Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 28
PM & FM Waveform Generation II
(Orthogonal Method)

Verigy Japan
August 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

Some devices containing demodulation functionalities need PM and FM signals, which should be generated by AWG.[1] The basic waveform generation ideas of PM and FM were discussed in the previous article,[2] where the carrier and modulation waveforms are created straightforwardly by calculating instantaneous phase of a rotating vector. This method is good enough for a simple sinusoidal carrier and modulation signals. Another elegant method is available, which is more flexible. It is the topic of the month.

PM (Phase Modulation)

Let's describe a carrier signal as $A\cos(\omega_p t+\theta)$ where A is the amplitude, ω_p is the frequency and θ is the phase offset. When it is phase-modulated, θ is represented as a function of time t, and it can be described as follows;

$$\begin{aligned} v(t) &= A\cos\left(\omega_p t + \theta(t)\right) \\ &= A\cos\left(\omega_p t\right)\cos\left(\theta(t)\right) - A\sin\left(\omega_p t\right)\sin\left(\theta(t)\right) \end{aligned} \tag{1}$$

In the direct method discussed in the previous article, the instantaneous phase $\omega_p t+\theta(t)$ is straightforwardly calculated. In the new method discussed in this month's issue, the second line of Equation (1) is the point. The second line of the equation can be illustrated as a diagram with using component functions as follows;

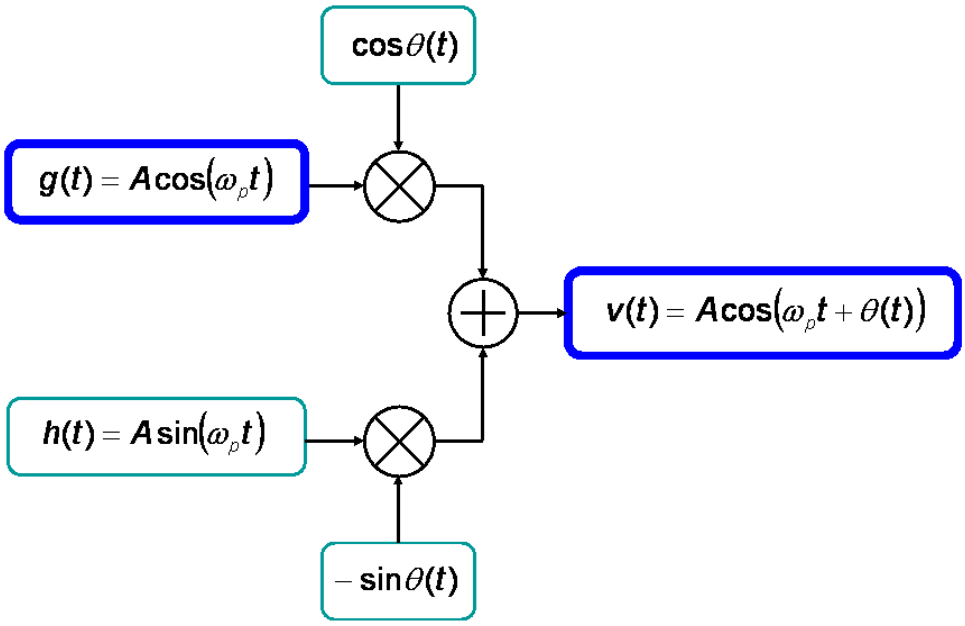


Figure 1: Diagram of Equation (1)

The diagram suggests that the PM modulated signal can be composed by multiplication and summation of simple component signals. The original carrier is $g(t)$. The newly introduced $h(t)$ is a

[1] Arbitrary Waveform Generator

[2] Mixed-signal Lecture Series: DSP-Based Testing – Fundamentals 25 PM & FM Waveform Generation I

modified carrier. The modulation signal $\theta(t)$ is incorporated in the cosine and sine functions. The final result is the modulated signal $v(t)$. The relationship of $g(t)$ and $h(t)$ is cosine and sine. So the modified carrier $h(t)$ can be generated by rotating the original $g(t)$ by -90 degrees. Consequently Figure 1 can be elaborated as Figure 2, which represents the whole picture of PM generation data processing.

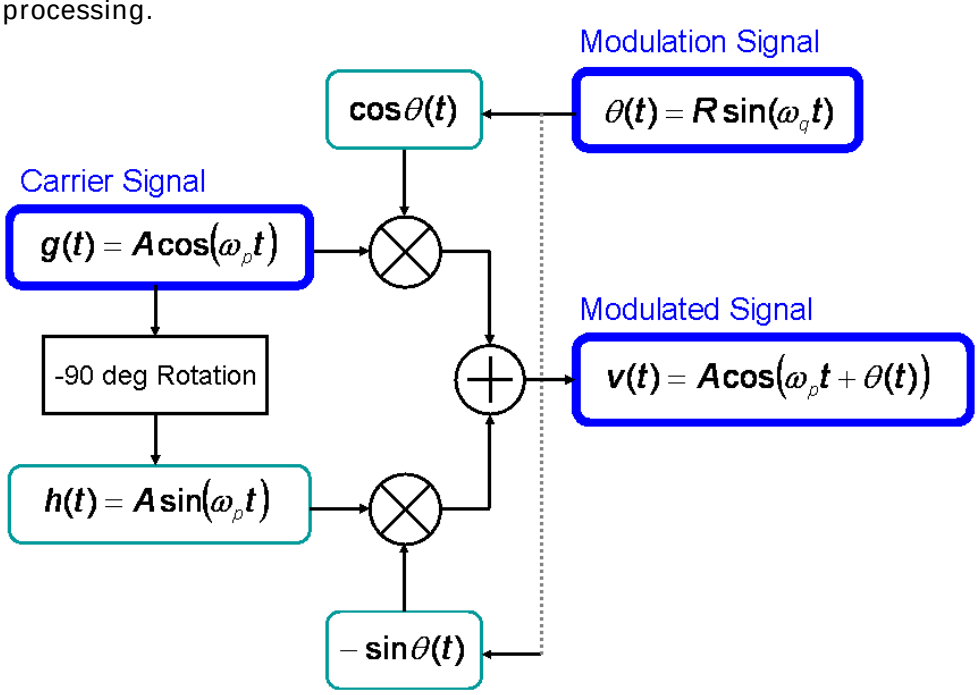
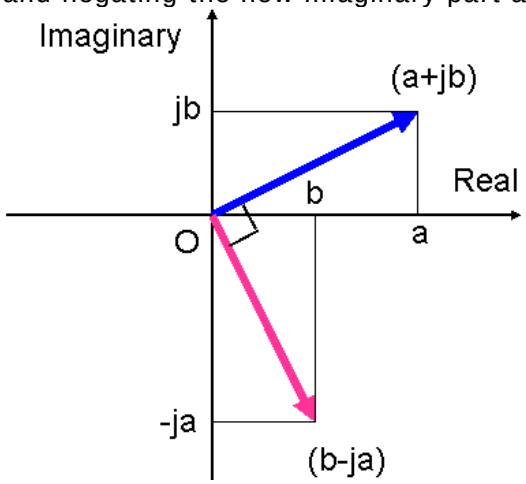


Figure 2: Diagram of PM Data Processing

This diagram can be programmed as the code in List 1. The carrier $g(t)$ generated at Line 13 is converted into the frequency domain spectrum at Line 16. Each one of the spectrum components which is complex number is rotated -90 degrees by multiplying –j (the negative imaginary unit) one by one. (Lines 20 and 21) This calculation is realized by exchanging the real and imaginary parts and negating the new imaginary part as Figure 3 depicts.



$$(-j)\cdot(a+jb) = b - ja$$

Figure 3: Rotation of Vector ($a+jb$) by -90 Degrees

Then IFFT processes the rotated frequency component to reconstruct $h(t)$. The modulation signal, which is described as $\theta(t)$ in Figure 2, is generated at Line 34. (dU[i]) The modulation signal is

incorporated into cosine and sine functions at Lines 38 to 41. The $g(t)$*cosine and $h(t)$*sine multiplications are performed at Lines 42 and 43, and the summation is done at Line 44.

```
01:   INT            i,Nawg,Np,Nq;
02:   DOUBLE         dFawg,dFresln,dFp,dP,dFq,dQ,dR;
03:   ARRAY_D        dGwave,dHwave,dU,dCOS,dSIN,dXwave,dYwave,dVwave;
04:   ARRAY_COMPLEX CGwave,CGsp,CHwave,CHsp;
05:
06:   dFawg=102.4 MHz;                          // AWG Condition
07:   Nawg=1024;
08:   dFresln=dFawg/Nawg;
09:   dFp=5.1 MHz;                              // Carrier Frequency
10:   Np=51;                                    // Np=(INT)(dFp/dFresln+0.5)
11:   dP=2.0*M_PI*Np/Nawg;
12:   dGwave.resize(Nawg);
13:   for (i=0;i<Nawg;i++) dGwave[i]=cos(dP*i);// g(t) Carrier
14:
15:   DSP_CONV_D_C(dGwave,CGwave,1.0,0.0);      // ARRAY_D→ARRAY_COMPLEX
16:   DSP_FFT(CGwave,CGSp,RECT);                // → Frequency Domain
17:   CHsp.resize(Nawg);
18:   CHsp[0]=CGsp[0];
19:   for (i=1;i<(Nawg/2);i++) {
20:      CHsp[i].real()= CGsp[i].imag();        // (-j) Multiplication
21:      CHsp[i].imag()=-CGsp[i].real();        // (-j)(A+jB)=B-jA
22:      CHsp[Nawg-i].real()= CHsp[i].real();   // Complex Conjugate
23:      CHsp[Nawg-i].imag()=-CHsp[i].imag();
24:   }
25:   CHsp[Nawg/2]=CZero();
26:   DSP_IFFT(CHsp,CHwave);                    // → Time Domain
27:   dHwave=CHwave.getReal();                  // h(t) = Rotated g(t)
28:
29:   dFq=300.0 kHz;                            // Modulation Frequency
30:   Nq=3;                                     // Nq=(INT)(dFq/dFresln+0.5)
31:   dQ=2.0*M_PI*Nq/Nawg;
32:   dR=(120.0/180.0)*M_PI;                    // Phase Deviation
33:   dU.resize(Nawg);                          // Theta
34:   for (i=0;i<Nawg;i++) dU[i]=dR*sin(dQ*i); // Modulation Signal
35:
36:   dCOS.resize(Nawg);
37:   dSIN.resize(Nawg);
38:   for (i=0;i<Nawg;i++) {
39:      dCOS[i]= cos(dU[i]);                   //  cos(theta(t))
40:      dSIN[i]=-sin(dU[i]);                   // -sin(theta(t))
41:   }
42:   DSP_MUL_VEC(dGwave,dCOS,dXwave);          // x(t)=g(t)*{ cos(theta(t)}
43:   DSP_MUL_VEC(dHwave,dSIN,dYwave);          // y(t)=h(t)*{-sin(theta(t)}
44:   DSP_ADD_VEC(dXwave,dYwave,dVwave);        // v(t)=x(t)+y(t)
45:
```

List 1: PM Program Example

Figure 4 shows the waveforms generated by the code in List 1. (a) is the carrier, (b) is its -90 degrees rotated signal, (c) is the modulation signal, and (d) is the final result of the PM modulated signal. Carefully examining the relationship of signals (c) and (d), the modulated signal $v(t)$ looks dense at rising slopes and sparse at falling slopes of the modulation signal $u(t)$. The density of lines of $v(t)$ corresponds to the instantaneous frequency of the signal.

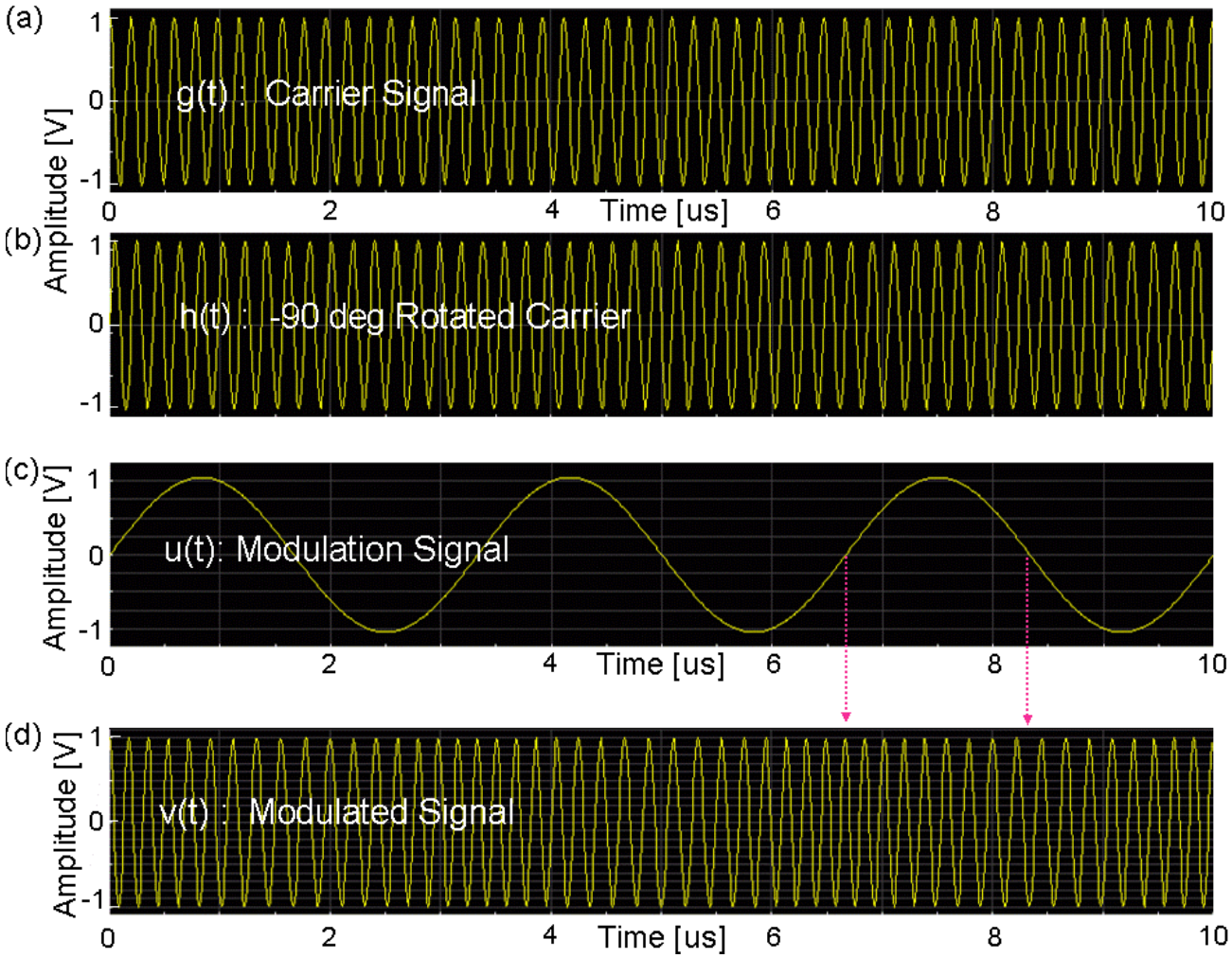


Figure 4 Waveforms in PM Signal Processing

The FFT spectrum of the modulated signal of Figure 4(d) is shown in Figure 5.

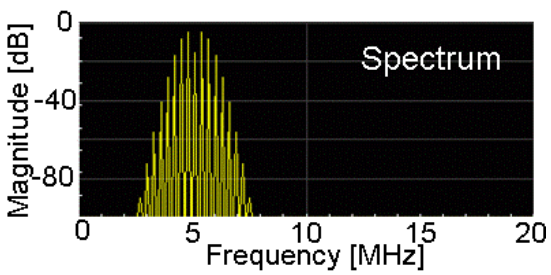


Figure 5 Frequency Spectrum of PM

The instantaneous frequency of the signal $v(t)$ is analyzed precisely as Figure 6(a), which is compared with the modulation signal in Figure 6(b). When the phase or the modulation signal is decreasing, the frequency of the modulated signal is low. When the phase is increasing, the frequency becomes high. So the gradient of the modulation signal curve looks proportional to the instantaneous frequency.

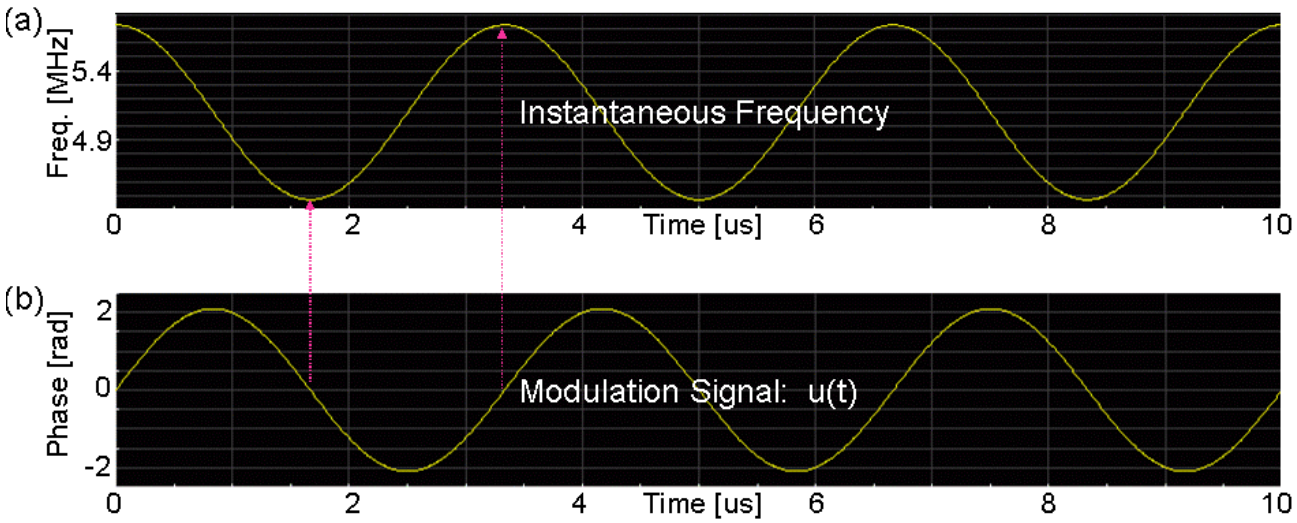


Figure 6 Frequency Trend of PM and Modulation Signal

FM (Frequency Modulation)

In FM modulation, the instantaneous frequency deviation is proportional to the modulation signal. By examining Figures 6(a) and (b) of PM, you can notice that the instantaneous frequency trend (a) is proportional to the slope of the modulation signal (b). This relationship corresponds to differential of mathematics. So if you can perform integral to the modulation signal in advance, you can expect that the instantaneous frequency trend of the final signal is proportional to the modulation signal movement. Consequently FM modulation could be realized by utilizing PM data processing scheme. The point is the pre-processing of integral to the modulation signal. The whole picture of FM data processing is illustrated as follows;

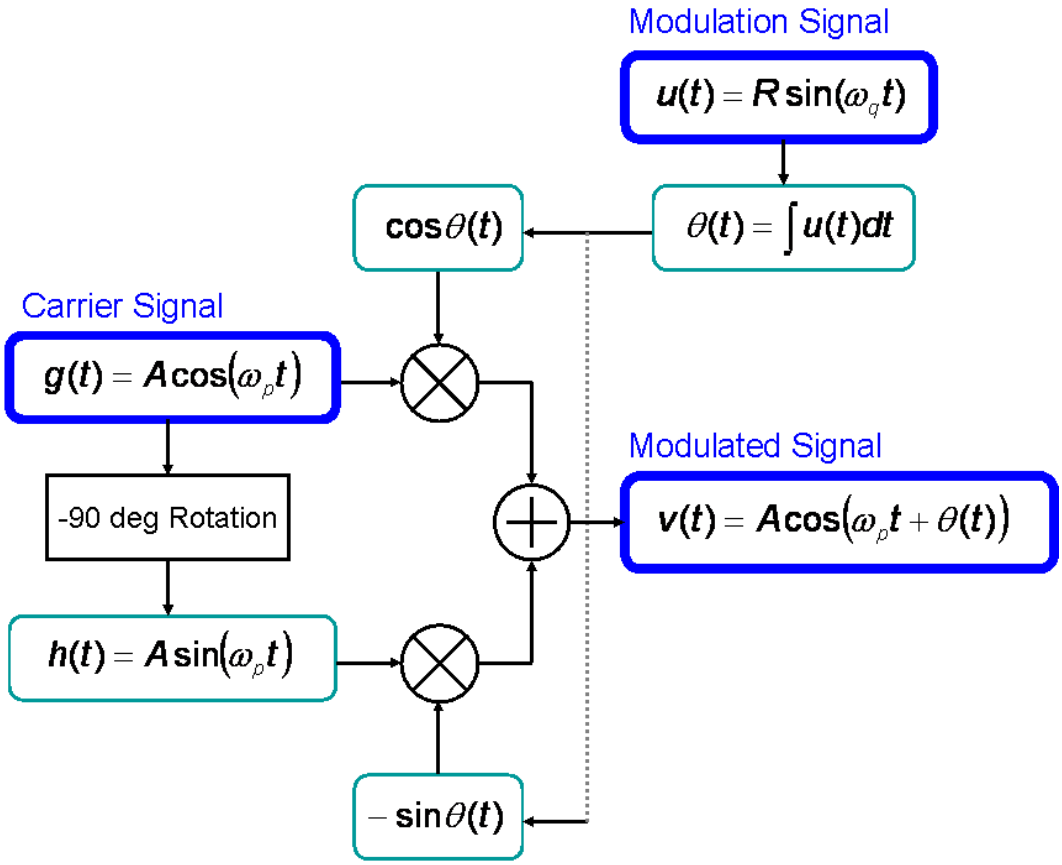


Figure 7 Diagram of FM Data Processing

PM processing is already established in the previous section. So the key in FM processing is the integral operation. In the last GoSemi newsletter article,[3] differential and integral operations are discussed. They can be performed by FFT&IFFT. Therefore the integral in FM processing in Figure 7 can be realized by another FFT&IFFT in addition to the processing for -90 degree rotation. Only one restriction for two FFT&IFFT operations is the whole number cycle condition for both the carrier signal and the modulation signal. So you must settle your test signal specification not to contain any discontinuity in those signals.

Let's look at the example program code in Lists 2.1-2.3. As you can see, the operation of List 2.1 is the same as List 1 in the PM discussion. This part creates the -90 degrees rotated carrier signal.

[3] Mixed Signal Lecture Series: DSP-Based Testing – Fundamentals 26 Differential/Integral Operation by FFT&IFFT

```
01:   INT            i,Nawg,Np,Nq;
02:   DOUBLE         dFawg,dFresln,dFp,dP,dFq,dQ,dR,dFd,dw,dkw;
03:   ARRAY_D        dGwave,dHwave,dCOS,dSIN,dXwave,dYwave,dVwave;
04:   ARRAY_D        dU,dU2,dIntU;
05:   ARRAY_COMPLEX CGwave,CGsp,CHwave,CHsp,CU2,CU2sp,CIntSp;
06:
07:   dFawg=102.4 MHz;                              // AWG Condition
08:   Nawg=1024;
09:   dFresln=dFawg/Nawg;
10:   dFp=5.1 MHz;                                  // Carrier Frequency
11:   Np=51;                                        // Np=(INT)(dFp/dFresln+0.5
12:   dP=2.0*M_PI*Np/Nawg;
13:   dGwave.resize(Nawg);
14:   for (i=0;i<Nawg;i++) dGwave[i]=cos(dP*i); // g(t) Carrier
15:
16:   DSP_CONV_D_C(dGwave,CGwave,1.0,0.0);       // ARRAY_D→ARRAY_COMPLEX
17:   DSP_FFT(CGwave,CGSp,RECT);                  // → Frequency Domain
18:   CHsp.resize(Nawg);
19:   CHsp[0]=CGsp[0];
20:   for (i=1;i<(Nawg/2);i++) {
21:      CHsp[i].real()= CGsp[i].imag();          // (-j) Multiplication
22:      CHsp[i].imag()=-CGsp[i].real();          // (-j)(A+jB)=B-jA
23:      CHsp[Nawg-i].real()= CHsp[i].real();     // Complex Conjugate
24:      CHsp[Nawg-i].imag()=-CHsp[i].imag();
25:   }
26:   CHsp[Nawg/2]=CZero();
27:   DSP_IFFT(CHsp,CHwave);                      // → Time Domain
28:   dHwave=CHwave.getReal();                    // h(t) = Rotated g(t)
```

List 2.1: FM Program Example (Part 1)

Figure 8(a) shows the carrier signal $g(t)$ which is 5.1MHz sinusoid. Figure 8(b) shows the signal $h(t)$ which is -90 degrees rotated sinusoid to $g(t)$.

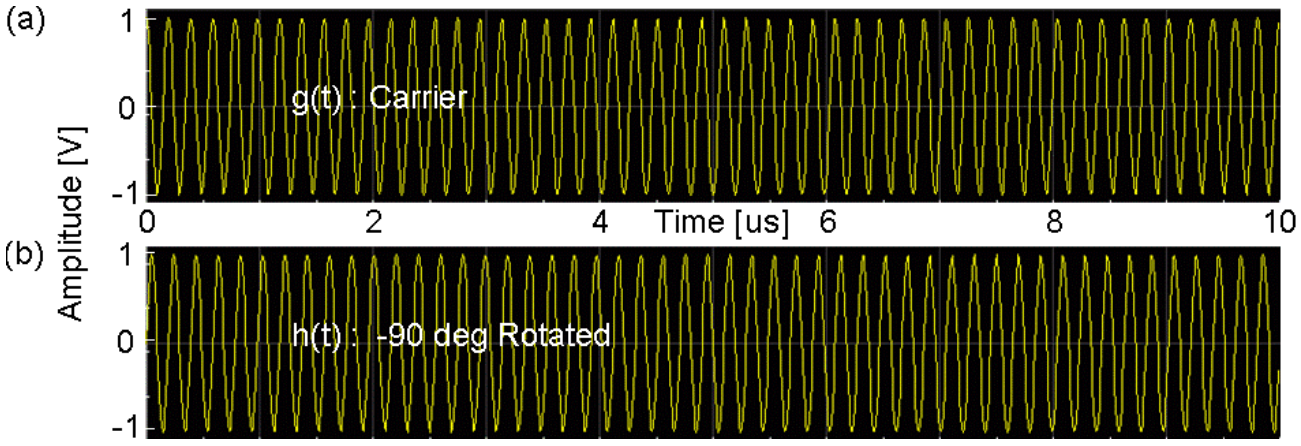


Figure 8: Carrier and -90 Degrees Rotated Signal

```
29:
30:   dFd=2.0 MHz;                                  // Frequency Deviation (pp)
31:   dR=dFd/2.0;
32:   dFq=300.0 kHz;                                // Modulation Frequency
33:   Nq=3;                                         // Nq=(INT)(dFq/dFresln+0.5
34:   dQ=2.0*M_PI*Nq/Nawg;
35:
36:   dU.resize(Nawg);
37:   for (i=0;i<Nawg;i++) dU[i]=dR*sin(dQ*i);  // [Hz]  Modulation Signal
38:   DSP_MUL_SCL(2.0*M_PI,dU,dU2);             // [rad] 2pi*(Freq.Dev)
39:
40:   DSP_CONV_D_C(dU2,CU2,1.0,0.0);            // Double→Complex
41:   DSP_FFT(CU2,CU2sp,RECT);
42:   CIntSp.resize(Nawg);
43:   dw=2.0*M_PI*dFresln;                      // omega
44:   for (i=0;i<(Nawg/2);i++) {
45:      dkw=1.0/(dw*i);
46:      CIntSp[i].real()= CU2sp[i].imag()*dkw;  // Integral Operation
47:      CIntSp[i].imag()=-CU2sp[i].real()*dkw;  // -j(A+jB)=B-jA
48:      CIntSp[Nawg-i].real()= CIntSp[i].real();// Complex Conjugate
49:      CIntSp[Nawg-i].imag()=-CIntSp[i].imag();
50:   }
51:   CIntSp[Nawg/2]=CZero();                   // Complex Zero
52:   DSP_IFFT(CIntSp,CIntU);
53:   dIntU=CIntU.getReal();                    // Integral u(t)
54:
```

List 2.2: FM Program Example (Part 2)

The frequency deviation and the modulation frequency are consolidated into the modulation signal at Lines 30 through 38 in List 2.2. The integral of the modulation signal is performed at Lines 40 through 53. For the further detail regarding integral, consult the last newsletter article.[4]

```
55:   dCOS.resize(Nawg);                         //  PM
56:   dSIN.resize(Nawg);
57:   for (i=0;i<Nawg;i++) {
58:      dCOS[i]= cos(dIntU[i]);                //  cos(Integral u(t))
59:      dSIN[i]=-sin(dIntU[i]);                // -sin(Integral u(t))
60:   }
61:   DSP_MUL_VEC(dGwave,dCOS,dXwave);          //  x(t)=g(t)*{ cos(Iu(t)}
62:   DSP_MUL_VEC(dHwave,dSIN,dYwave);          //  y(t)=h(t)*{-sin(Iu(t)}
63:   DSP_ADD_VEC(dXwave,dYwave,dVwave);        //  v(t)=x(t)+y(t)
64:
```

List 2.3: FM Program Example (Part 3)

The integral operated signal is incorporated into the cosine and sine waveforms at Lines 57 through 60. The original signal $g(t)$ is multiplied to the modified cosine at Line 61, and the -90 deg rotated signal $h(t)$ is multiplied to the modified sine at Line 62. Finally the multiplied waveforms are summed together at Line 63. This is the way FM can be realized by utilizing PM operation.

---

[4] Mixed Signal Lecture Series: DSP-Based Testing – Fundamentals 26 Differential/Integral Operation by FFT&IFFT

Figure 9 illustrates the waveforms in the FM modulation process described in List 2. (a) is the 300kHz modulation signal $u(t)$ generated with +/-1[MHz] of amplitude. (b) is the integral of $u(t)$, which is denoted as $\theta(t)$. By multiplying $\cos\theta(t)$ to $g(t)$ and $-\sin\theta(t)$ to $h(t)$, and they are summed as $v(t)$, shown in (c). The instantaneous frequency of $v(t)$ is analyzed and illustrated in (d), whose trend looks very similar to (a).

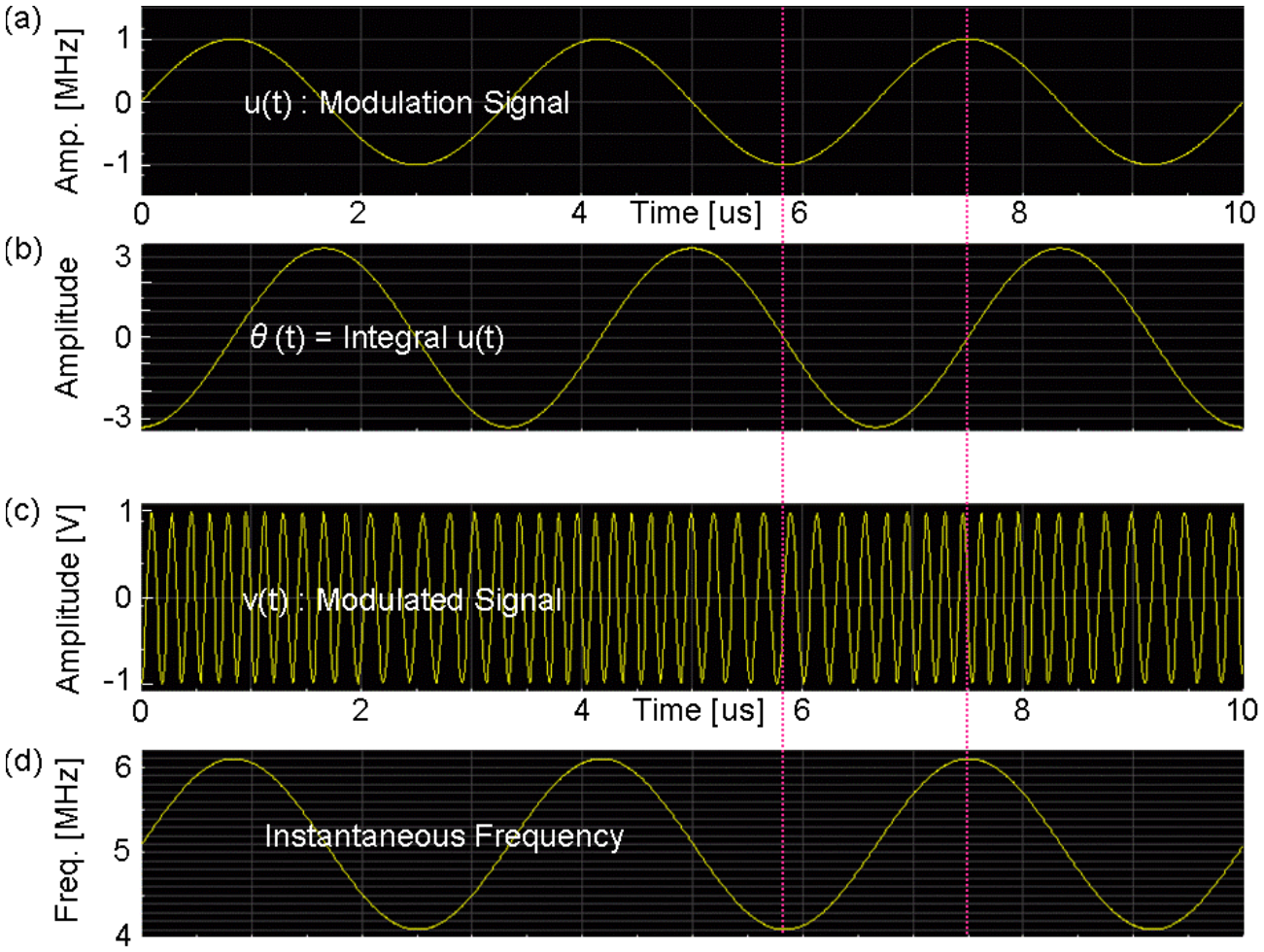


Figure 9 FM Modulated Signal

The FFT spectrum of the signal $v(t)$ is shown in Figure 10.

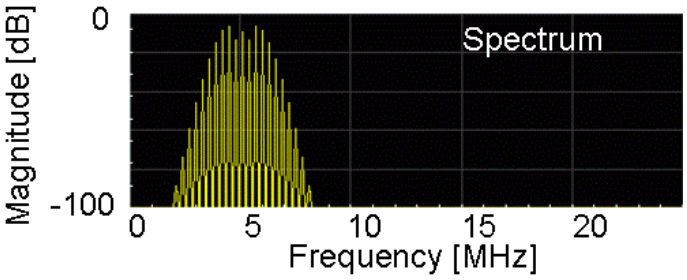


Figure 10 FM Spectrum

Summary

1. PM and FM modulated signals can be generated by the signal processing illustrated in the diagrams below.

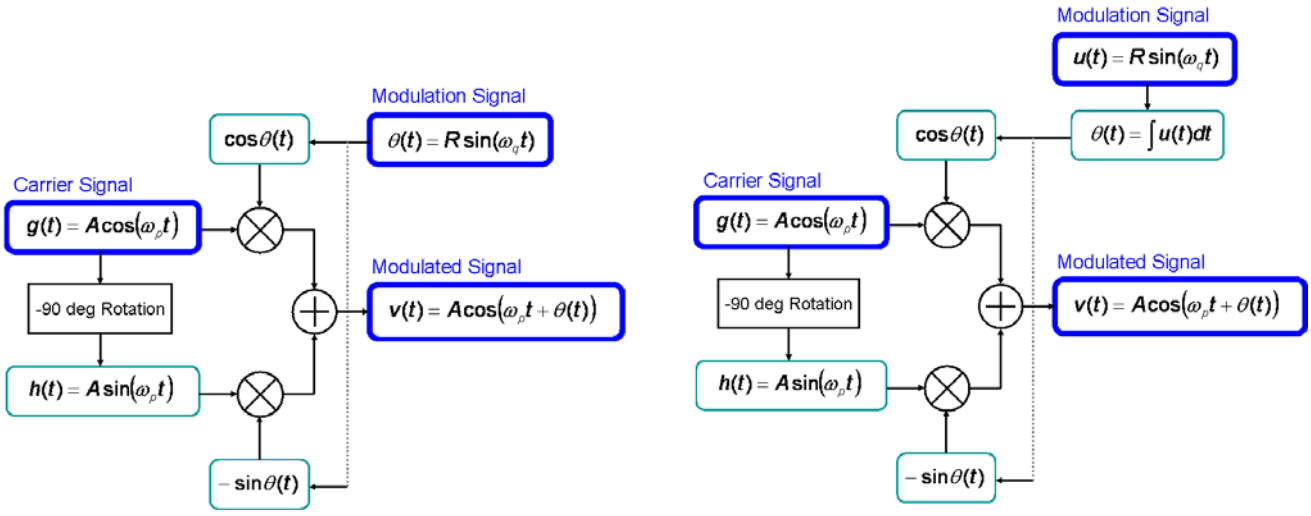


Figure 11: Block Diagrams of PM (Left) and FM (Right) Signal Processing

2. FM utilizes the PM processing. FM performs integral in advance to the modulation signal.

3. Operations of the -90 degrees rotation and the integral are realized by using FFT&IFFT, so that the number of signal cycles in the unit waveform period must be designed as whole numbers.

4. The signal processing scheme discussed here can be applied to more complex carrier and modulation signals. (Separate newsletter articles will be reported about specific situations.)

5. With looking at the diagrams in Figure 11, this processing technique may be called “orthogonal method” or “I/Q method”.