N}n}$$

$$m(n) - m^*(n) e^{-j\frac{2\pi}{N}n} = j \cdot c \cdot x_w(2n+1) \cdot e^{-j\frac{\pi}{N}n}$$

对序列 $m(n)$ 进行 N 点 DFT，得到序列 $M(k)$ 及其共轭序列 $M^*(k)$（其中 $0 \le k < N$）：

$$M(k) = \sum_{n=0}^{N-1} m(n) e^{-j\frac{2\pi}{N}kn}$$

$$M^*(k) = \left(M(k)\right)^* = \sum_{n=0}^{N-1} m^*(n)e^{j\frac{2\pi}{N}kn}$$

将 $N-k-1$ 作为自变量代入 $M^*(k)$：

$$M^*(N-k-1) = \sum_{n=0}^{N-1} m^*(n)e^{j\frac{2\pi}{N}(N-k-1)n} = \sum_{n=0}^{N-1}\left(m^*(n)e^{-j\frac{2\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn}$$

那么就有：

$$M(k) + M^*(N-k-1) = \sum_{n=0}^{N-1} m(n)e^{-j\frac{2\pi}{N}kn} + \sum_{n=0}^{N-1}\left(m^*(n)e^{-j\frac{2\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1}\left(m(n) + m^*(n)e^{-j\frac{2\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1}\left(c\cdot x_w(2n)\cdot e^{-j\frac{\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn}$$

$$M(k) - M^*(N-k-1) = \sum_{n=0}^{N-1} m(n)e^{-j\frac{2\pi}{N}kn} - \sum_{n=0}^{N-1}\left(m^*(n)e^{-j\frac{2\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1}\left(m(n) - m^*(n)e^{-j\frac{2\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn} = j\cdot\sum_{n=0}^{N-1}\left(c\cdot x_w(2n+1)\cdot e^{-j\frac{\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn}$$

继续对 $A(k)$ 进行展开（按自变量 k 的奇偶性进行分解）

$$A(k) = \sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\right)e^{-j\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)}$$

$$= e^{-j\frac{\pi}{2N}\left(k+\frac{1}{2}\right)}\cdot\sum_{n=0}^{2N-1}\left(c\cdot x_w(n)\cdot e^{-j\frac{\pi}{2N}n}\right)e^{-j\frac{\pi}{N}kn}$$

$$= e^{-j\frac{\pi}{2N}\left(k+\frac{1}{2}\right)}\left(A_{even}(k) + A_{odd}(k)\right)$$

$$A_{even}(k) = \sum_{n=0}^{N-1}\left(c\cdot x_w(2n)\cdot e^{-j\frac{\pi}{2N}(2n)}\right)e^{-j\frac{\pi}{N}k(2n)}$$

$$= \sum_{n=0}^{N-1}\left(c\cdot x_w(2n)\cdot e^{-j\frac{\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn}$$

$$= M(k) + M^*(N-k-1)$$

$$A_{odd}(k) = \sum_{n=0}^{N-1}\left(c\cdot x_w(2n+1)\cdot e^{-j\frac{\pi}{2N}(2n+1)}\right)e^{-j\frac{\pi}{N}k(2n+1)}$$

$$= \sum_{n=0}^{N-1}\left(c\cdot x_w(2n+1)\cdot e^{-j\frac{\pi}{N}n}e^{-j\frac{\pi}{2N}}\right)e^{-j\frac{2\pi}{N}kn}e^{-j\frac{\pi}{N}k}$$

$$= e^{-j\left(\frac{\pi}{2N}+\frac{\pi}{N}k\right)}\cdot\sum_{n=0}^{N-1}\left(c\cdot x_w(2n+1)\cdot e^{-j\frac{\pi}{N}n}\right)e^{-j\frac{2\pi}{N}kn}$$

$$= \frac{e^{-j\frac{\pi}{N}\left(k+\frac{1}{2}\right)}}{j}\Big(M(k) - M^*(N-k-1)\Big)$$

$$= e^{-j\frac{\pi}{N}\left(k+\frac{1}{2}+\frac{N}{2}\right)}\Big(M(k) - M^*(N-k-1)\Big)$$

到此为止，对于 $0 \le k < N$，所有的 $A(k)$ 都已被求出，由此便可推出 $T_1(k)$、$T_2(k)$ 序列并得到最终结果 $X(k)$：

$$T_1(k) = \frac{1}{2}\Big(A^*(k) + A(k)\Big) = \mathrm{Re}\{A(k)\}$$

$$T_2(k) = \frac{1}{2j}\Big(A^*(k) - A(k)\Big) = -\mathrm{Im}\{A(k)\}$$

$$X(k) = \cos\big(\varphi(k)\big)\,T_1(k) - \sin\big(\varphi(k)\big)\,T_2(k) = \cos\big(\varphi(k)\big)\,\mathrm{Re}\{A(k)\} + \sin\big(\varphi(k)\big)\,\mathrm{Im}\{A(k)\}$$

此外，考察序列 $m(n)$ 的一个性质：

$$m(n) = \frac{1}{2}\big(c \cdot x_w(2n) + j \cdot c \cdot x_w(2n+1)\big)e^{-j\frac{\pi}{N}n}$$

$$= x(2n) \cdot \overbrace{\frac{c}{2}w(2n)e^{-j\frac{\pi}{N}n}}^{\rho_{even}(n)} + x(2n+1) \cdot \overbrace{j\frac{c}{2}w(2n+1)e^{-j\frac{\pi}{N}n}}^{\rho_{odd}(n)}$$

$$= x(2n)\rho_{even}(n) + x(2n+1)\rho_{odd}(n)$$

$$\rho_{even}(n) = \frac{c}{2}w(2n)e^{-j\frac{\pi}{N}n}$$

$$\rho_{odd}(n) = j\frac{c}{2}w(2n+1)e^{-j\frac{\pi}{N}n} = \frac{c}{2}w(2n+1)e^{-j\left(\frac{\pi}{N}n-\frac{\pi}{2}\right)}$$

如此便可得结论：窗序列 $w(n)$ 和实常数 c 是恒定不变的（不论时域采样序列 $x(n)$ 为何值），那么只需预先计算出 $0 \le n < N$ 时所有的 $\rho_{even}(n)$ 和 $\rho_{odd}(n)$ 的值，在计算 $m(n)$ 的值的时候就不再需要考虑 $w(2n)$、$w(2n+1)$ 以及 c 的值，这样每处理一组时域采样序列 $x(n)$，都能够节省 N 次对 c 的乘法运算和 $2N$ 次对 $w(n)$ 的乘法运算。

第三部分：总结

最后总结基于 N 点 DFT 计算序列 $X(k)$ 的值的流程（DFT 过程可采用 FFT 来实现）：

第一步：构造序列 $m(n)$（$0 \le n < N$）；

第二步：计算 $m(n)$ 的 N 点 DFT，得到序列 $M(k)$（$0 \le k < N$）；

第三步：计算 $A_{even}(k)$ 和 $A_{odd}(k)$ 的值，构造序列 $A(k)$（$0 \le k < N$）；

第四步：利用序列 $A(k)$ 构造序列 $X(k)$（$0 \le k < N$）。

算法伪代码如下（输入为时域采样序列 x[]、窗序列 w[]、实常数 c，输出为序列 X[]）：

```
//  ρ_even[0...N - 1], ρ_odd[0...N - 1]:
for (n = 0; n < N; ++n) {
    rho_even[n] = 0.5c * w[2n] * pow(E, -1j * PI * n / N);
    rho_odd[n] = 0.5c * w[2n + 1] * pow(E, -1j * (PI * n / N - PI / 2));
}

//  m[0...N-1]:
for (n = 0; n < N; ++n) {
    m[n] = x[2n] * rho_even[n] + x[2n + 1] * rho_odd[n];
}

//  M[0...N-1]:
M = DFT(m)

//  A[0...N - 1]:
```

```
for (k = 0; k < N; ++k) {
   A_even = M[k] + M[N - k - 1].conjugate();
   A_odd = (M[k] - M[N - k - 1].conjugate()) * pow(E, -1j * PI * (k + 0.5 + 0.5N) / N);
   A[k] = pow(E, -1j * PI * (k + 0.5) / 2N) * (A_even + A_odd)
}

// X[0...N - 1]:
for (k = 0; k < N; ++k) {
   phi = (k + 0.5) * PI / 2;
   X[k] = cos(phi) * re(A[k]) + sin(phi) * im(A[k]);
}
```

不难注意到 X[k] 的值仅依赖于 A[k] 的值，因此 A[] 实际上是不必需的，完全可以把上面的伪代码的后半部分简化（这样也省去了一次长度为 N 的遍历）：

```
...
// X[0...N - 1]:
for (k = 0; k < N; ++k) {
   phi = (k + 0.5) * PI / 2;
   A_even = M[k] + M[N - k - 1].conjugate();
   A_odd = (M[k] - M[N - k - 1].conjugate()) * pow(E, -1j * PI * (k + 0.5 + 0.5N) / N);
   u = pow(E, -1j * PI * (k + 0.5) / 2N) * (A_even + A_odd)
   X[k] = cos(phi) * re(u) + sin(phi) * im(u);
}
```

附录 1：DFT/IDFT 的定义

由于在不同的文献对于 DFT/IDFT 的定义可能有所不同，为了避免混淆，特在此列出在本文中所用的 DFT/IDFT 定义。

N 点复序列 $f(n)$（$0 \le n < N$）的 N 点 DFT 的定义为：

$$F(k) = \mathrm{DFT}\{f(n)\}(k) = \sum_{n=0}^{N-1} f(n)e^{-j\frac{2\pi}{N}kn}$$

N 点复序列 $F(k)$（$0 \le k < N$）的 N 点 IDFT 的定义为：

$$f(n) = \mathrm{IDFT}\{F(k)\}(n) = \frac{1}{N}\sum_{k=0}^{N-1} F(k)e^{j\frac{2\pi}{N}kn}$$

附录 2：与专利 WO2001033411 的关系

本文所述算法的总体思路借鉴了专利 WO2001033411[2]，但算法主要流程的计算式与该专利的权利要求（Claim）部分并不相同，因此使用本文所述的算法理论上不会构成对该专利的应用。

参考文献

编号	引用文献	DOI/URL
1	Bosi, M. and Goldberg, R.E. 2003. Introduction to Digital Audio Coding and Standards. Springer US.	10.1007/978-1-4615-0327-9
2	WO/2001/033411, Fast Modified Discrete Cosine Transform Method	WO2001033411