Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 41
FM Stereo Waveform Generation

Verigy Japan
October 2011

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

Commercial FM radio systems broadcast audio stereo programs. Testing of FM receiver devices requires FM signals modulated by stereo-composite signals as stimuli. This article introduces a practical example of FM signal generation.

Composite Stereo Structure

FM stereo composite signal has a frequency spectrum structure as depicted in Figure 1. If the radio program is not stereo, the audio signal is 100% scaled and modulates the carrier signal. If the program is stereo, the left channel signal L and the right channel signal R are scaled and multiplexed as L+R and L-R, and modulate the carrier. The instantaneous frequency deviation is represented as follows;

$$\Delta f = \left(0.9 \times \left(\frac{L+R}{2} + \frac{L-R}{2} \sin 2\omega_p t \right) + 0.1 \times \sin \omega_p t \right) \times f_{dev} \qquad (1)$$

where ω_p is the pilot frequency (angular frequency) and f_{dev} is a frequency deviation of FM. The pilot frequency is specified as 19 kHz. The frequency deviation f_{dev} is specified as 75 kHz.
If a radio receiver can receive monaural signal only, it receives the L+R signal so that the audio signal is not lost. The stereo composite signal is compatible to the monaural only radio.

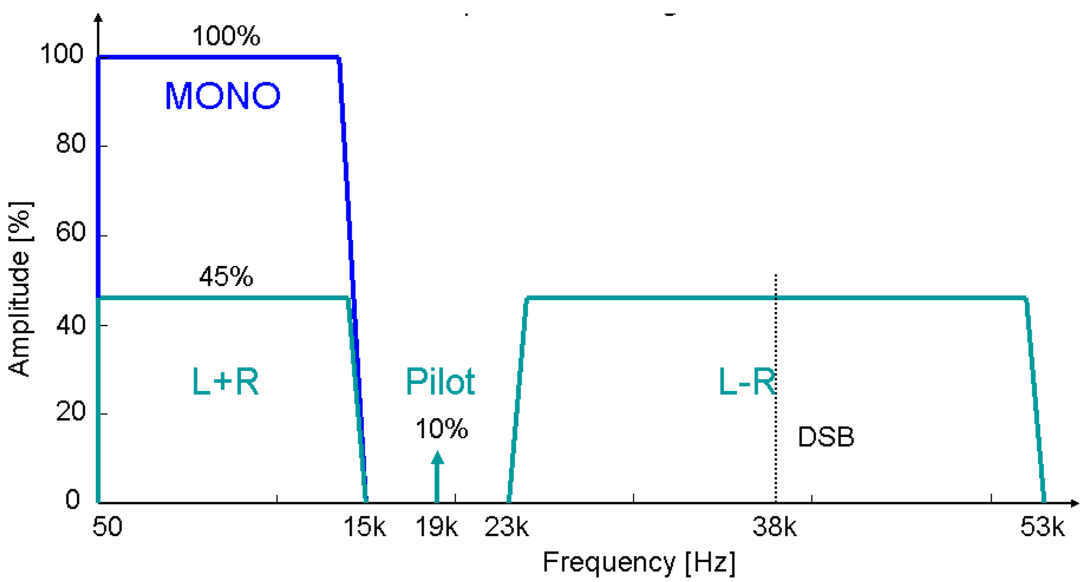

Figure 1: Stereo Composite Signal

The pilot signal is an indicator of stereo signal. If it is detected in a received signal, the receiver demodulates the signal in the stereo mode. The pilot signal has another mission. Its frequency is doubled so that it becomes 38 kHz, and the sub-carrier signal for DSB (double side band) demodulation is generated with refering the pilot.
Figure 2 illustrates Equation (1) as a block diagram for calculation. This diagram suggests how to program a composite signal. The L and R-channel signals are scaled by 45% and combined into two signals of L+R and L-R. The L-R signal is multiplied by the sub-carrier. Let's denote the L and R signals as $A_L\cos(\omega_L t + \phi_L)$ and $A_R\cos(\omega_R t + \phi_R)$. Then the DSB modulation of (L-R) and the sub-carrier can be described as follows;

$$
\begin{aligned}
(L-R)\sin 2\omega_p t &= \{A_L\cos(\omega_L t+\phi_L)-A_R\cos(\omega_R t+\phi_R)\}\sin 2\omega_p t \\
&= A_L\cos(\omega_L t+\phi_L)\cdot\sin 2\omega_p t - A_R\cos(\omega_R t+\phi_R)\cdot\sin 2\omega_p t \\
&= \frac{A_L}{2}\{\sin((2\omega_p+\omega_L)t+\phi_L)+\sin((2\omega_p-\omega_L)t-\phi_L)\} \\
&\quad -\frac{A_R}{2}\{\sin((2\omega_p+\omega_R)t+\phi_R)+\sin((2\omega_p-\omega_R)t-\phi_R)\}
\end{aligned}
\tag{2}
$$

As Equation (2) shows, the key of the DSB modulation is that the sub-carrier component does not remain at the end of the multiplication. So when demodulating the DSB signal, the sub-carrier signal must be synthesized by the pilot signal. Therefore the phase relationship of the pilot and the sub-carrier is important. The pilot signal is not just a stereo mode indicator but the reference signal for generating the sub-carrier in radio receivers. If the sub-carrier would not be synthesized correctly, the demodulated sound could not be separated into the L and R channels. Then the both L and R signals can be heard from the L and R speakers or headphones.

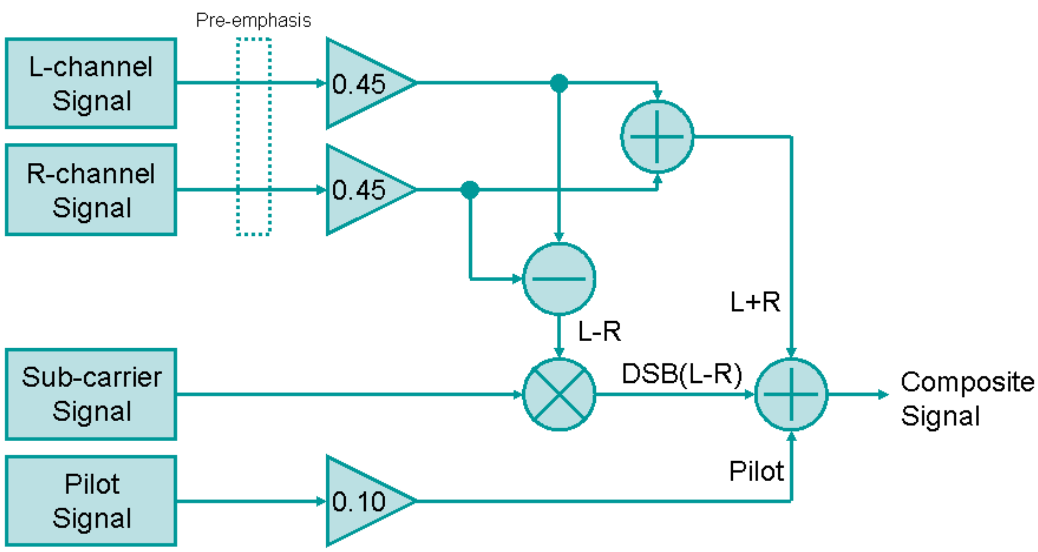

Figure 2: Stereo Composite Signal Generation

When the stereo composite signal is correctly demodulated, L+R and L-R are recovered. When they are summed up as (L+R)+(L-R)=2L, the L-channel signal can be extracted. When they are subtracted as (L+R)-(L-R)=2R, the R-channel signal can be extracted.
Figure 3 shows the experimentally programmed audio test signals. The 19 kHz pilot signal must accurately be generated. So the total number of unit waveform points N is settled to 524,288 points, and the sampling frequency Fs is settled to 498.0736 Msps. Then the frequency resolution becomes 950 Hz (=19 kHz/20). A single cycle cosine waveform is programmed for the L channel, and three cycle cosine waveform is programmed for the R channel. Their amplitudes are scaled to 0.45. Figure 3 shows L+R and L-R signals as well.
Figure 4 shows the DSB modulation. The sub-carrier is created as a sine waveform with the phase offset of zero. The L-R signal is multiplied by the sub-carrier, and the L-R waveform appears on the envelope of the DSB signal.
Figure 5 shows the stereo composite signal. The L+R signal, the DSB signal and the 10% pilot signal are simply summed up altogether. The pilot is a sine with a half of the phase offset of the sub-carrier so that the pilot's phase is zero. The frequency spectrum of the composite is illustrated in the figure too. The DSB magnitude looks halved compared to Figure 1, however the L and –R signals are split into the LSB and USB components and the total energy of the L-R is the same as the L+R. This is

normal.

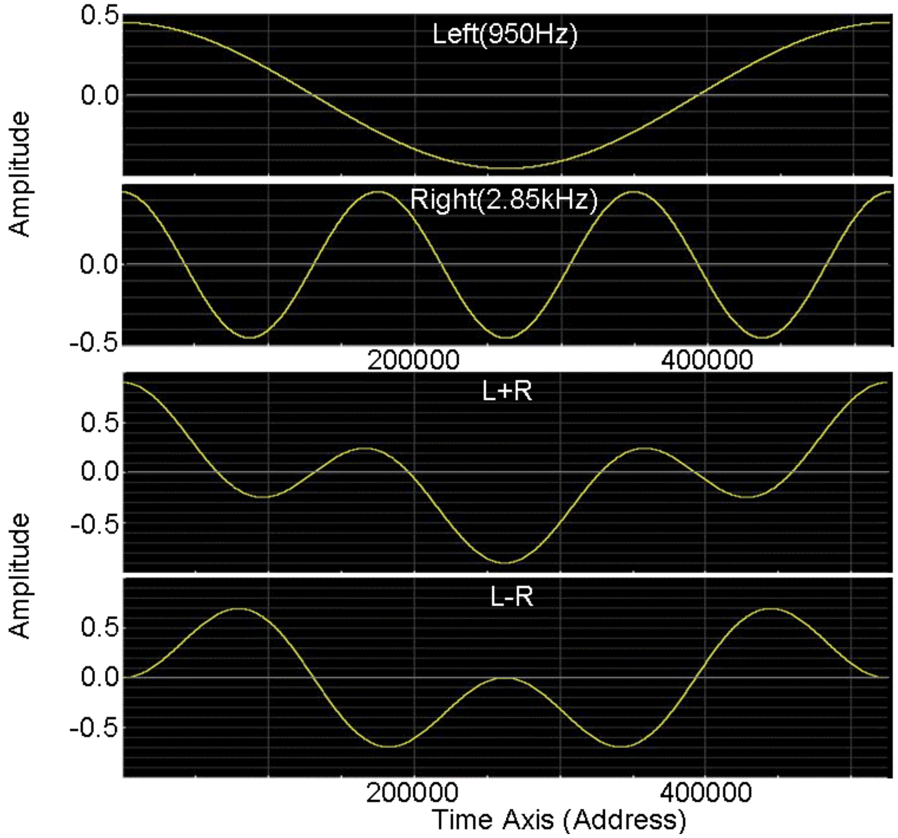

Figure 3: **L, R-Channel Inputs and L+R, L-R**

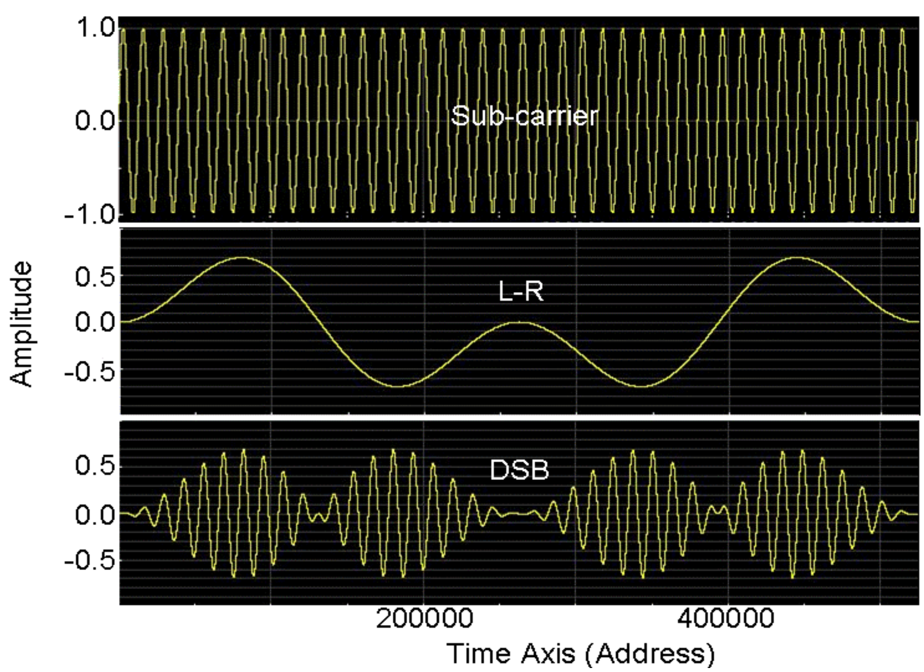

Figure 4: **DSB Modulation**

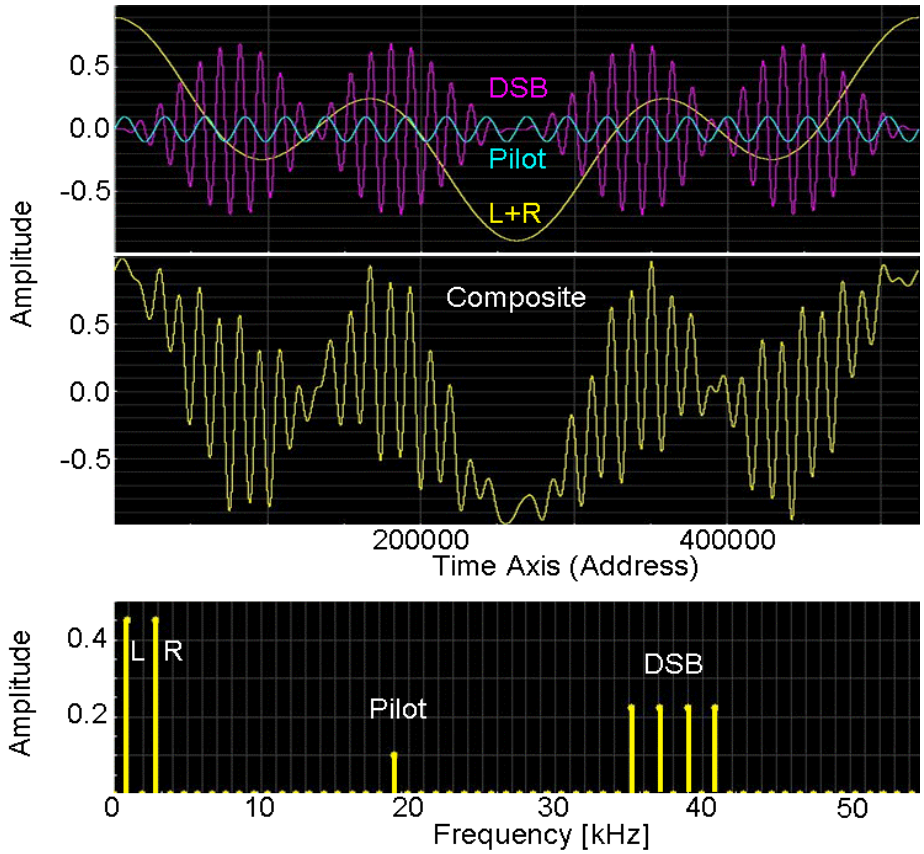

Figure 5: **Stereo Composite**

FM Generation

FM modulation processing is already discussed in the past Newsletter article.[1] Figure 6 is the block diagram of the signal processing. As discussed in that issue, the FM processing makes use of the PM (phase modulation) processing. The modulation signal is an integral of the composite signal in this case. So the composite signal should be integrated in advance and then wrapped with the cosine and sine functions.

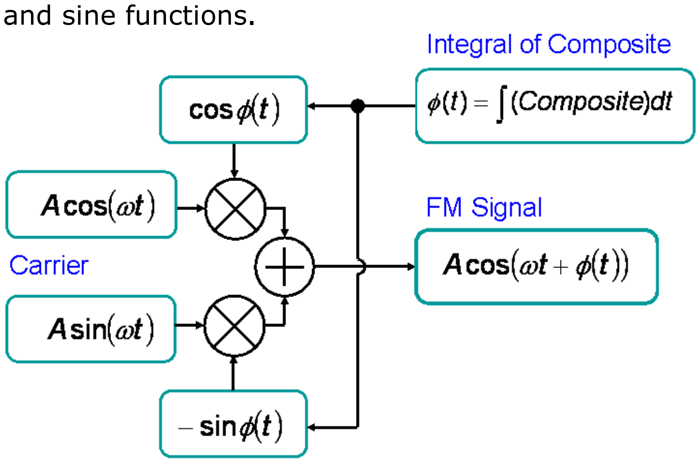

Figure 6: **FM Processing**

[1] DSP–Based Testing — Fundamentals 28 PM & FM Waveform Generation II

Figure 7 shows the composite signal and its integrated waveform, which is scaled to the frequency deviation of 75 kHz and converted into the unit of radian. This waveform is directly wrapped with cosine and sine functions and each of them are multiplied to the RF carrier cosine and sine waveforms.
The carrier frequency is selected as 76 MHz which is an exact multiple of 950 Hz. According to the diagram of Figure 6, the carrier signal is created as cosine and sine waveforms. The cosine carrier waveform is multiplied by the cosine modulation component, the sine carrier waveform is multiplied by the sine modulation component respectively, and they are combined and scaled to 0.5V of the maximum amplitude for downloading to an AWG (arbitrary waveform generator). This is the final FM stereo RF signal. Figure 8 illustrates the FFT spectrum of the FM RF signal.

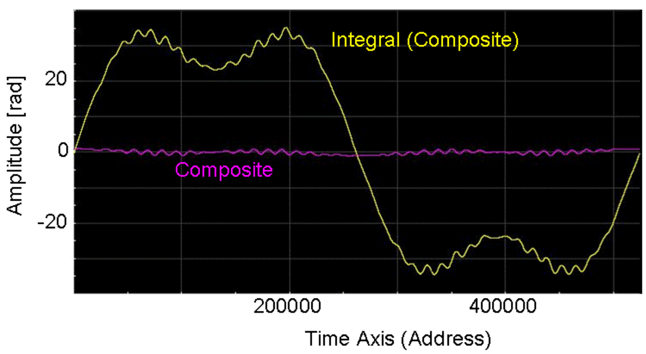

Figure 7: Integrated Stereo Composite

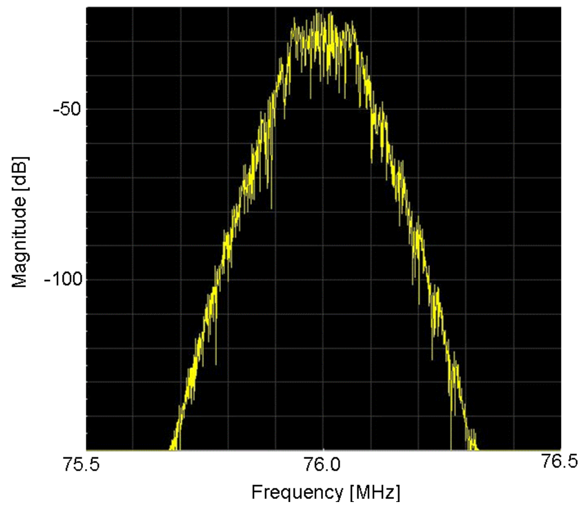

Figure 8: 76MHz FM Signal

This final waveform is downloaded into the VHF AWG in the MBAV8, and when the output signal is received by an appropriate radio receiver, you can actually hear the 950 Hz from the left speaker and the 2.85 kHz from the right speaker clearly separated with each other. If the pilot signal would not correctly be created, two tones would be heard from both speakers.

Example Program Code

```
INT           i,Nawg,Nsp,iMax,iMin,Mleft,Mrght,Mplt,Msub,Mcar;
DOUBLE        dFawg,dFresln,dFleft,dFrght,dFplt,dFsub,dFcar;
DOUBLE        dPleft,dPrght,dPplt,dPsub,dPcar,dAleft,dArght;
DOUBLE        dQleft,dQrght,dFdev,dR,dkw,dw,dMax,dMin;
ARRAY_D       dWaveLEFT,dWaveRGHT,dWavePLT,dWaveSUB,dWaveDSB,dWaveCMP;
ARRAY_D       dWaveLpR,dWaveLmR,dWave1,dWave2,dWaveCAR1,dWaveCAR2;
ARRAY_D       dWaveX,dWaveY,dWaveFM,dCOS,dSIN,dIntCMP,dSpFM;
ARRAY_COMPLEX CWaveCMP,CIntSp,CIntCMP,CSpCMP;
//-----------------------------------------------------------------------------
Nawg=524288;
Nsp=Nawg/2;
//-----------------------------   Pilot Signal    ------------------------------
dFplt=19.0 kHz;                                  // Pilot Freq. (Reference)
Mplt=20;                                         // # of cycles
dFresln=dFplt/(DOUBLE)Mplt;                      // Freq. Resolution 950Hz
dFawg=dFresln*Nawg;                              // AWG 498.0736Msps

dPplt=2.0*M_PI*Mplt/Nawg;
dWavePLT.resize(Nawg);
for (i=0;i<Nawg;i++) dWavePLT[i]=0.1*sin(dPplt*i); // 10% sin(No phase offset)
//-----------------------------   Sub-carrier Signal      ----------------------
dFsub=2.0*dFplt;
Msub=2*Mplt;
dPsub=2.0*M_PI*Msub/Nawg;
dWaveSUB.resize(Nawg);
for (i=0;i<Nawg;i++) dWaveSUB[i]=sin(dPsub*i);     // sin(No phase offset)
//---------------------------    Left Channel Signal     -----------------------
dFleft=1.0 kHz;
Mleft=(INT)(dFleft/dFresln+0.5);
dFleft=dFresln*Mleft;                          // 950Hz
dPleft=2.0*M_PI*Mleft/Nawg;
dAleft=1.0;                                    // Full-scale 1.0
dQleft=0.0;                                    // Phase offset
dWaveLEFT.resize(Nawg);                        // 45%
for (i=0;i<Nawg;i++) dWaveLEFT[i]=0.45*dAleft*cos(dPleft*i+dQleft);
//---------------------------    Right Channel Signal     ----------------------
dFrght=3.0 kHz;
Mrght=(INT)(dFrght/dFresln+0.5);
dFrght=dFresln*Mrght;                          // 2850Hz
dPrght=2.0*M_PI*Mrght/Nawg;
dArght=1.0;                                    // Full-scale 1.0
dQrght=0.0;                                    // Phase offset
dWaveRGHT.resize(Nawg);                        // 45%
for (i=0;i<Nawg;i++) dWaveRGHT[i]=0.45*dArght*cos(dPrght*i+dQrght);
//---------------------------     FM Carrier Signal    -------------------------
dFcar=76.0 MHz;
Mcar=(INT)(dFcar/dFresln+0.5);
dFcar=dFresln*Mcar;                            // 76MHz
dPcar=2.0*M_PI*Mcar/Nawg;
dWaveCAR1.resize(Nawg);
dWaveCAR2.resize(Nawg);
for (i=0;i<Nawg;i++) {
   dWaveCAR1[i]=cos(dPcar*i);
   dWaveCAR2[i]=sin(dPcar*i);
}
```

```
//------------------------     (L+R)        (L-R)      ------------------------
  DSP_ADD_VEC(dWaveLEFT,dWaveRGHT,dWaveLpR);      //  L+R
  DSP_SUB_VEC(dWaveLEFT,dWaveRGHT,dWaveLmR);      //  L-R
//------------------------     AM (DSB) Modulation    ------------------------
  DSP_MUL_VEC(dWaveLmR,dWaveSUB,dWaveDSB);
//------------------------     Summing (Composite)     -----------------------
  DSP_ADD_VEC(dWaveLpR,dWavePLT,dWave1);
  DSP_ADD_VEC(dWave1,dWaveDSB,dWaveCMP);
//------------------------     FM Deviation     ------------------------------
  dFdev=75.0 kHz;                                 //  Freq. Deviation (+/-75kHz)
  dR=2.0*M_PI*dFdev;                              //  [rad]
  DSP_CONV_D_C(dWaveCMP,CWaveCMP,dR,0.0);         //  Scaling to [rad] deviation
  DSP_FFT(CWaveCMP,CSpCMP,RECT);
//------------------------     Integral Composite      -----------------------
  CIntSp.resize(Nawg);
  dw=2.0*M_PI*dFresln;
  CIntSp[0]=CSpCMP[0];
  for (i=1;i<Nsp;i++) {
     dkw=1.0/(dw*i);                              // -j/kw
     CIntSp[i].real()= CSpCMP[i].imag()*dkw;
     CIntSp[i].imag()=-CSpCMP[i].real()*dkw;
     CIntSp[Nawg-i]=Conjugate(CIntSp[i]);
  }
  CIntSp[Nsp]=COMPLEX(0.0,0.0);
  DSP_IFFT(CIntSp,CIntCMP);
  dIntCMP=CIntCMP.getReal();
//-------------------------    PM Preparation    ------------------------------
  dCOS.resize(Nawg);
  dSIN.resize(Nawg);
  for (i=0;i<Nawg;i++) {
     dCOS[i]= cos(dIntCMP[i]);
     dSIN[i]=-sin(dIntCMP[i]);
  }
//-------------------------    PM Processing   --------------------------------
  DSP_MUL_VEC(dWaveCAR1,dCOS,dWaveX);         // cos*cos
  DSP_MUL_VEC(dWaveCAR2,dSIN,dWaveY);         // sin*sin
  DSP_ADD_VEC(dWaveX,dWaveY,dWaveFM);         // Sum
//-------------------------    Amplitude Scaling for AWG    -------------------
  DSP_ABS_MINMAX(dWaveFM,&dMin,&dMax,&iMin,&iMax);
  DSP_MUL_SCL(0.5/fabs(dMax),dWaveFM,dWaveFM);// Max 0.5V Scaling for AWG
//-----------------------------------------------------------------------------
```