Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 26
Differential / Integral Operations by FFT&IFFT

Verigy Japan
June 2010

Preface to the Series

ADC and DAC are the most typical mixed signal devices. In mixed signal testing, analog stimulus signal is generated by an arbitrary waveform generator (AWG) which employs a D/A converter inside, and analog signal is measured by a digitizer or a sampler which employs an A/D converter inside. The stimulus signal is created with mathematical method, and the measured signal is processed with mathematical method, extracting various parameters. It is based on digital signal processing (DSP) so that our test methodologies are often called DSP-based testing.

Test/application engineers in the mixed signal field should have thorough knowledge about DSP-based testing. FFT (Fast Fourier Transform) is the most powerful tool here. This corner will deliver a series of fundamental knowledge of DSP-based testing, especially FFT and its related topics. It will help test/application engineers comprehend what the DSP-based testing is and assorted techniques.

Editor's Note

For other articles in this series, please visit the Verigy web site at www.verigy.com/go/gosemi.

Preface

In DSP-based testing, differential and integral mathematical operations are necessary sometimes. You may encounter differential operations more frequently than integral operations. For example, group delay of a filter is defined as a differential of phase rotation. Differential operation is usually substituted by simple linear gradient calculation of two adjacent data and it would work well in many cases. On the other hand, integral operations are not frequently performed in our testing activities. FM waveform generation may be the only one example. PM waveform is easily programmable by relatively simple equations; however, FM waveform is more complex than PM. FM can be realized by utilizing PM with integral operation. The detailed method of FM generation with integral PM will be discussed next month. So this month differential and integral operations are discussed as background knowledge for the next month article.

Signal Expressed by Fourier Series

A signal $x(t)$ in our DSP-based testing is a band-limited periodic function, which can be expressed as a Fourier series as follows.

$$x(t) = \sum_{k=0}^{N-1} \left(C_k e^{jk\omega t} \right) \tag{1}$$

where C_k is frequency spectrum component, ω is angular frequency, t is time, and j is the imaginary unit. The frequency component C_k can be derived by DFT(or FFT) of the time domain data. If the spectrum components C_k are known, the signal $x(t)$ can be reconstructed by performing Inverse DFT(IFFT).

Differential

When we need a differential dx/dt at sampling points, we usually calculate $\triangle x/\triangle t=(x_{n+1}-x_n)/(t_{n+1}-t_n)$ as a substitute of dx/dt by looking at every two adjacent points (t_n, x_n) and (t_{n+1}, x_{n+1}). Even if $\triangle x/\triangle t$ is approximation, it is practical and good enough in many cases. There is an inconvenience in this method. The derived value would represent the value at the mid location of the two points. It is not the value right at the sampling point.

The differential of signal $x(t)$ in Equation (1) can be derived as follows.

$$\frac{d}{dt} x(t) = \frac{d}{dt} \sum_{k=0}^{N-1} \left(C_k e^{jk\omega t} \right) = \sum_{k=0}^{N-1} \left(C_k \cdot jk\omega e^{jk\omega t} \right) = \sum_{k=0}^{N-1} \left(jk\omega C_k \cdot e^{jk\omega t} \right) \tag{2}$$

Equation (2) means that the differential of $x(t)$ can be calculated by modifying each one of the spectrum components C_k multiplied by $jk\omega$ one by one, and performing IDFT (or IFFT) to the modified spectrum $jk\omega C_k$. The processing steps are listed as the diagram as follows.

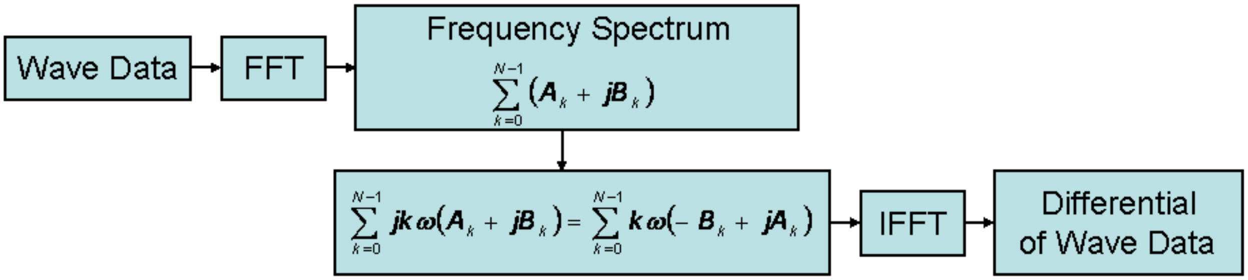

Figure 1: Differential Operation Process by FFT&IFFT

As Figure 1 shows, when multiplying $jk\omega$ to C_k respectively, firstly the real and imaginary components of $C_k(=A_k+jB_k)$ are exchanged as $-B_k+jA_k$, and secondly $k\omega$ is multiplied.

List 1 is an example coding. Firstly the original waveform "dWave[]" is converted into the complex number form at Line 13 for full-plane spectrum. FFT is performed to the time domain data at Line 14. Lines 18 through 22 show the core operation of differentiation illustrated in Figure 1. IFFT is performed at Line 26 for reconstructing the differential result in the time domain.

```
01:  INT           i,Ndata,Nsp;
02:  DOUBLE        dFsmp,dFresln,dkw;
03:  ARRAY_D       dWave,dDifferential;
04:  ARRAY_COMPLEX CSp,CSpj,CWave;
05:
06:  dFsmp=...;                              // Sampling Rate
07:  Ndata=...;                              // # of Data
08:  dFresln=dFsmp/Ndata;                    // Frequency Resolution
09:
10:  dWave.resize(Ndata);
11:  dWave[]=...;                            // Original Waveform
12:
13:  DSP_CONV_D_C(dWave,CWave,1.0,0.0);      // Double → Complex
14:  DSP_FFT(CWave,CSp,RECT);                // Frequency Domain
15:  Nsp=Ndata/2;
16:  CSpj.resize(Ndata);
17:  for (i=0;i<Nsp;i++) {
18:     dkw=2.0*M_PI*dFresln*i;              // 2*pi*f*i
19:     CSpj[i].real()=-CSp[i].imag()*dkw;   // Differential
20:     CSpj[i].imag()= CSp[i].real()*dkw;
21:     CSpj[Ndgt-i].real()= CSpj[i].real(); // Complex Conjugate
22:     CSpj[Ndgt-i].imag()=-CSpj[i].imag();
23:  }
24:  CSpj[Nsp]=CZero();                      // Complex Zero
25:
26:  DSP_IFFT(CSpj,CWave);                   // Time Domain
27:  dDifferential.resize(Ndata);            // Differential Container
28:  dDifferential=CWave.getReal();
29:
```

List 1: Example Code for Differential Operation

This program is tested by the curve of Equation (3), which is illustrated in Figure 2(a). The parameters are specified in the graph area. The number of signal cycles should be a whole number. The curve is constructed by 1-cycle sine, 2-cycle sine and 3-cycle sine in the unit test period (UTP).

$$v(t) = \sin(\omega t + \theta_1) + \sin(2\omega t + \theta_2) + \sin(3\omega t + \theta_3) \tag{3}$$

The differential of Equation (3) can mathematically be resolved as follows.

$$\frac{d}{dt}v(t) = \omega \cdot \cos(\omega t + \theta_1) + 2\omega \cdot \cos(2\omega t + \theta_2) + 3\omega \cdot \cos(3\omega t + \theta_3) \tag{4}$$

So the theoretical curve can be plotted as the blue line in Figure 2(b). The yellow dots in Figure 2(b) are derived by the conventional $\triangle x / \triangle t$ method. List 1 code based on the FFT&IFFT method illustrated in Figure 1 delivers the red dots in Figure 2(b). Figure 2(c) is the close-up view of a part of (b). Yellow dots are slightly off the blue line, while entire red dots are right on the blue line so that FFT&IFFT method derives more accurate results than conventional method in this case.

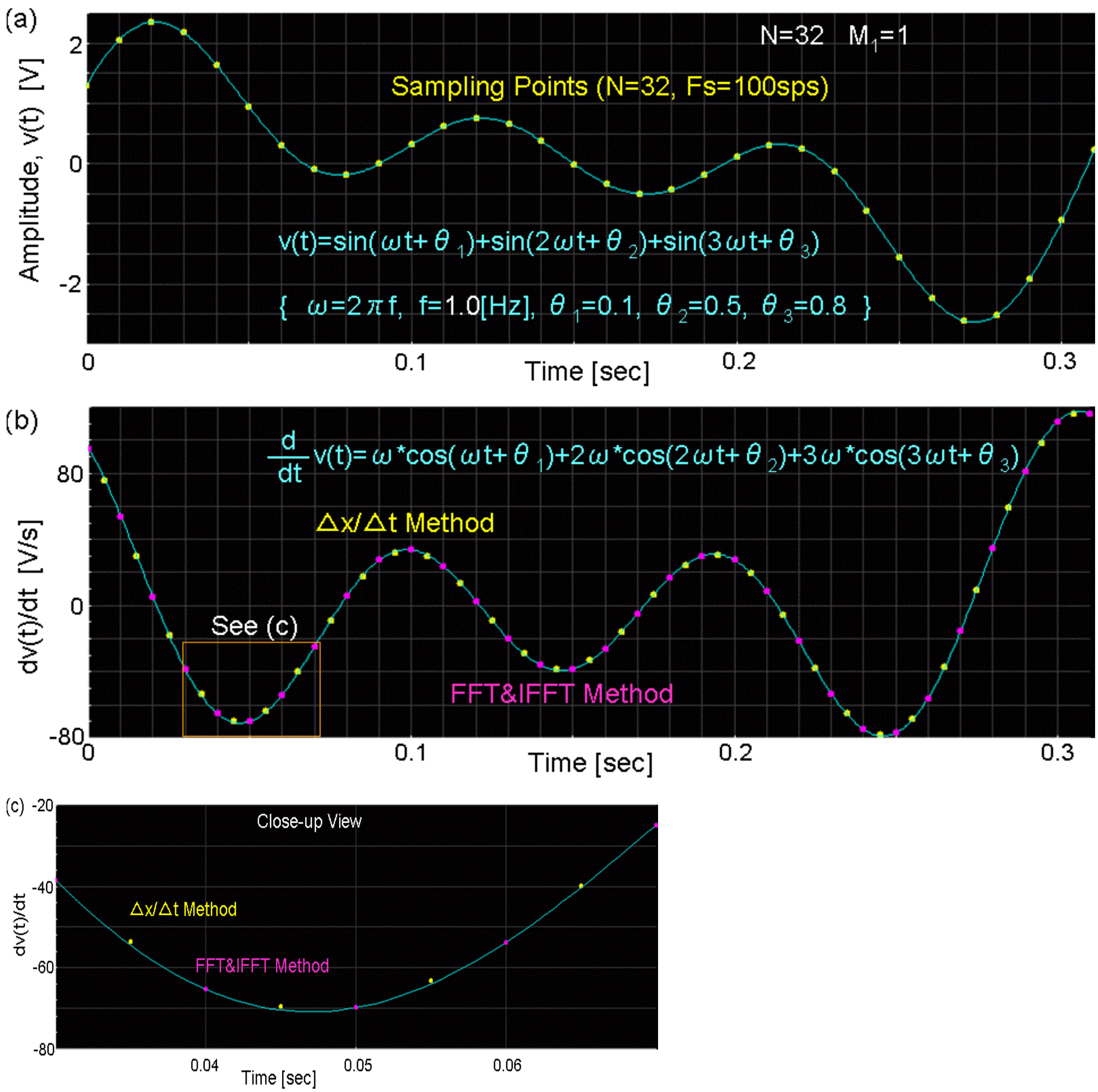

Figure 2: Differential Result under Good Integer Number Condition (M_1=1)

Figure 3 illustrates the fractional number cycles situation as $M_1=1.1$. The UTP contains 1.1 cycles, 2.2 cycles and 3.3 cycles of sine waveforms. Once the coherency is lost, the FFT&IFFT method provides miserable result as the red dots in Figure 3(b). It looks acceptable around the center but the areas near the both ends are extremely off the track. On the contrary, the conventional method gives better performance than FFT&IFFT method near the both ends. Generally speaking $\triangle x/\triangle t$ method seems good enough for many cases.

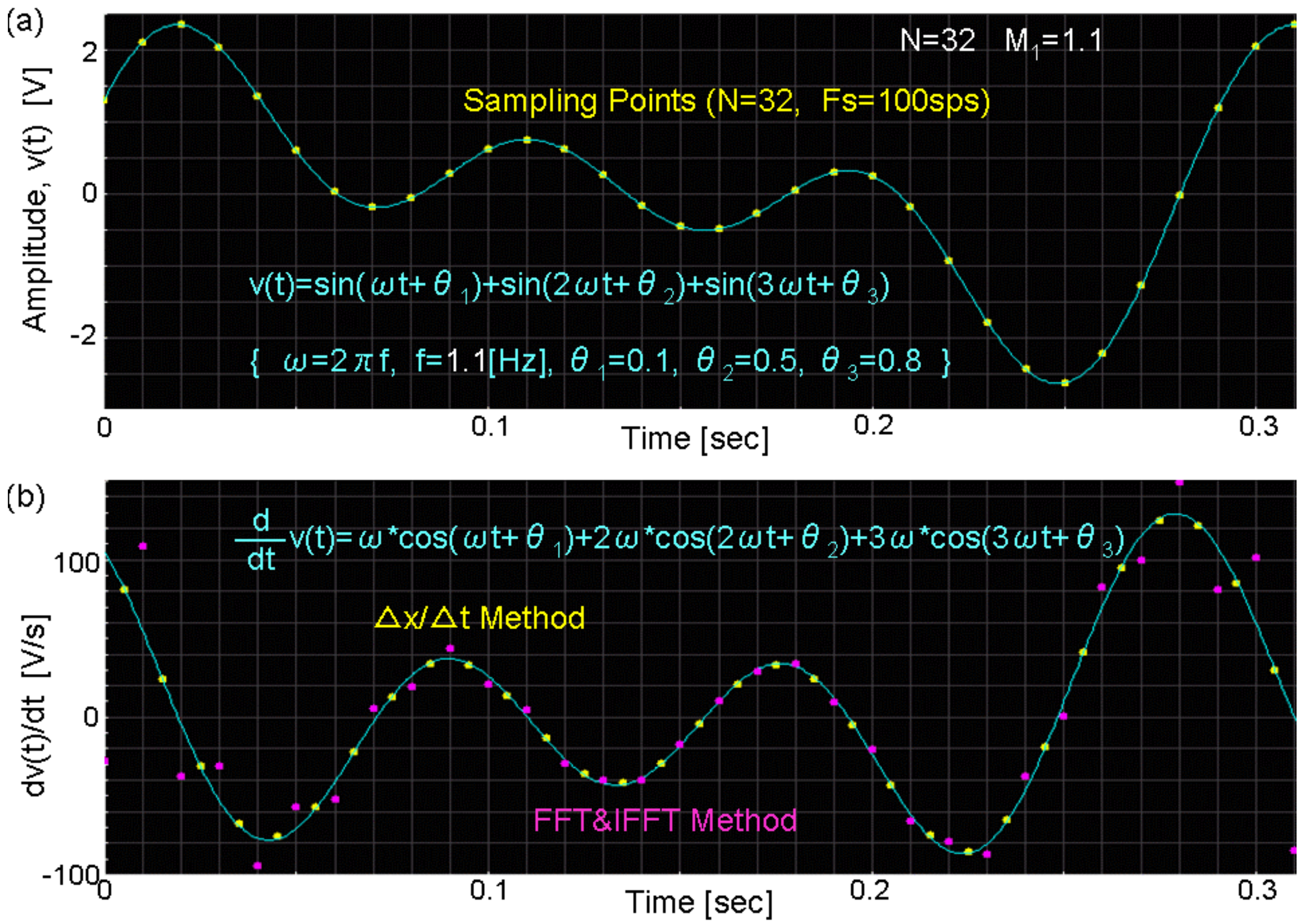

Figure 3: Differential Result under Bad Fractional Condition ($M_1=1.1$)

Integral

The integral of signal $x(t)$ is described as follows.

$$\int x(t)dt = \int \sum_{k=0}^{N-1} \left(C_k e^{jk\omega t} \right) dt = \sum_{k=0}^{N-1} \left(C_k \cdot \frac{1}{jk\omega} e^{jk\omega t} \right) = \sum_{k=0}^{N-1} \left(\frac{-j}{k\omega} C_k \cdot e^{jk\omega t} \right) \qquad (5)$$

Equation (5) means that the integral of $x(t)$ can be calculated by multiplying $-j/k\omega$ to the frequency spectrum components C_k and performing IDFT (or IFFT) to the modified spectrum. The processing steps are illustrated in Figure 4.

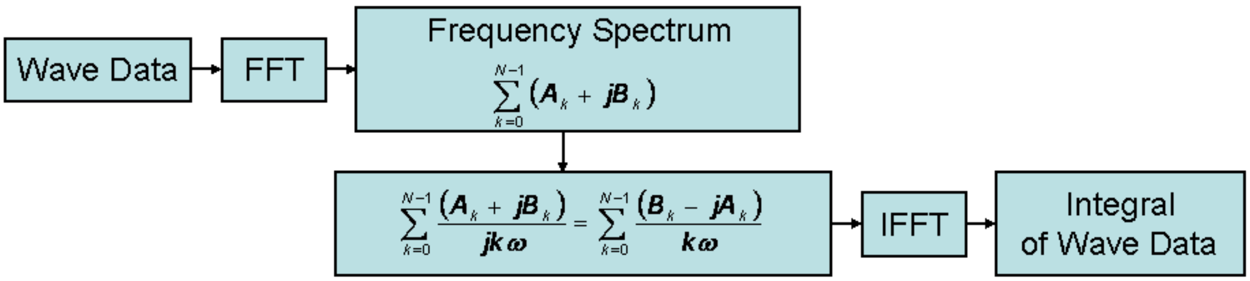

Figure 4: Integral Operation Process by FFT&IFFT

As Figure 4 shows, firstly the real and imaginary components of $C_k(=A_k+jB_k)$ are exchanged as B_k-jA_k, and secondly $1/k\omega$ is multiplied.

List 2 is an example coding. Firstly the original waveform "dWave[]" is converted into the complex number form at Line 13. FFT is performed to the time domain data at Line 14. Lines 19 through 23 show the core operation of integral illustrated in Figure 4. IFFT is performed at Line 27 for reconstructing the integral result in the time domain.

```
12:
13:  DSP_CONV_D_C(dWave,CWave,1.0,0.0);        // Double → Complex
14:  DSP_FFT(CWave,CSp,RECT);                  // Frequency Domain
15:  Nsp=Ndata/2;
16:  CSpj.resize(Ndata);
17:  CSpj[0]=CSp[0];                           // DC Component
18:  for (i=1;i<Nsp;i++) {
19:     dkw=2.0*M_PI*dFresln*i;                // 2*pi*f*i
20:     CSpj[i].real()= CSp[i].imag()/dkw;     // Integral
21:     CSpj[i].imag()=-CSp[i].real()/dkw;
22:     CSpj[Ndgt-i].real()= CSpj[i].real();   // Complex Conjugate
23:     CSpj[Ndgt-i].imag()=-CSpj[i].imag();
24:  }
25:  CSpj[Nsp]=CZero();                        // Complex Zero
26:
27:  DSP_IFFT(CSpj,CWave);                     // Time Domain
28:  dIntegral.resize(Ndata);
29:  dIntegral=CWave.getReal();
30:
```

List 2: Example Source Code for Integral Operation

The program is tested with the same curve in Equation (3). The integral of Equation (3) can mathematically be resolved as follows.

$$\int v(t)dt = -\frac{1}{\omega}\cdot\cos(\omega t+\theta_1) - \frac{1}{2\omega}\cdot\cos(2\omega t+\theta_2) - \frac{1}{3\omega}\cdot\cos(3\omega t+\theta_3) + C \qquad (6)$$

The original curve is shown in Figure 5(a) which is identical to Figure 2(a). The theoretical integral curve can be plotted as the blue line in Figure 5(b). The constant *C* is adjusted for the curve to begin from 0. The yellow dots in Figure 5(b) are plotted by accumulating each one of the sampling points in Figure 5(a) with multiplying the sampling period. It is a conventional integral approximation. The code in List 2 based on the FFT&IFFT method illustrated in Figure 4 delivers the red dots in Figure 5(b). As you can see, since the red dots are right on the blue curve, FFT&IFFT method can derive really accurate values as well as differential operation. The conventional approximation is not so accurate.

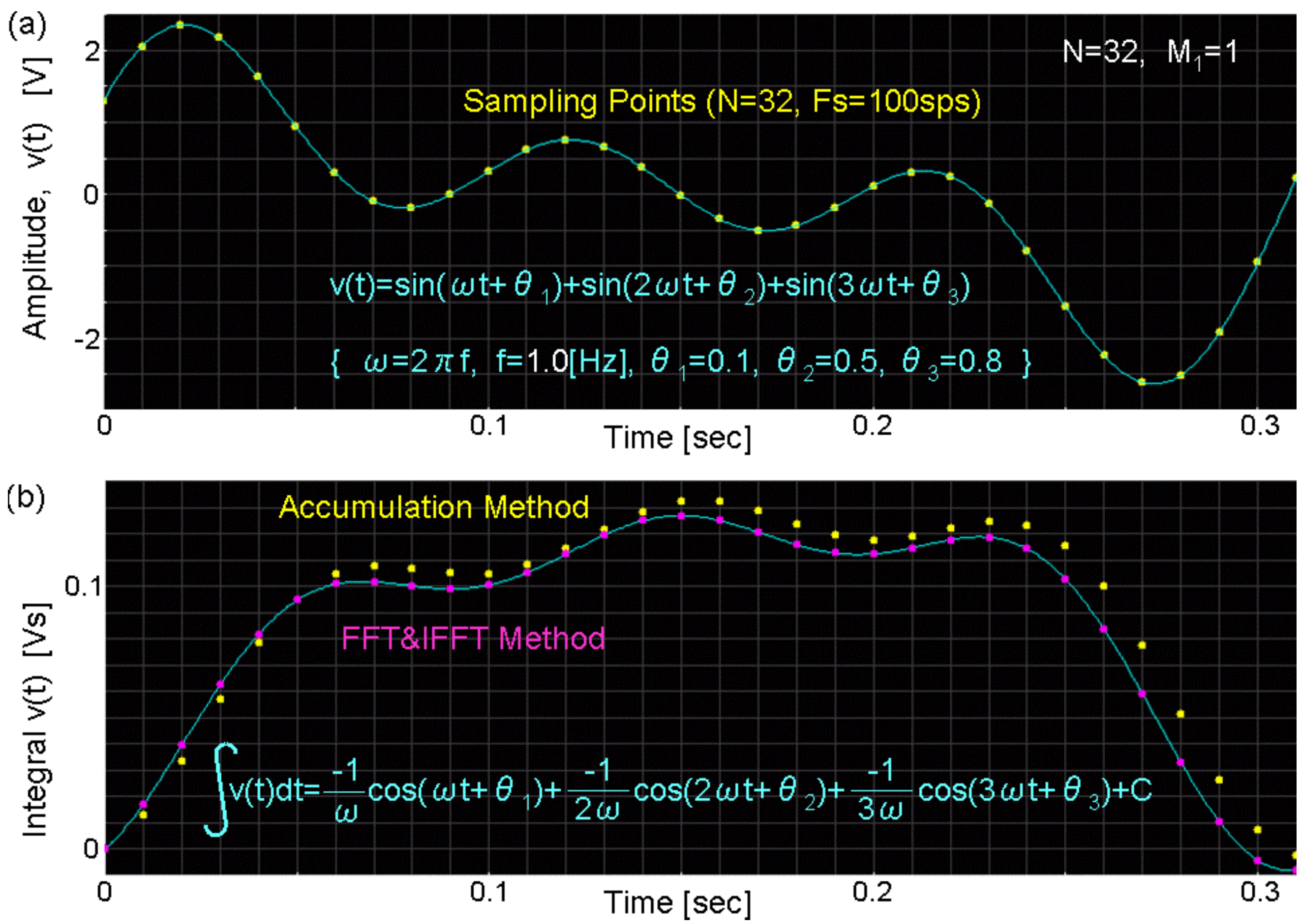

Figure 5: Integral Result under Good Integer Number Condition (M_1=1)

Figure 5 shows the case under the good integer number condition where the original waveform is constructed by the combination of exact 1-cycle, 2-cycle and 3-cycle sinusoids. Figure 6 is the case when the coherency is lost. The original curve in Figure 6(a) is the same as Figure 3(a), constructed by the combination of 1.1-cycle, 2.2-cycle, and 3.3-cycle sinusoids. In the integral operation, error would be accumulated so that FFT&IFFT method cannot deliver any good approximate result when coherency is not conformed. The simple accumulation method seems still better than FFT&IFFT method in this case.

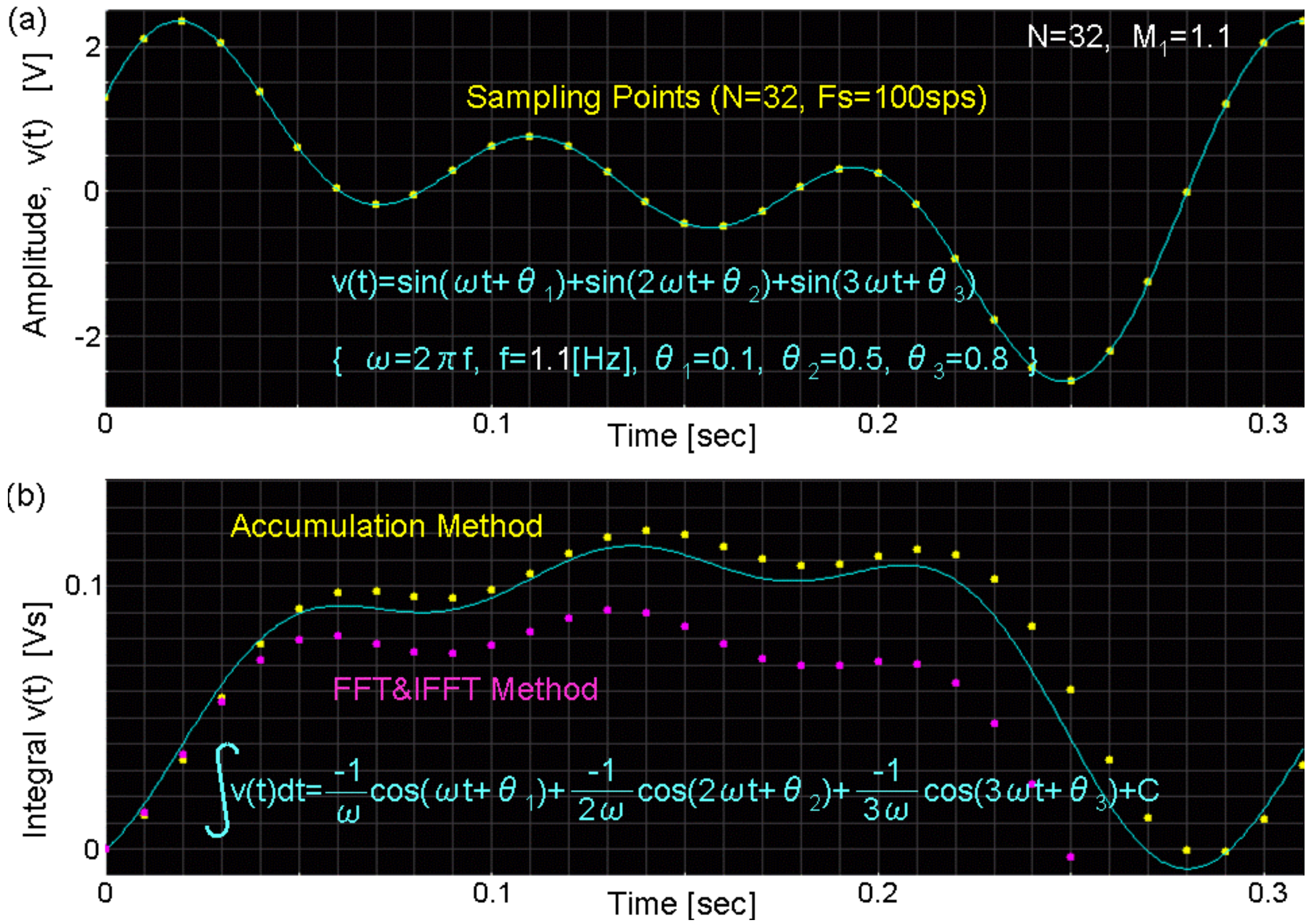

Figure 6: **Integral Result under Bad Fractional Number Condition (M_1=1.1)**

As a summary, when strict coherency condition is conformed, FFT&IFFT method is applicable in both differential and integral operations. However, once whole number condition is lost, conventional approximation method is better than FFT&IFFT method.