Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 22
Trend Removal (Part 2)

Verigy Japan

February 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.

Trend Removal (Part 2)

If you uploaded waveform data from a DUT ADC or a digitizer, you may have experienced ugly DC offset drift in the waveform and the noise floor of its FFT spectrum was extremely slanted. It can often occur when DC blocking capacitors are provided in the signal path in your DUT board. Following up the last article, these more difficult situations are discussed in this issue.

Non-integer Number of Cycles

The situation is more difficult than the previous one. Figure 1 shows the captured signal. The UTP (Unit Test Period) contains 4096 points and it captures 37.5 cycles of the sinusoidal waveform as Figure 2. For a fractional number of cycles of the sinusoid, you should expect to have a severely smeared spectrum around the tone besides the slanted noise floor because of the DC drift. Anyway, let's look at Figure 3, that shows the miserable FFT result with no windowing. It represents two serious issues. One is the spectrum leakage around the tone because of the fractional cycles in the UTP. The other is the severely slanted slope starting from the DC location because of the DC offset drift already discussed in the previous section. The strategy to address this situation is basically the same as the previous case. But you cannot apply DSP_SIN_FIT() method anymore to estimate the DC offset trend because the number of cycles in the UTP is not a whole number anymore. So if you could remove the trend and get a flat noise floor with any other method, you need to still cope with the issue of fractional number cycles.

The key tool in the situation is the Tabei & Ueda method which was already learned in a previous article.[1] By applying this method you can estimate accurate frequency, amplitude and phase of the target tone. Incidentally in this particular situation the tool will be deployed twice. Firstly by estimating the tone parameters, you can reconstruct the tone waveform approximately and then subtract the major signal to highlight the DC offset trend. Secondly you can accurately estimates the target signal by the tool for highlighting the residual noise.

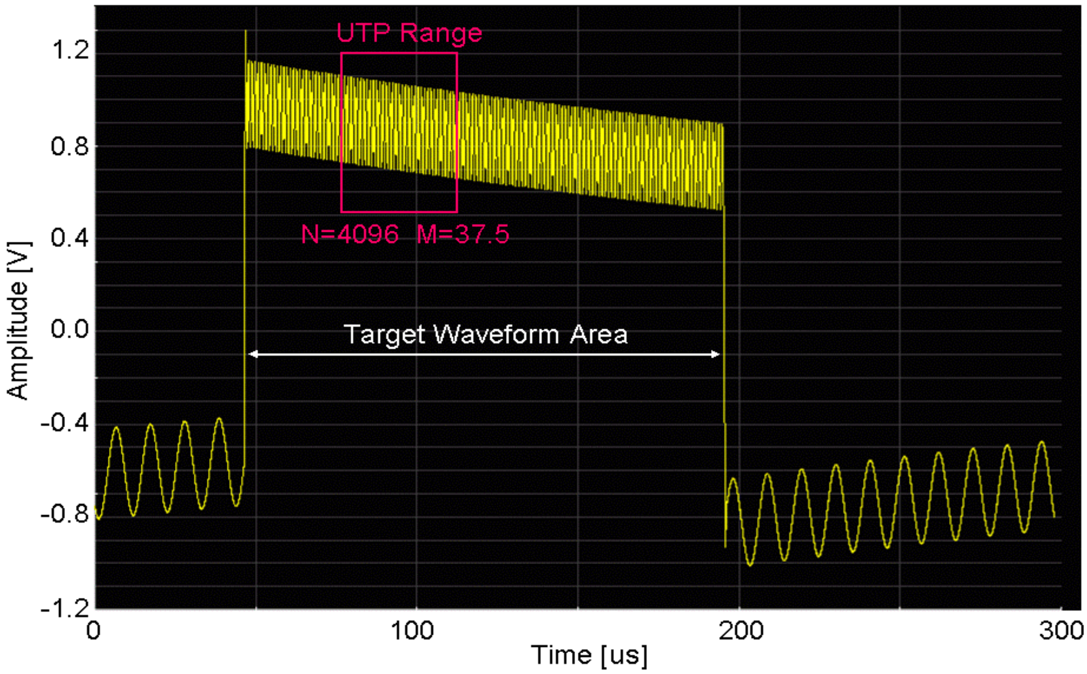

Figure 1: Waveform with DC Offset Drift

[1] Hideo Okawara's Lecture Series "DSP-Based Testing – Fundamentals 12 Spectrum Estimation"

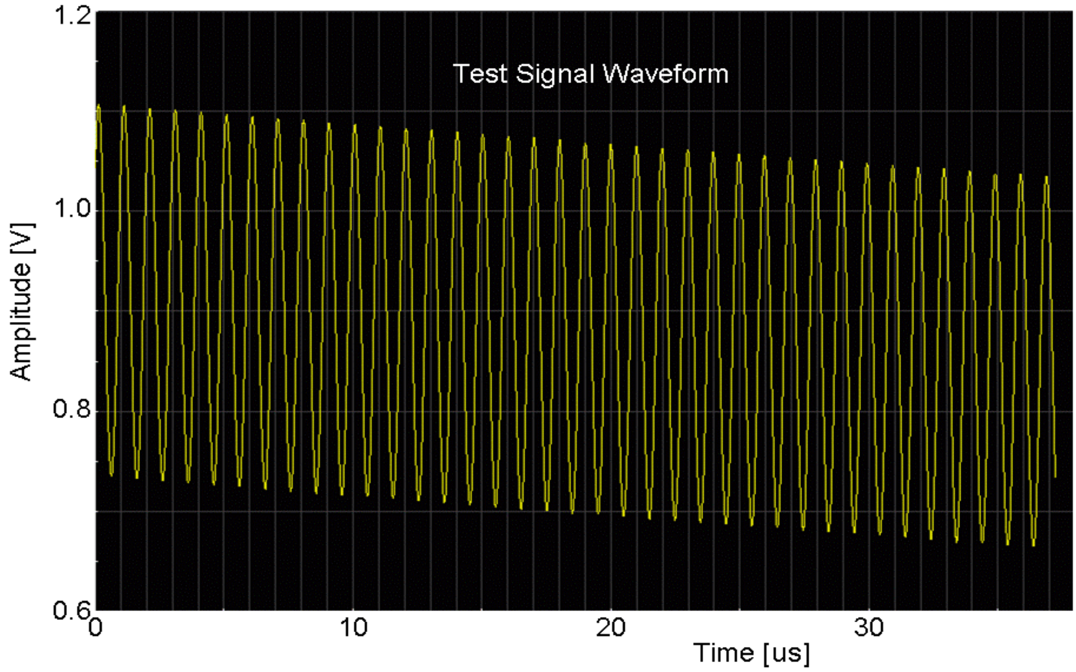

Figure 2: UTP Waveform (Fractional Number of Cycles: M=37.5)

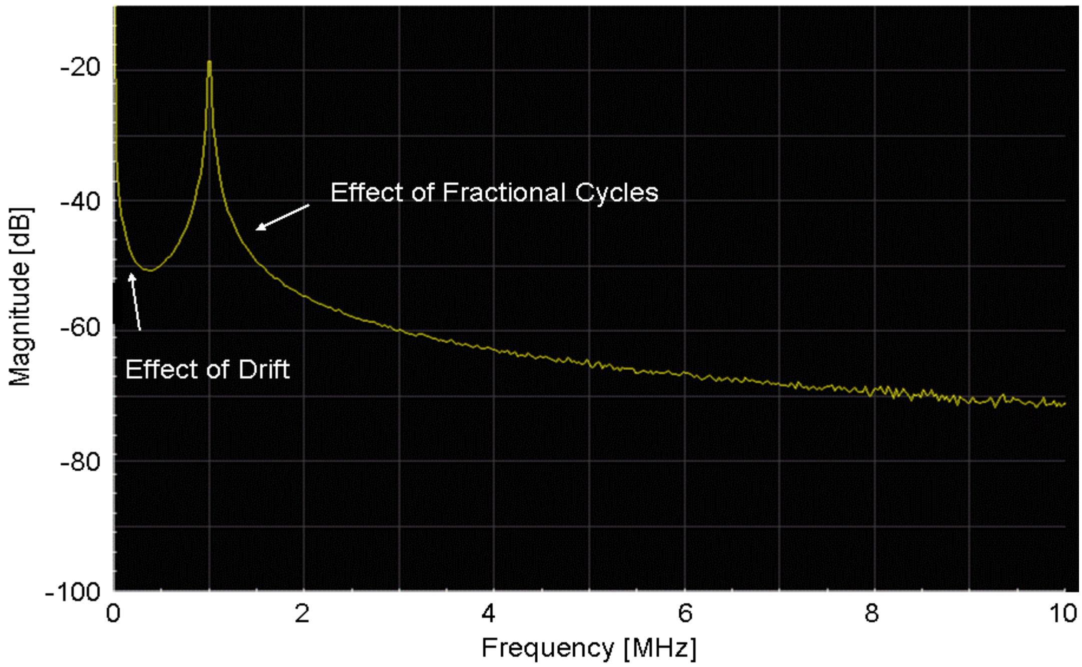

Figure 3: FFT Spectrum of Figure 2 Waveform with No Windowing

List 1 illustrates the procedure of DC trend removal. FFT with Hanning window at Line 11 and data processing up to Line 28 are the coding of Tabei & Ueda's interpolation method. By comparing the height of the tallest and the second tallest spectral lines, you can estimate the frequency (dFx), amplitude (dAx) and phase (dPx) of the target signal. This is the thrill of the method.

By using the estimated parameters of dFx, dAx and dPx as Lines 34 through 36, the signal tone can be reconstructed as illustrated as the red line in Figure 4, which is subtracted from the original yellow line signal in the figure at Line 37. Then the DC trend is close-up as the yellow line in Figure 5. A least square curve fit at Line 43 can well approximate the curve as the red line in Figure 5. The trend is approximated by the 2^{nd} order polynomial curve at Line 45.

```
INT           i,N,iMax;
DOUBLE        dMax,dR,dFx,dAx,dPx,dX,dY;
COMPLEX       CX,CY,CZ;
ARRAY_D       dVdata,dSp,dTemp;
ARRAY_COMPLEX CSp;

//  dVdata[] is the original waveform data
//  M=37.5 !!  Fractional number cycles of sine wave

//  Tabei & Ueda Estimation (1st for highlighting the trend)
DSP_FFT(dVdata,CSp,HANNING);                 // FFT with HANNING window
DSP_RECT_POL(CSp,dSp,dTemp);                 // Spectrum Magnitude
dMax=0.0;                                    // Tallest spectrum search
for (i=35;i<=40;i++) if (dSp[i]>dMax) { dMax=dSp[i]; iMax=i; }
if (dSp[iMax-1]>dSp[iMax+1]) {                   // 2nd Tallest Bin
   dR=dSp[iMax-1]/dSp[iMax];
   dFx=iMax+(1.0-2.0*dR)/(1.0+dR);               // Estimated Frequency
} else {
   dR=dSp[iMax+1]/dSp[iMax];
   dFx=iMax-(1.0-2.0*dR)/(1.0+dR);               // Estimated Frequency
}
dX=dFx-iMax;
dY=M_PI*dX;
dAx=-dSp[iMax]*(dY/sin(dY))*(dX-1.0)*(dX+1.0);// Estimated Amplitude
CX=CSp[iMax];
CY.real()=cos(dY); CY.imag()=-sin(dY);
CZ=CX*CY;
dPx=atan2(CZ.imag(),CZ.real());                  // Estimated Phase
INT     Nx;
DOUBLE  dP;
ARRAY_D dWave,dResidual,dXaxis,dCoef,dTrend,dVmodified;

dP=2.0*M_PI*dFx/N;
dWave.resize(N);                           // Estimated Signal Container
for (i=0;i<N;i++) dWave[i]=dAx*cos(dP*i+dPx);
DSP_SUB_VEC(dVdata,dWave,dResidual);          // Residual+DC Trend

dXaxis.resize(N);
for (i=0;i<N;i++) dXaxis[i]=(DOUBLE)i;        // X-axis Data
Nx=2;
dCoef.resize(Nx+1);                           // Coefficients Container
LeastSquareCurveFit(dXaxis,dResidual,dCoef,Nx); // Curve Fit Routine
dTrend.resize(N);                             // DC Trend Curve
for (i=0;i<N;i++) dTrend[i]=dCoef[0]+dCoef[1]*i+dCoef[2]*i*i;
DSP_SUB_VEC(dVdata,dTrend,dVmodified);        // Trend Removed
DSP_SPECTRUM(dVmodified,dSp,DB,1.0,RECT,0);// Trend Removed Spectrum

```

List 1: 1st Estimation for DC Trend Removal

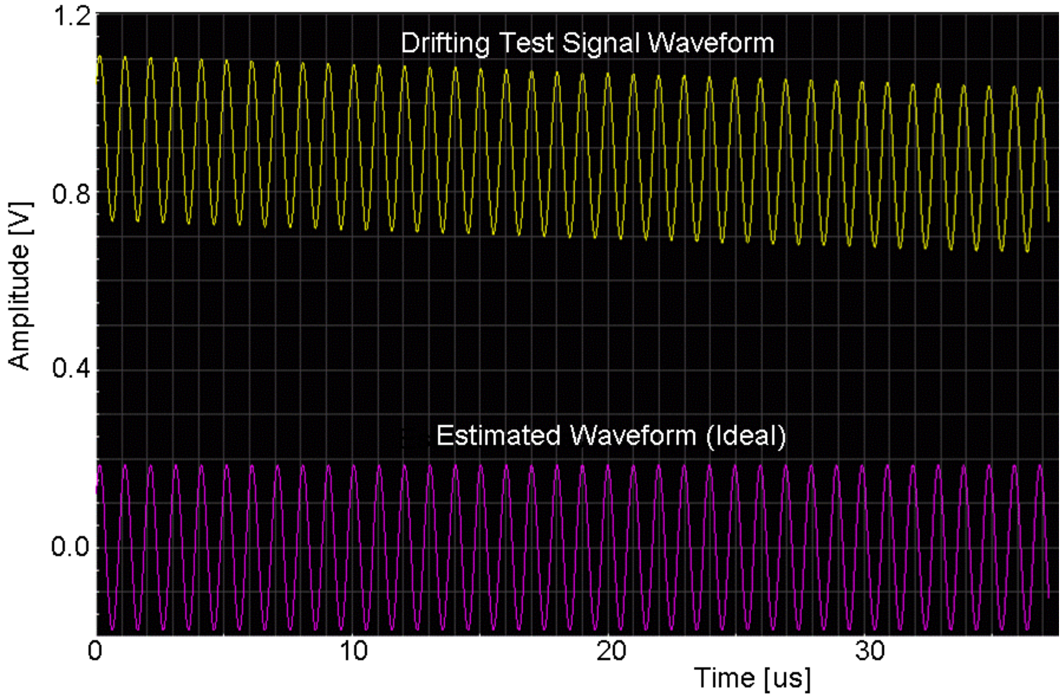

Figure 4: Original Waveform and Estimated Waveform (No DC)

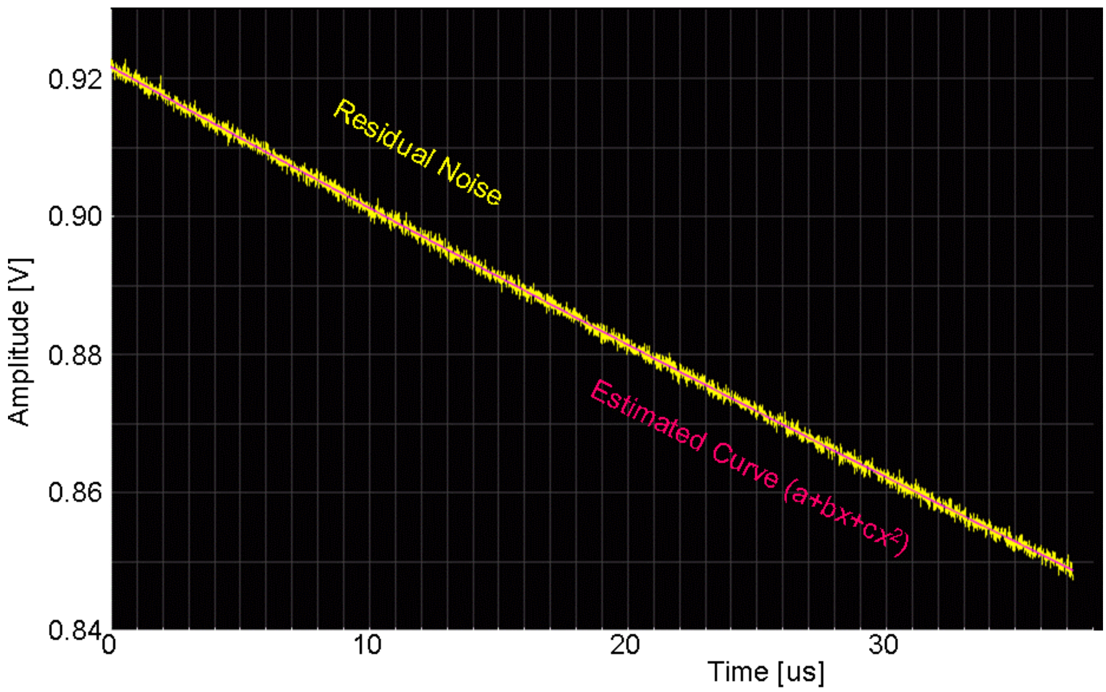

Figure 5: Residual Noise and Its Curve Fit Estimation

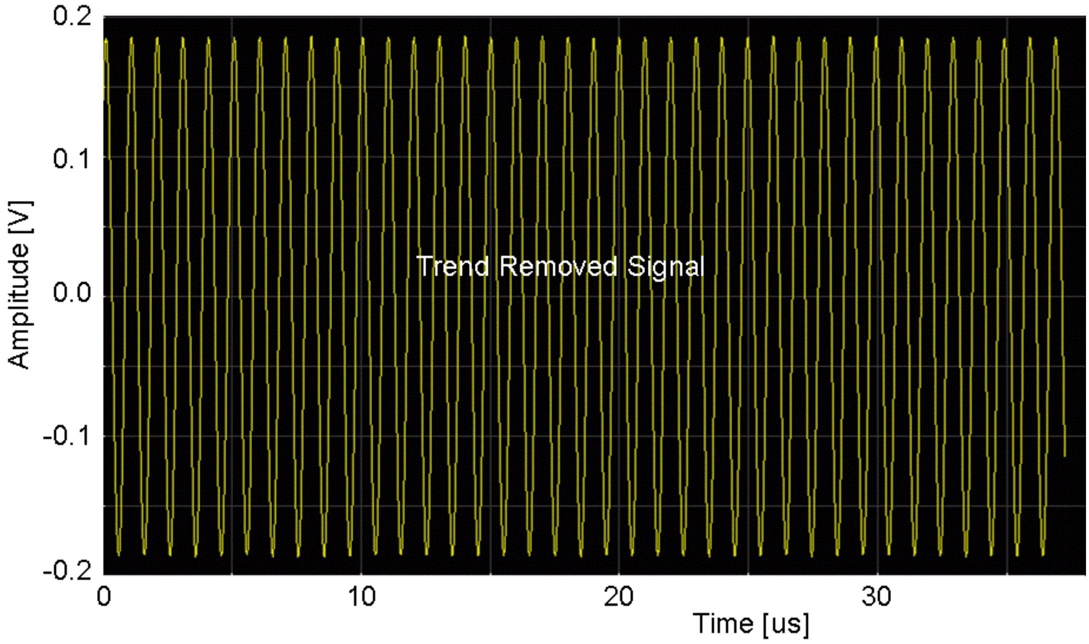

Figure 6: DC Trend Removed Waveform (After 1st Estimation)

Now that the approximated DC trend is subtracted from the original waveform, the target signal trend is compensated as Figure 6. Applying FFT to this waveform with no windowing, the frequency spectrum looks as Figure 7. Comparing to Figure 3, the DC trend problem is improved. However the fractional number cycles problem is still there. So the Tabei & Ueda method should be applied again to estimate the signal tone accurately. Lines 55 through 72 in List 2 are the 2nd operation of the method. dFx1, dAx1 and dPx1 are the estimated parameters, and then the accurate signal tone can be reconstructed as Lines 73 to 75. By subtracting the signal tone from the waveform in Figure 6, the residual noise is uncovered as Figure 8. The FFT frequency spectrum of the noise is illustrated in Figure 9. The estimated signal tone is overlaid in the figure as well. By collecting the noise power, you can calculate the S/N value with comparing the power of the estimated signal (dAx^2). You can figure it out from either the time domain data in Figure 8 or from the frequency domain data in Figure 9.

This is the way the DC trend can be removed in the fractional number cycle's situation, and the true noise floor can be uncovered.

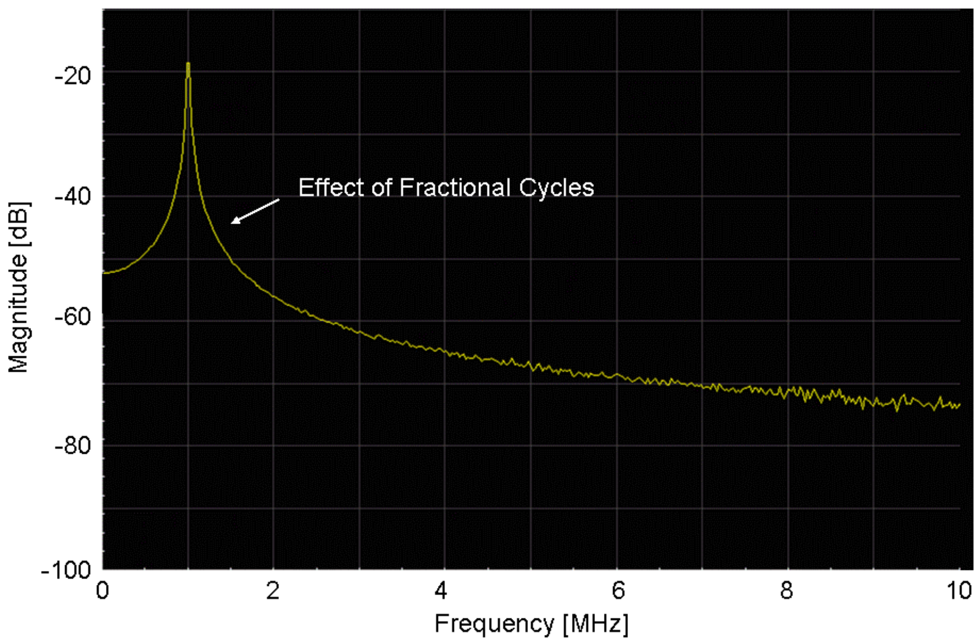

Figure 7: .... DC Trend Removed Spectrum (After 1st Estimation)

```
DOUBLE        dFx1,dAx1,dPx1;
ARRAY_D       dSp1,dWave1,dNoise,dSp2;
ARRAY_COMPLEX CSp1;

//  Tabei & Ueda Estimation (2nd EStimation for Accurate Tone)
DSP_FFT(dVmodified,CSp1,HANNING);        // FFT with HANNING window
DSP_RECT_POL(CSp1,dSp1,dTemp);           // Spectrum Magnitude
dMax=0.0;                                // Tallest spectrum search
for (i=35;i<=40;i++) if (dSp[i]>dMax) { dMax=dSp[i]; iMax=i; }
if (dSp1[iMax-1]>dSp1[iMax+1]) {         // 2nd Tallest Bin
   dR=dSp1[iMax-1]/dSp1[iMax];
   dFx1=iMax+(1.0-2.0*dR)/(1.0+dR);              // Estimated Frequenc
} else {
   dR=dSp1[iMax+1]/dSp1[iMax];
   dFx1=iMax-(1.0-2.0*dR)/(1.0+dR);              // Estimated Frequenc
}
dX=dFx1-iMax;
dY=M_PI*dX;
dAx1=-dSp1[iMax]*(dY/sin(dY))*(dX-1.0)*(dX+1.0);// Estimated Amplitud
CX=CSp1[iMax];
CY.real()=cos(dY); CY.imag()=-sin(dY);
CZ=CX*CY;
dPx1=atan2(CZ.imag(),CZ.real());                 // Estimated Phase
dP=2.0*M_PI*dFx1/N;
dWave1.resize(N);
for (i=0;i<N;i++) dWave1[i]=dAx1*cos(dP*i+dPx1);
DSP_SUB_VEC(dVmodified,dWave1,dNoise);   // Suppress signal.
DSP_SPECTRUM(dNoise,dSp2,DB,1.0,RECT,0); // Noise Only Spectrum
```

List 2: 2nd Estimation for Fractional Cycle Signal with DC Trend Removed

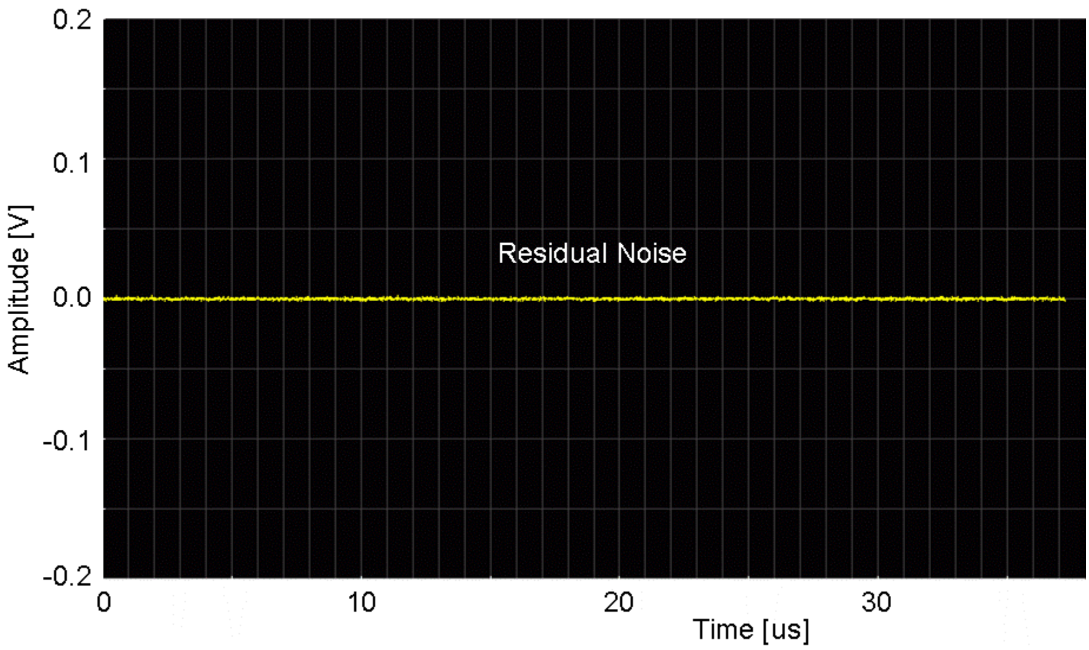

Figure 8: Trend Removed Residual Noise

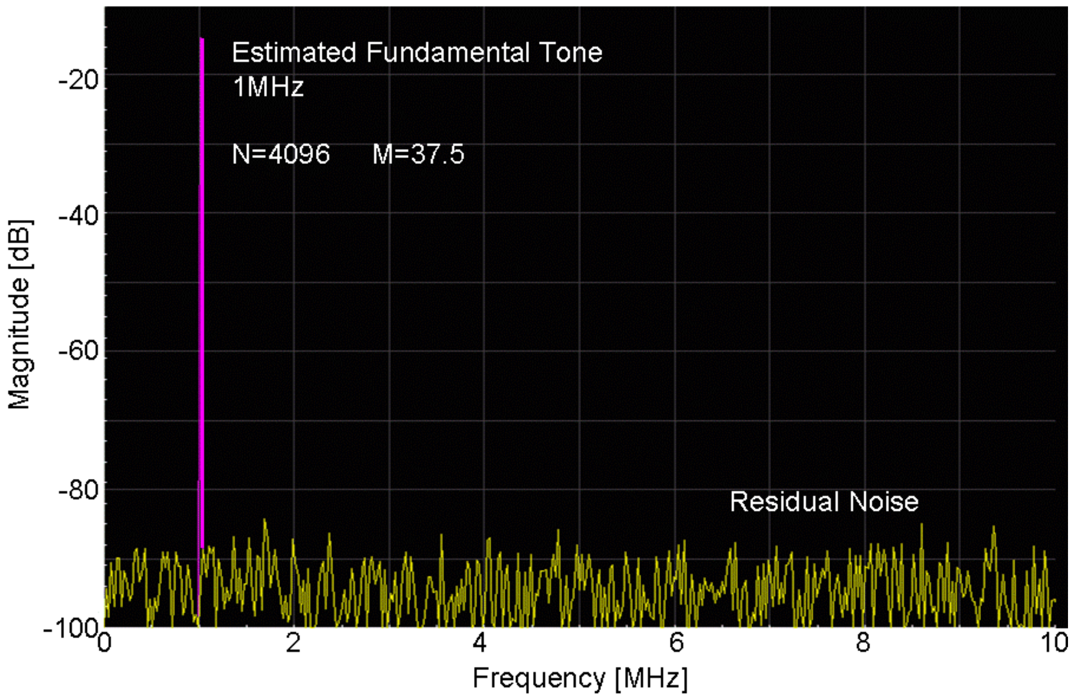

Figure 9: Trend Removed Residual Noise Spectrum and Estimated Signal

Appendix

For comprehension, Figures 3, 7 and 9 are combined into a single graph.

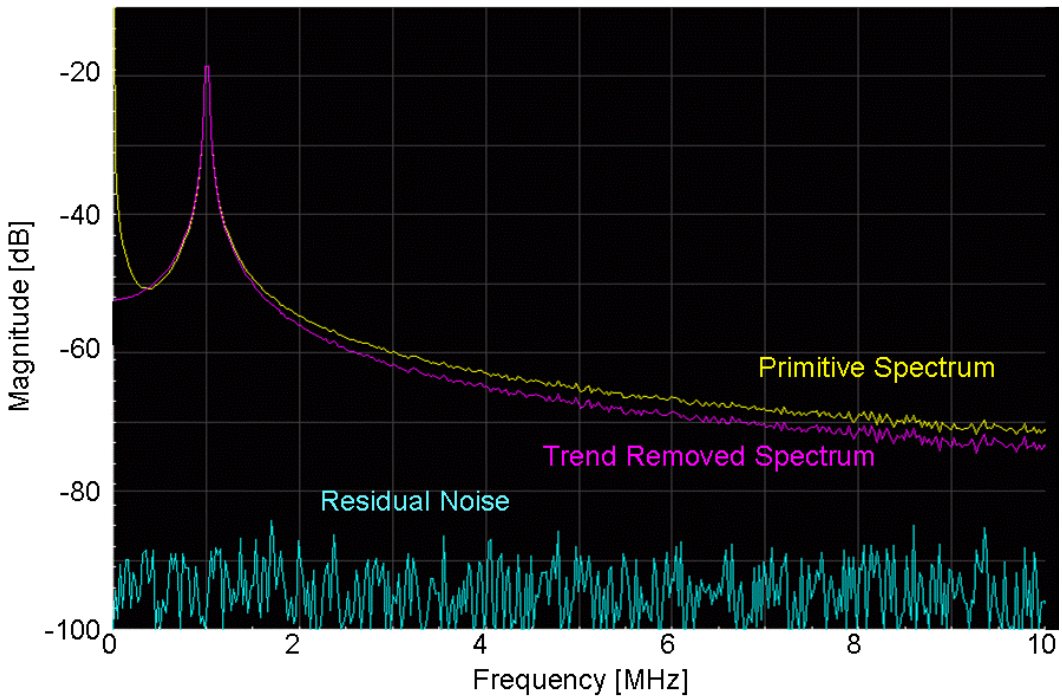

Figure 10: Combined Spectra Appearance