FileName: ComputingDFT.doc

COMPUTING THE DFT

In this section, I will explain the calculation method of the program DFT V0.1, before explore the program code, let have a general view of DFT calculation first.

General View of computing DFT

The most obvious approach to computing the DFT and IDFT is to implement the below equation directly. The method is straightforward, but relatively low. Expressing the imaginary exponentials in term of sinusoidal and cosines, we have DFT.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2pkn/N}$$

$$= \sum_{n=0}^{N-1} x[n] (\cos 2pkn/N - j \sin 2pkn/N)$$

$$R(X[k]) = \sum_{n=0}^{N-1} x[n] \cos 2pkn/N$$

$$I(X[k]) = -\sum_{n=0}^{N-1} x[n] \sin 2pkn/N$$

$$|X[k]| = \sqrt{(R(X[k]))^2 + (I(X[k]))^2}$$

Similarly, we have IDFT

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2pkn/N}$$

$$R(x[n]) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cos 2pkn/N$$

$$I(x[n]) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \sin 2pkn/N$$

$$|x[n]| = \sqrt{(R(x[n]))^2 + (I(x[n]))^2}$$

Direct implement the equation above is costly in terms of the number of multiplications involved. There require N^2 complex multiplication, and each complex multiplication require 4 real multiplication and 2 real addition as (a+jb)(c+jd)=(ac+ bd) + j(ad+bd).

Many of the multiplication turn out to be redundant, because of the exponentials specified in the DFT and IDFT equation are calculated many times as k and n vary. This is particularly true with lengthy transforms. High efficient FFT algorithms are available for reducing the redundancy.

Program code

In the program I use the straightforward way to implement the calculation because of the simplicity. The function below provides the DFT calculation. The input parameter is x[n] and number of sample. The program is divided into two parts, the first perform DTF and get the N samples F[k], the second part copy the N samples along the frequency axis. The characteristic is because of the sampling. When calculation is complete return the DFT values to the caller.

```
Public Function DFT(x() As Double, NoOfSample As Integer) As Variant

    Dim n As Integer
    Dim ans As Single           'Temporary answer
    Dim T As Double
    Dim pi As Double            '3.14..
    Dim i As Integer
```

```
 Dim k As Integer

 Dim Fr(1000) As Double         'real part of F
 Dim Fi(1000) As Double         'image part of F
 Dim F(1000) As Double          'magnitude of F
 Dim Φ(1000) As Double          'phase of F


 pi = 4 * Atn(1)

'******Perform DTF transformation******
For k = 0 To NoOfSample - 1

    'Real Part
    ans = 0
    For n = 0 To NoOfSample - 1
        ans = ans + x(n) * Math.Cos(2 * pi * k * n / NoOfSample)
    Next n
    Fr(k) = ans

    'Image Part
    ans = 0
    For n = 0 To NoOfSample - 1
        ans = ans + x(n) * Math.Sin(2 * pi * k * n / NoOfSample)
    Next n
    Fi(k) = ans

  'magnitude
  F(k) = (Fr(k) ^ 2 + Fi(k) ^ 2) ^ 0.5
  If Fi(k) = 0 Then
    Φ(k) = pi / 2
  Else
    Φ(k) = Math.Atn(Fr(k) / Fi(k))  'inverse trigonometric function of Tan
  End If

Next k


'******Fill the repetitive samples in Frequency Domain******
 'fill N samples signal in the time domain and form a peridic signal
 i = 0
 For k = NoOfSample To 1000
   F(k) = F(i)
   i = i + 1
   If i = NoOfSample Then i = 0
 Next k

 DFT = F()       '* return DFT calculation result to caller

End Function
```