Reduce the peak-to-peak variation of the noise floor
- Pseudo Video Filter for the digitized data -

Akinori Maeda

Verigy Japan
June 2009

Introduction

This document introduces the technique to reduce the peak-to-peak variation of the noise floor of the FFT calculation results. This technique keeps the SNR value but reduces the peak-to-peak variation of the noise floor. Therefore, you can find the harmonics or the spurious spectrums that amplitudes are very close to the noise floor without losing noise floor characteristics. You can calculate the SNR or THD value from this result and the calculated SNR or THD value is same as the results calculated by the ordinary method.

This technique uses the “Frequency Domain Averaging”. It calculates the average of each spectrum bin of the plural FFT results.

The comparison results of this “Frequency Domain Averaging” method, the “Time Domain Averaging” method and the “All data FFT” method are shown in this document. The “Frequency Domain Averaging” and “All data FFT” methods keep the SNR value and do not miss the spurious spectrums. The “Frequency Domain Averaging” is faster calculation time compare to the “All data FFT” because the “All data FFT” needs to calculate the FFT with large number of data. The “Time Domain Averaging” method is the fastest calculation time but it does not keep the SNR value and has the possibility to miss the spurious spectrums.

Using the “Frequency Domain Averaging”, it reveals the small amplitude spurious or harmonics spectrums that are difficult to identify by the ordinary method.

The FFT calculation is widely used to evaluate the analog performance of the mixed signal devices. Especially, SNR, THD and SFDR are the key measurement parameters of the ADC (Analog to Digital Converter) and the DAC (Digital to Analog Converter). These parameters are tested at the production but the harmonics or the spurious spectrum characteristics are sometimes evaluated to understand the detail analog performance of the device. And also the spectrum characteristics are checked for the test program debugging.

The spectrum performance is given by the FFT calculation of the digitized data sampled by the Digitizer or the device itself (captured digital data). Evaluating the FFT results carefully, you can find the harmonics characteristics and the spurious characteristics of the device or the test environment. But sometimes the amplitude of these spectrums is small and they are buried in the noise floor. If this situation happens, it is difficult to identify that the spectrum is a part of the noise floor or a part of the distortion or spurious.

Also, you may need to check the noise floor shapes to check or debug your environment of the test. In normal case, it is difficult to identify the shape of the noise floor because the deviation of the amplitude of the noise floor is very large.

When you measure the spectrums by the spectrum analyzer, the "Video Filter" is the solution in this situation. Using the Video Filter, you can reduce the deviation of the peak-to-peak of the noise floor. Therefore, you can easily identify that the spectrum is a part of the noise floor, or a part of spurious or harmonics. And you can identify the shape of the noise floor.

This "Frequency Domain Averaging" reduces the peak-to-peak variation of the floor noise as like the "Video Filter" of the spectrum analyzer.

Solution

To reduce the peak-to-peak variation of the noise floor, the averaging method is used. This variation can be reduced by calculating the average of the frequency spectrum bins. To do this averaging, you must measure same waveforms plural times. Then you need to calculate the FFT of each waveform and average the frequency bins of the FFT results. At here, "averaging" means "Root Mean Square", not the simple average.

Figure 1 shows one example of this solution. The blue line shows the spectrums of the normal measured result, and the red line shows the averaged result of four FFT results. The SNR values calculated from these two spectrums are:

- Normal measurement (Blue Line): 43.71 dB
- Average results (Red Line): 43.80 dB

Both SNR results are almost same. This solution keeps the total noise value (SNR result), but it reduces the variation of the noise floor.

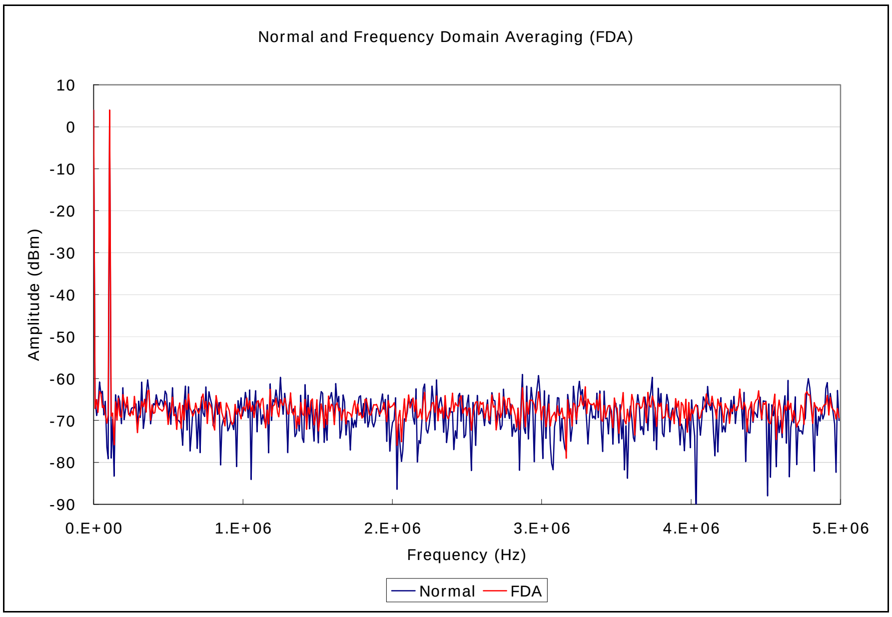

Figure 1: Effectiveness of Frequency Domain Averaging

The fundamental amplitude and the DC component by this averaging method are same as the normal measurement. The Table 1 shows the comparison of the amplitude of the some spectrums by this averaging method and the normal measurement.

Table 1: Amplitude of the spectrums

Frequency [Hz]	Normal [dBm]	FDA [dBm]	Comment
0.000E+00	**3.9710**	**3.9713**	**DC**
9.766E+03	-64.9337	-67.1164	
1.953E+04	-68.8213	-65.0104	
2.930E+04	-66.8377	-68.0678	
3.906E+04	-60.8267	-63.1064	
4.883E+04	-63.6996	-63.5751	
5.859E+04	-63.0335	-66.9060	
6.836E+04	-68.6831	-66.4491	
7.813E+04	-65.4295	-68.5964	
8.789E+04	-76.4195	-70.6577	
9.766E+04	-79.1314	-68.6716	
1.074E+05	**3.9794**	**3.9794**	**Fundamental**
1.172E+05	-79.0079	-68.9778	
1.270E+05	-69.7556	-68.2120	
1.367E+05	-83.2856	-75.7948	
1.465E+05	-63.8728	-65.5434	
1.563E+05	-67.5864	-69.9470	
1.660E+05	-64.1268	-64.7236	
1.758E+05	-65.4125	-68.8431	
1.855E+05	-70.7516	-69.2851	
1.953E+05	-62.1950	-63.8434	
2.051E+05	-69.8362	-65.3075	

At here, 'FDA' means the "Frequency Domain Averaging".

Detail of the Frequency Domain Averaging

As it is described previously, to calculate the "Frequency Domain Averaging", several same waveforms must be measured and FFT calculations must be performed for these waveforms.

We assume that the unit of the FFT results is 'voltage'. If the FFT results are in complex form, you need to convert to the amplitude form. If the unit of the FFT results is 'dBm', you need to convert to the unit of 'voltage'. Then calculate the square of each spectrum:

$$Spectrum\_power_{ij} = Spectrum\_data_{ij}{}^{2}$$

Where, 'Spectrum_data$_{ij}$' is the amplitude of the 'i'th spectrum bin of the 'j'th waveform and 'Spectrum_power$_{ij}$' is the squared result of the 'Spectrum_data$_{ij}$'.

Then calculate the mean of the each spectrum over the waveforms.

$$Spectrum\_average\_power_{i} = \frac{\sum_{j=1}^{n} Spectrum\_power_{ij}}{n}$$

The number of waveforms is 'n'. Lastly, calculate the square root to get the averaged spectrum.

$$Spectrum\_average\_data_{i} = \sqrt{Spectrum\_average\_power_{i}}$$

At here, the unit of the data is volt. But when you calculate the SNR and THD values or the draw the spectrums in dBm, you can calculate these values from 'Spectrum_average_power$_{i}$' easily. Therefore, 'Spectrum_average_data' may not be calculated in the actual application.

$$Noise\_power = \sum_{i=1}^{m-1} Spectrum\_average\_power_i$$

At here, 'm' is the total number of frequency bins. It does not include the DC component, 'i=0'.

$$Harmonic\_power = \sum_{i=2}^{x} Spectrum\_average\_power_{fundamental \times i}$$

Where, 'x' is the order number of harmonics you want to sum. '$Spectrum\_average\_power_{fundamental}$' is the spectrum power of the fundamental signal. Therefore, '$Spectrum\_average\_power_{fundamental \times i}$' is the spectrum power of the 'i'th harmonic.

The SNR and THD values are calculated as:

$$SNR = 10 \times \log\left(\frac{Spectrum\_average\_power_{fundamental}}{Noise\_power - Spectrum\_average\_power_{fundamental}}\right)$$

$$THD = 10 \times \log\left(\frac{Harmonic\_power}{Spectrum\_average\_power_{fundamental}}\right)$$

Implementation Example

The example C++ implementation of the "Frequency Domain Averaging" is shown in below. This example is created under the Verigy V93000 SOC Test System Environment (SmarTest). The measured data is stored in the ARRAY_D variable 'measured_data'. In this example, 'measured_data' is the data from the Digitizer but you can apply this calculation to the data from Digital Capture.

```
ARRAY_D  measured_data;     // measured raw data
ARRAY_D  spectrum_data;     // final spectrum data
ARRAY_D  extracted_measured_data;     // extracted temporally data from 'measured_data'
ARRAY_D  extracted_spectrum_data;     // extracted temporally data of FFT results
INT  number_of_data;     // number of data in 'measured_data'
INT  number_of_sample_in_one_set;     // number of data in one set of measured data
INT number_of_average;     // number of average. Number of sets in 'measured_data'
// number_of_data = number_of_sample_in_one_set * number_of_average
INT  i, j;

spectrum_data.resize(number_of_sample_in_one_set/2);
extracted_spectrum_data.resize(number_of_sample_in_one_set/2);

extracted_measured_data = ARRAY_D(measured_data, number_of_sample_in_one_set, 0);
// extract first one set of the data
DSP_SPECTRUM(extracted_measured_data, spectrum_data, PWR, 1.0, RECT, 0);
// Calculate the FFT. Reference is 1.0V and Vpeak^2 is the unit of 'spectrum_data'

for (i = 1; i < number_of_average;  i++) {
     extracted_measured_data = ARRAY_D(measured_data, number_of_sample_in_one_set,
number_of_sample_in_one_set*i);
// extract next one set of the data
DSP_SPECTRUM(extracted_measured_data, extracted_spectrum_data, PWR, 1.0, RECT, 0);
// Calculate the FFT. Reference is 1.0V and Vpeak^2 is the unit of 'spectrum_data'
     for (j = 0; j < number_of_sample_in_one_set/2; j++) {
          spectrum_data[j]  += extracted_spectrum_data[j];
     }
}

// get the final results
for (i = 0; i < number_of_sample_in_one_set/2; i++) {
     spectrum_data[i] /= number_of_average;
     spectrum_data[i] = sqrt(spectrum_data[i]);
}
```

If you need to calculate the SNR or the THD, below C++ codes should be used.

```
ARRAY_D  measured_data;     // measured raw data
ARRAY_D  spectrum_data;     // final spectrum data
ARRAY_D  extracted_measured_data;     // extracted temporally data from 'measured_data'
```

```
ARRAY_D  extracted_spectrum_data;     // extracted temporally data of FFT results
INT  number_of_data;     // number of data in 'measured_data'
INT  number_of_sample_in_one_set;     // number of data in one set of measured data
INT number_of_average;     // number of average. Number of sets in 'measured_data'
// number_of_data = number_of_sample_in_one_set * number_of_average
INT  i, j;
DOUBLE  total_power;
DOUBLE signal_power;
DOUBLE harmonic_power;
DOUBLE snr;
DOUBLE thd;
INT  number_of_periods_in_one_set;
INT  number_of_harmonics_to_calculate;

spectrum_data.resize(number_of_sample_in_one_set/2);
extracted_spectrum_data.resize(number_of_sample_in_one_set/2);
number_of_periods_in_one_set = …;
number_of_harmonics_to_calculate = …;

extracted_measured_data = ARRAY_D(measured_data, number_of_sample_in_one_set, 0);
// extract first one set of the data
DSP_SPECTRUM(extracted_measured_data, spectrum_data, PWR, 1.0, RECT, 0);
// Calculate the FFT. Reference is 1.0V and Vpeak^2 is the unit of 'spectrum_data'

for (i = 1; i < number_of_average;  i++) {
     extracted_measured_data = ARRAY_D(measured_data, number_of_sample_in_one_set,
number_of_sample_in_one_set*i);
// extract next one set of the data
DSP_SPECTRUM(extracted_measured_data, extracted_spectrum_data, PWR, 1.0, RECT, 0);
// Calculate the FFT. Reference is 1.0V and Vpeak^2 is the unit of 'spectrum_data'
     for (j = 0; j < number_of_sample_in_one_set/2; j++) {
          spectrum_data[j]  += extracted_spectrum_data[j];
     }
}

// calculate the THD power;
signal_power = spectrum_data[number_of_periods_in_one_set] / num_of_average;
thd_power = 0.0;
for (i = 2; i < number_of_harmonics_to_calculate; i++) {
     // upper harmonics may be image spectrum
     j = number_of_periods_in_one_set*i;
     if (j > number_of_sample_in_one_set/2) {
          j = number_of_sample_in_one_set - j;
     }
     thd_power += spectrum_data[j];
}

// calculate THD results
thd = 10.0 * log10(thd_power / number_of_average / signal_power);

// get the final results and calculate the total power
total_power = 0.0;
for (i = 0; i < number_of_sample_in_one_set/2; i++) {
     spectrum_data[i] /= number_of_average;
     total_power += spectrum_data[i];
     spectrum_data[i] = sqrt(spectrum_data[i]);
}

// calculate SNR results
snr = 10.0 * log10(signal_power / (total_power - signal_power - spectrum_data[0]));
```

Comparison to other average methods

The "Frequency Domain Averaging" method requires measuring same waveforms plural times. Using these measured results, two other averaging methods can be considered. One is the "Time Domain Averaging" and another is the "All data FFT".

Time domain averaging

In the "Time Domain Averaging" method, it calculates the average of the each time-domain waveform data, and then calculates FFT using the averaged time domain data.

Below figures and table show the comparison of the "Time Domain Averaging" method and the "Frequency Domain Averaging" method. In Table 2, 'NPD' is the Noise Power Density, 'Normal' shows the results using the single waveform data and this is the results of the ordinary method, 'FDA shows the results of the "Frequency Domain Averaging" method and 'TDA' shows the results of the "Time Domain Averaging" method.

From these figures and table, you can understand that the "Time Domain Averaging" does not keep the original noise information. The "Time Domain Averaging" method averages the noise itself and the noise reduces to 1/sqrt(number_of_average). In below figures and table, 4 times of averaging is performed and the "Time Domain Averaging" reduces the noise about a half (-6 dB).

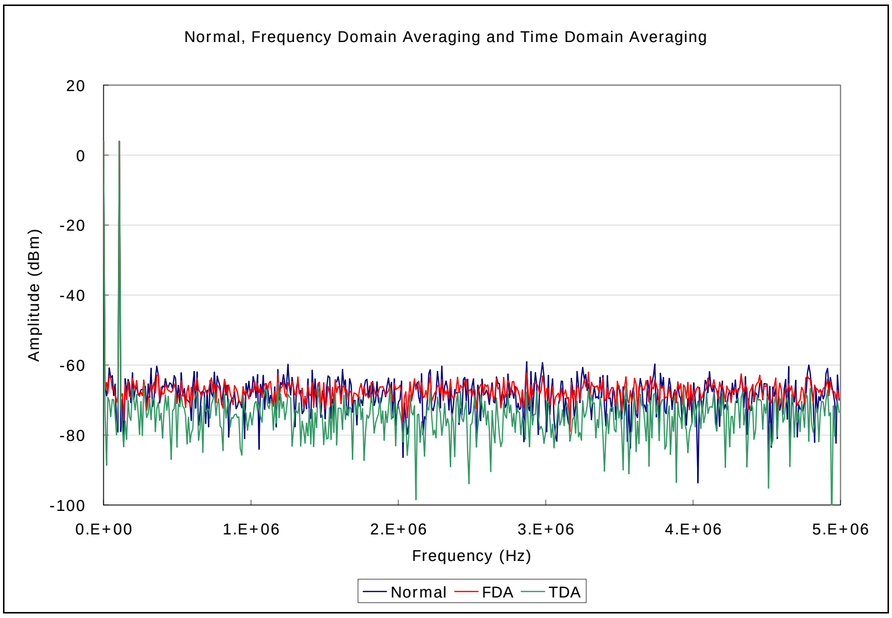

Figure 2: Normal, Frequency Domain Averaging and Time Domain Averaging

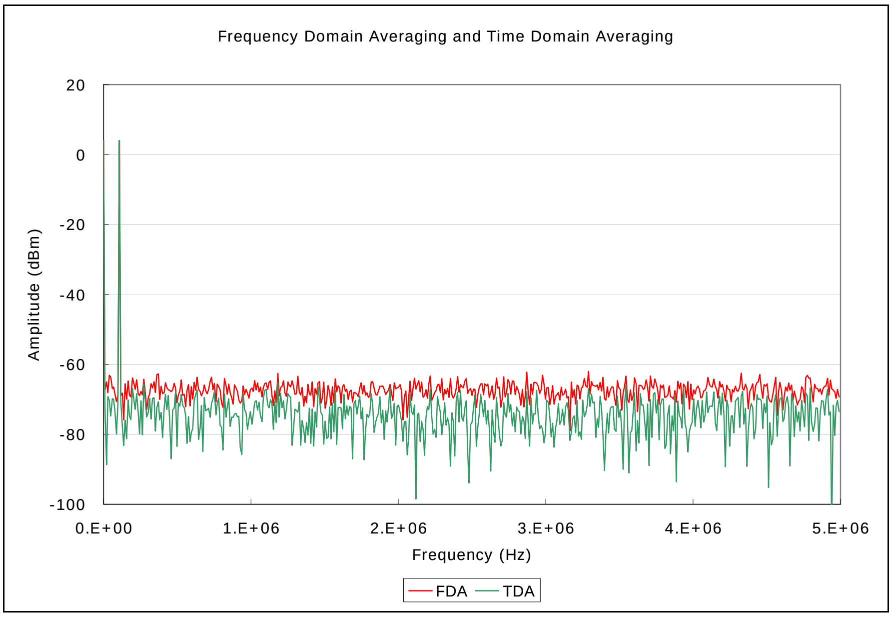

Figure 3: Frequency Domain Averaging and Time Domain Averaging

Table 2: Comparison of SNR and NPD

	Normal	FDA	TDA	
SNR	43.71	43.80	49.54	dB
NPD	-106.71	-106.81	-112.54	dBm/Hz

All data FFT

When you measure the waveform and get the measured data, all of the data that includes plural waveforms are located in one single array variable in most of the cases. If the FFT is calculated once for this array variable that includes plural same waveforms, the FFT result may be same as the result of the "Frequency Domain Averaging" method. If this is true, this way reduces the complexity of the calculation of the "Frequency Domain Averaging." This method is called "All data FFT" method.

Below figures and table show the comparison of the "All data FFT" method and the "Frequency Domain Averaging" method. In Table 3 'NPD' is the Noise Power Density, 'Normal' shows the results using the single waveform data and this is the results of the ordinary method, 'FDA' shows the results of the "Frequency Domain Averaging" method and 'ADF' shows the results of the "All data FFT" method.

From these figures and table, you can understand that the "All data FFT" method keeps the original noise information. The "All data FFT" method gives the same SNR value as the "Frequency Domain Averaging" method. If you only need the SNR or NPD value, the "All data FFT" method and the "Frequency Domain Averaging" method are same.

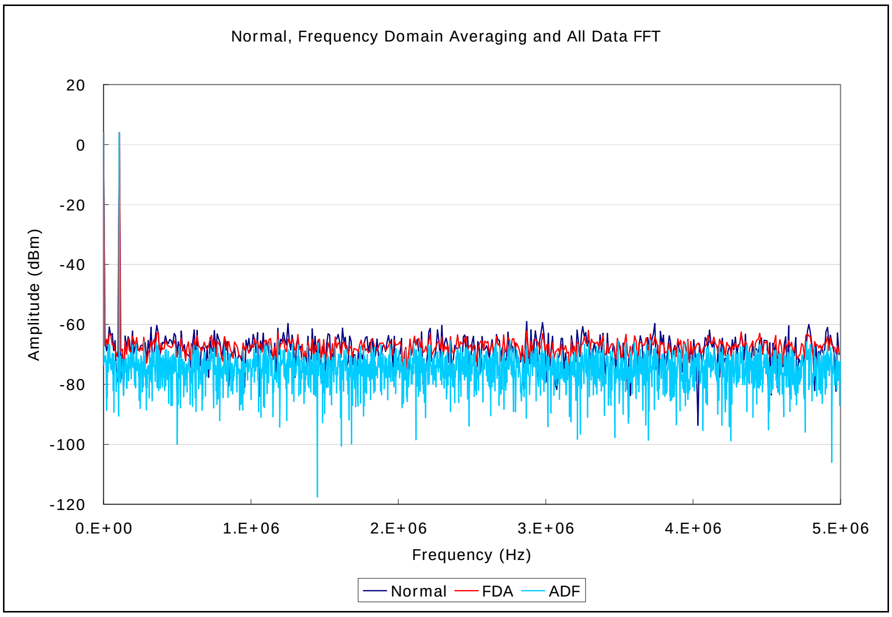

Figure 4: Normal, Frequency Domain Averaging and All data FFT

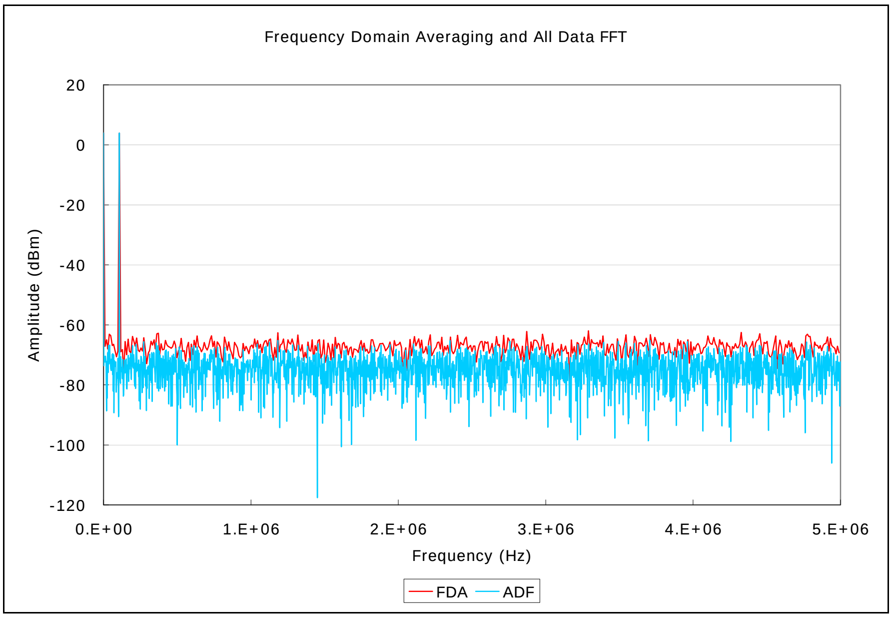

Figure 5: Frequency Domain Averaging and All data FFT

Table 3: Comparison of SNR and NPD

	Normal	FDA	ADF	
SNR	43.71	43.80	43.79	dB
NPD	-106.71	-106.81	-106.80	dBm/Hz

The relation of "All data FFT", "Time Domain Averaging"

Based on the precise checking of the each data value of above examples, the "Time Domain Averaging" is the results of picking up the every fourth data of "All data FFT". Every forth data of "All data FFT" and every data of "Time Domain Averaging" are same value.

Therefore, the "Time Domain Averaging" method does not show the spectrums that located at the between picked up data points. If the harmonics or the spurious spectrum exists between the picked up data point, the "Time Domain Averaging" method ignores this harmonics or the spurious spectrum.

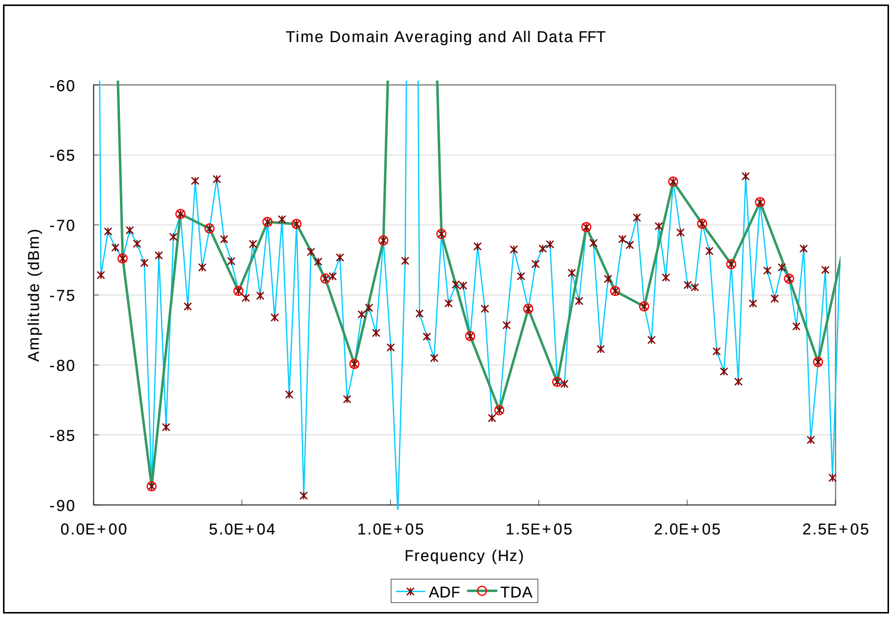

Figure 6: Time Domain Averaging and All data FFT

Spurious

If the spectrum includes the spurious, how do all above three averaging methods show the spurious? Below shows the simulation results, in that three averaging methods are applied to the signals including the spurious. In this simulation, three spurious are added to the fundamental sine waveform. These spurious are:

- 0.075% of full scale, -62.5dBfs, 0.3125 of the sampling frequency.
- 0.075% of full scale, -62.5dBfs, 0.15625 of the sampling frequency
- 0.05% of full scale, -66.0dBfs, 0.0048843125 of the sampling frequency

In this simulation, four plural waveforms are used. Therefore, the number of average of the "Frequency Domain Averaging" and "Time Domain Averaging" methods is 4. Below figures show the results of the spectrum of each method.

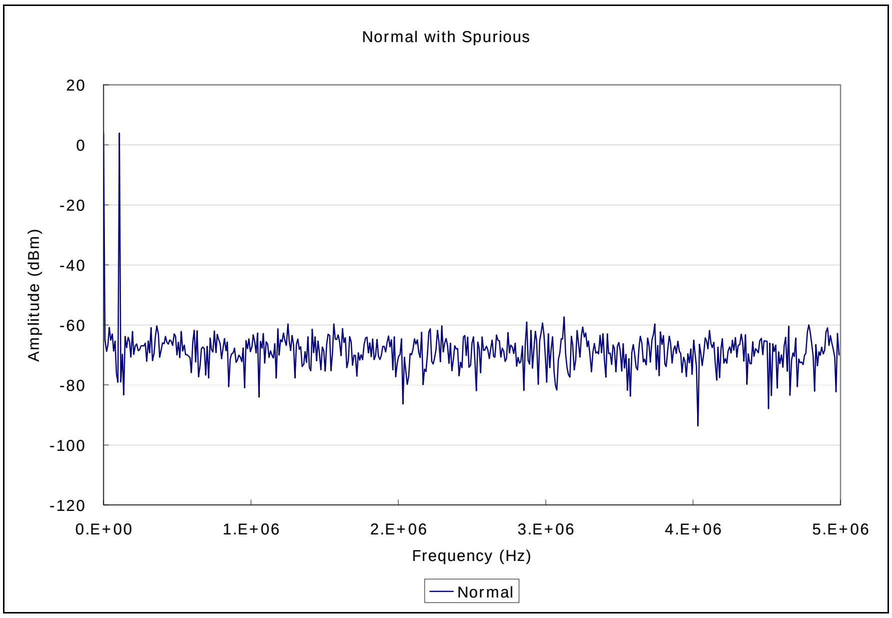

Figure 7: Normal with spurious

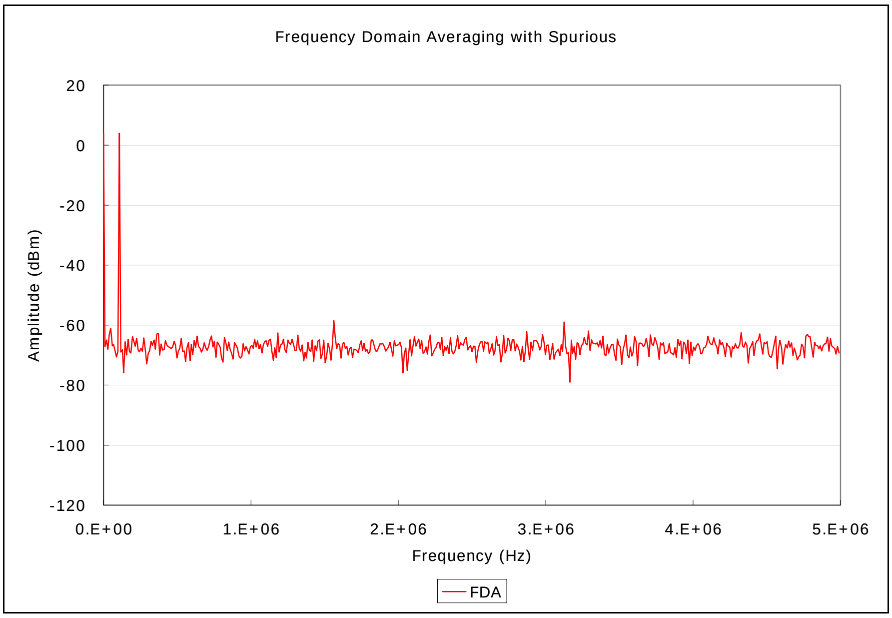

Figure 8: Frequency Domain Averaging with spurious

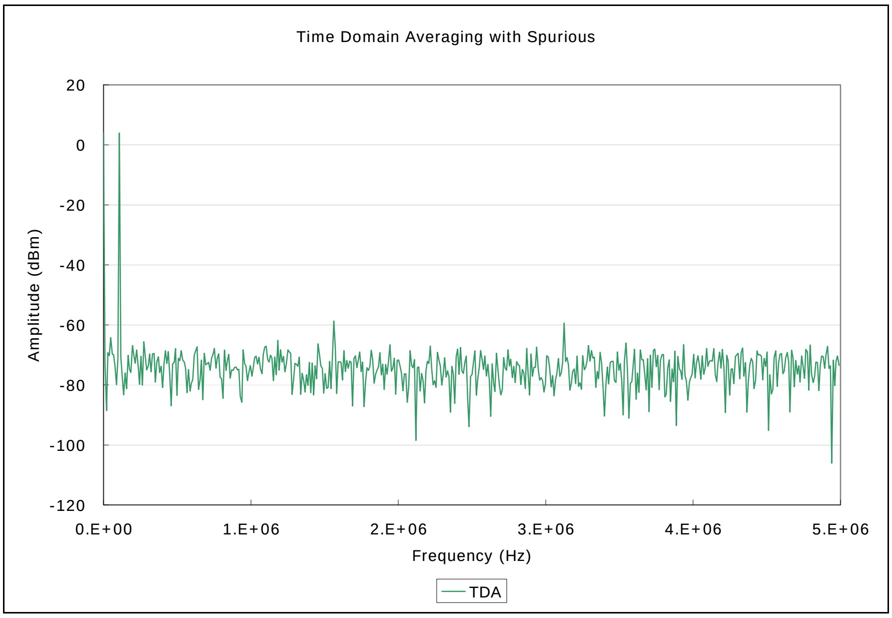

Figure 9: Time Domain Averaging with spurious

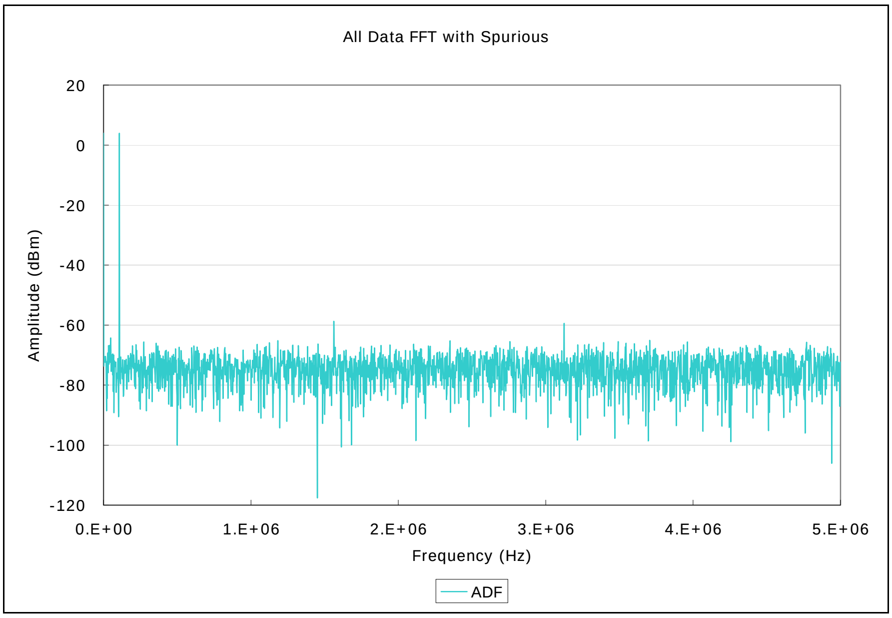

Figure 10: All data FFT with spectrum

A Figure 7 shows the spectrum of the FFT results of the normal single waveform. It is hard to identify the spurious signals. All of the spurious signals are buried in the noise floor. But all of three averaging methods clearly show the spurious signals.

Below figures and table shows the comparison of three averaging methods.

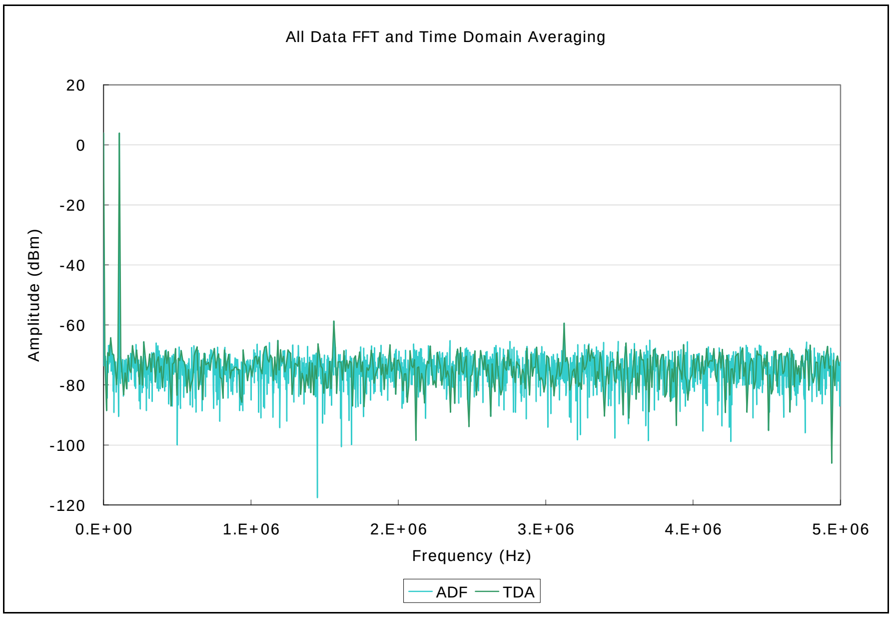

Figure 11: All data FFT and Time Domain Averaging

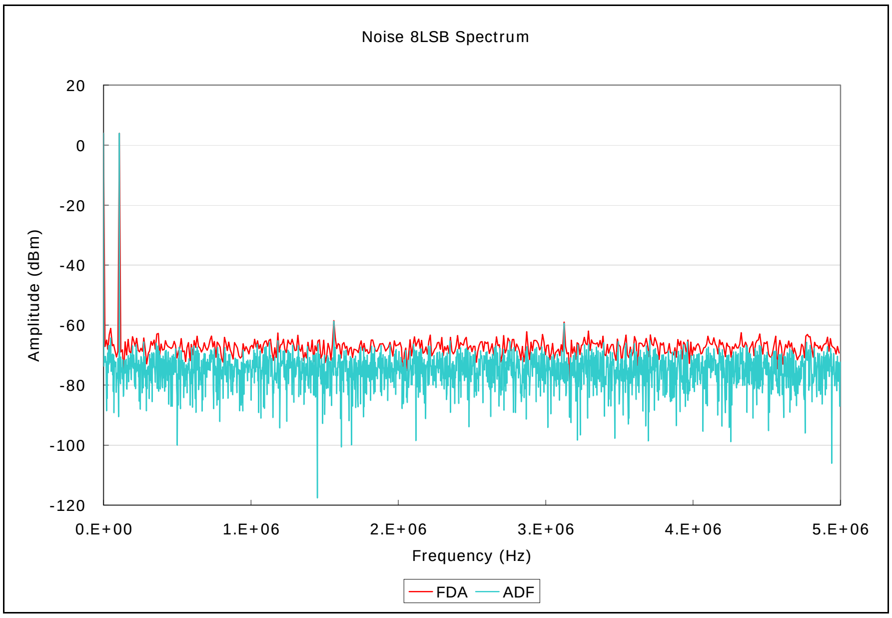

Figure 12: All data FFT and Frequency Domain Averaging

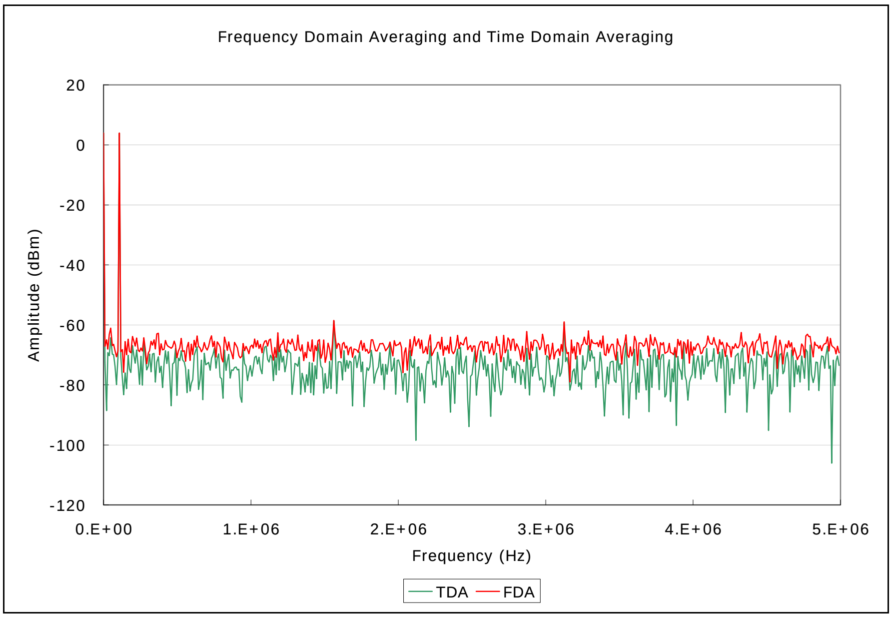

Figure 13: Time Domain Averaging and Frequency Domain Averaging

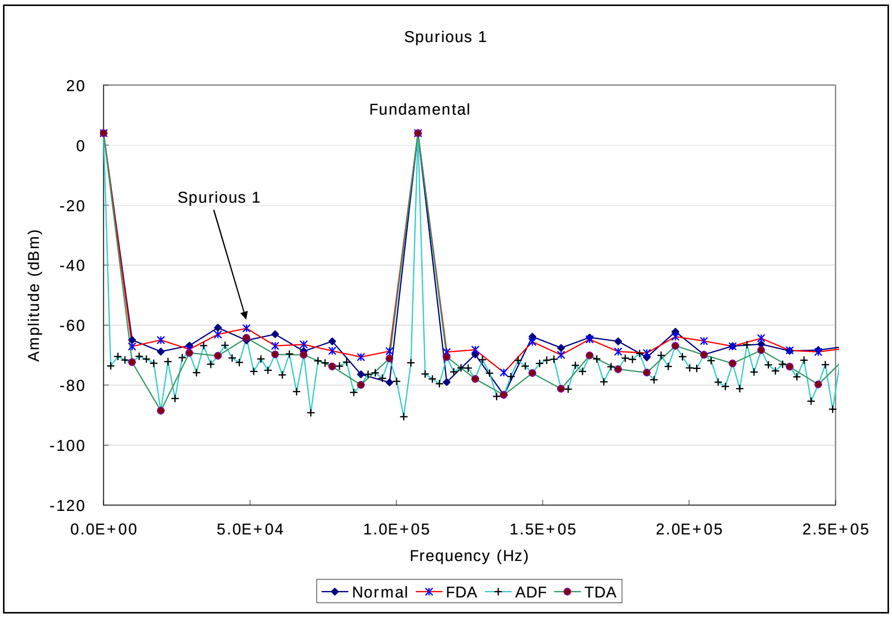

Figure 14: Spectrum around Spurious1

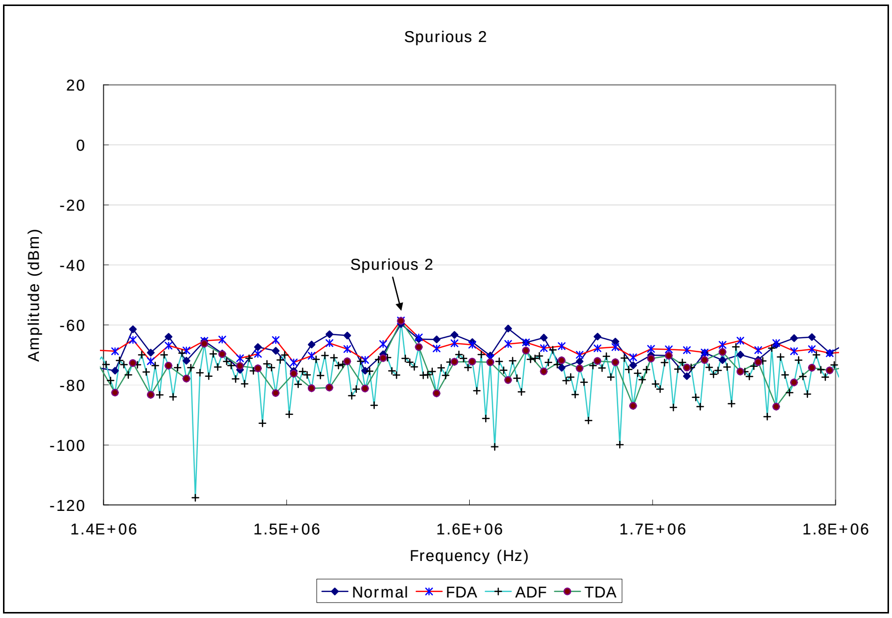

Figure 15: Spectrum around Spurious2

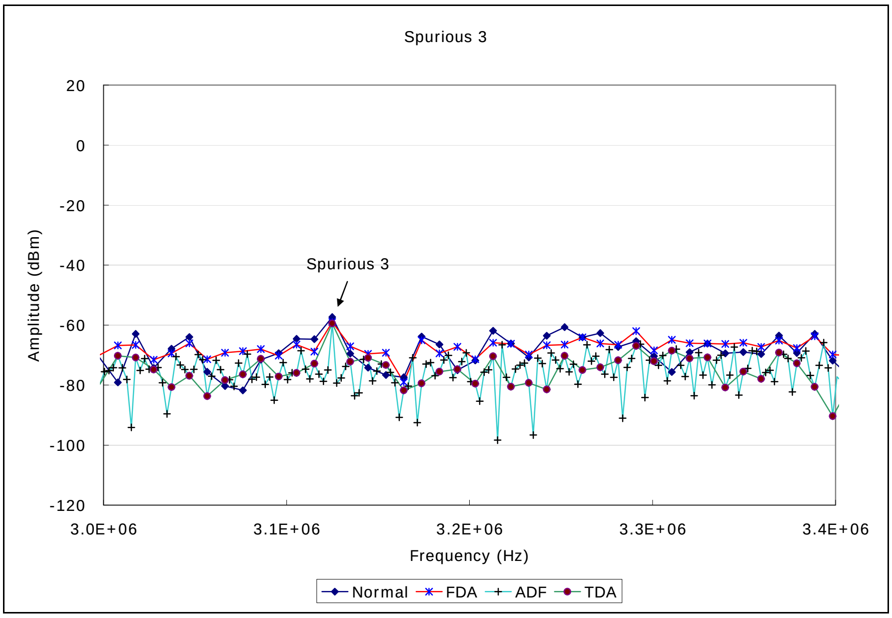

Figure 16: Spectrum around Spurious3

Table 4: Comparison of all of three averaging methods

	ADF	Normal	FDA	TDS	
Fundamental	3.98	3.98	3.98	3.98	dBm
Spurious 1	-64.26	-65.12	-61.06	-64.26	dBm
Spurious 2	-58.72	-59.70	-58.50	-58.72	dBm
Spurious 1	-59.46	-57.30	-59.00	-59.46	dBm
NPD	-106.69	-106.60	-106.69	-112.12	dBm/Hz
SNR	43.68	43.60	43.69	49.12	dB

As it is described at the section “All data FFT”, the amplitude of the spurious is same between the “All data FFT” and the “Time Domain Averaging”. The spurious amplitude is not exactly same among these averaging methods and original spectrum, but similar spurious amplitude is obtained. These averaging methods are effective to detect the spurious even if it is difficult to identify by the ordinary method.

Position of the spurious

In above simulation, the frequency of the spurious is exactly same as the frequency of the spectrum bin. But in the real world, the spurious may exist at any frequency. Below figures show the simulation results when the spurious exists at the different frequency from the spectrum bin.

In this simulation, the number of average is fixed, eight times and the position of the spurious is changed 1/8 frequency of the spacing between frequency bins. The frequency of the spurious is changed from 3.125 MHz (bin 320) to 3.134765625 MHz (bin 321) in 0.001220703125 MHz step. This is the 1/8 of the frequency difference (0.009765625 MHz) between bin 320 and bin 321.

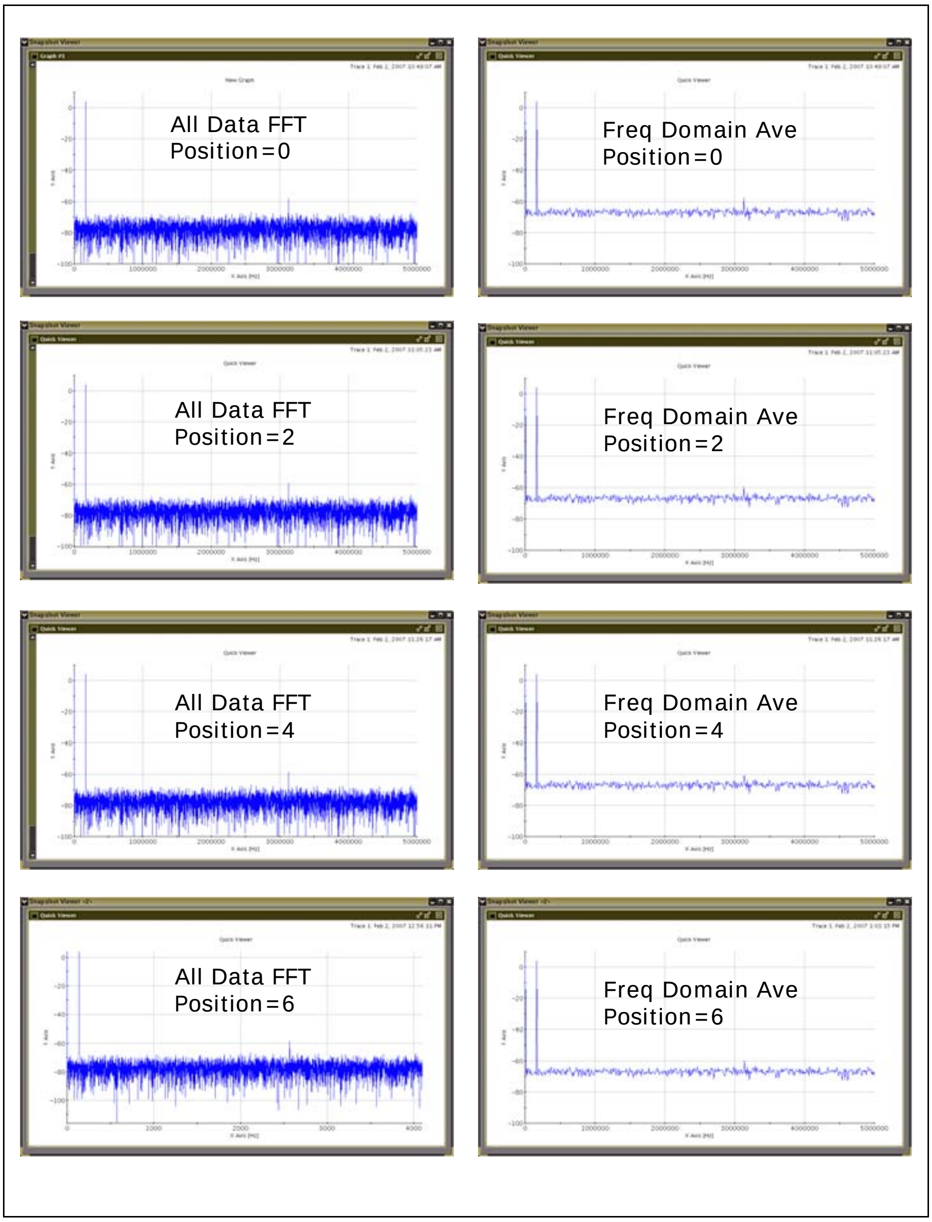

Figure 17: Frequency Domain Averaging. Position of the spurious

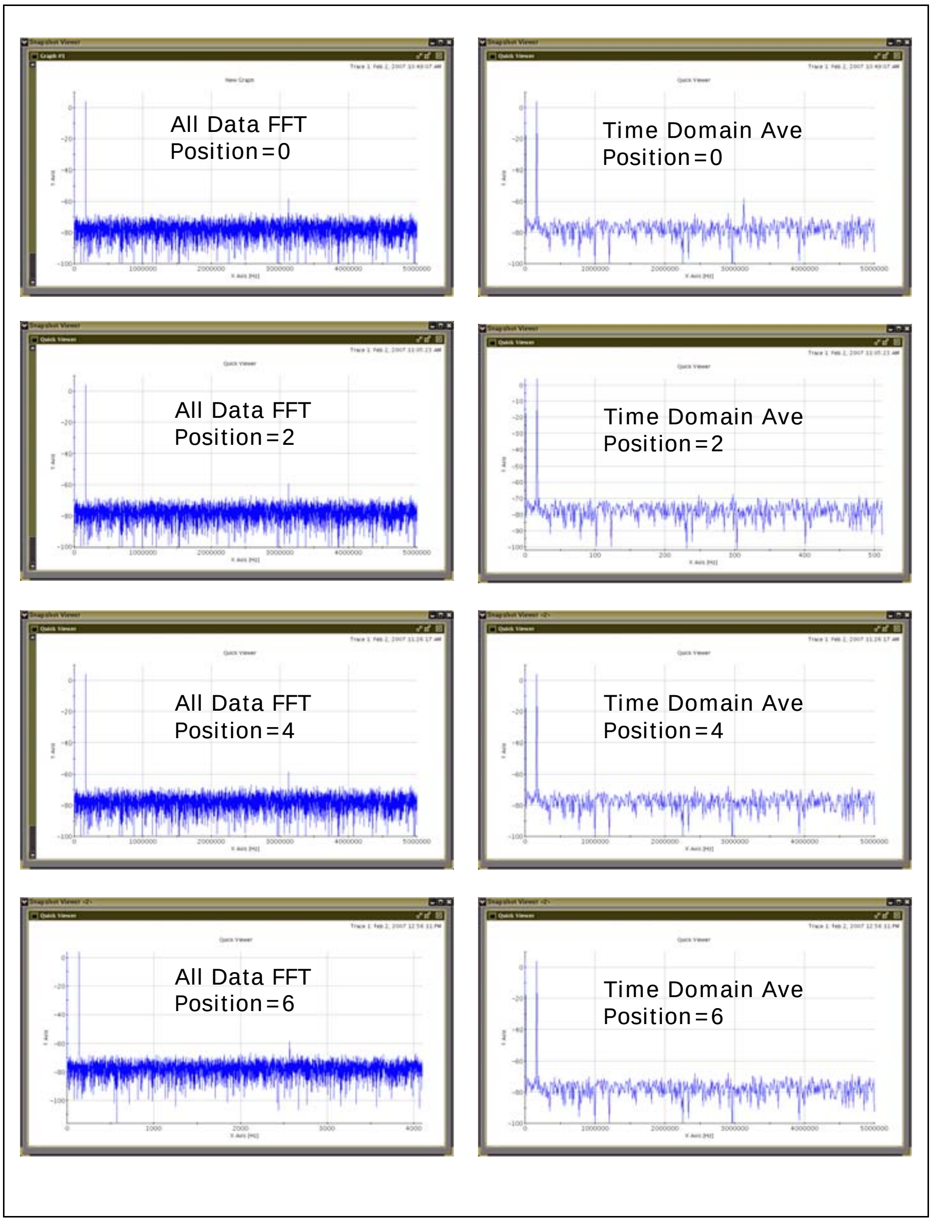

Figure 18: Time Domain Averaging. Position of the spurious

Below table shows the summary of the simulation.

Table 5: Position of the spurious

	Frequency Domain Average		Time Domain Average		All Data FFT	
Spurious Position	Frequency bin 320	Frequency bin 321	Frequency bin 320	Frequency bin 321	Spurious	Unit
320.0	-57.50	-66.69	-58.02	-76.31	-58.02	dBm
320.1	-57.63	-66.48	-81.43	-76.31	-58.04	dBm
320.2	-59.30	-64.94	-81.43	-76.31	-59.40	dBm
320.3	-60.92	-63.05	-81.43	-76.31	-59.44	dBm
320.4	-65.66	-61.18	-81.43	-76.31	-58.41	dBm
320.5	-64.74	-58.47	-81.43	-76.31	-58.25	dBm
320.6	-65.73	-58.52	-81.43	-76.31	-58.49	dBm
320.7	-65.63	-59.69	-81.43	-76.31	-59.50	dBm
321.0	-66.78	-57.39	-81.43	-57.87	-57.87	dBm

As it is described at the section "All data FFT", the "Time Domain Averaging" is the results of picking up the every 'n'th data of "All data FFT". Therefore, if the frequency of the spurious is not on the frequency bin, the "Time Domain Averaging" can not show the spurious spectrum. This "Time Domain Averaging" method can not be used if the spurious spectrum is important for the parameter of the test. If the parameter of the test does not care about the spurious signal, like THD measurement, this averaging method can be sued.

On the other hand, the "Frequency Domain Averaging" can show the spurious even if the frequency of the spurious is not same as the frequency of the bin. This averaging method can be used for any parameter of the test.

Number of average

Setting the large number of averaging, you can get less variation of the floor noise.

Below figures show the spectrum of the simulation results changing the number of average.

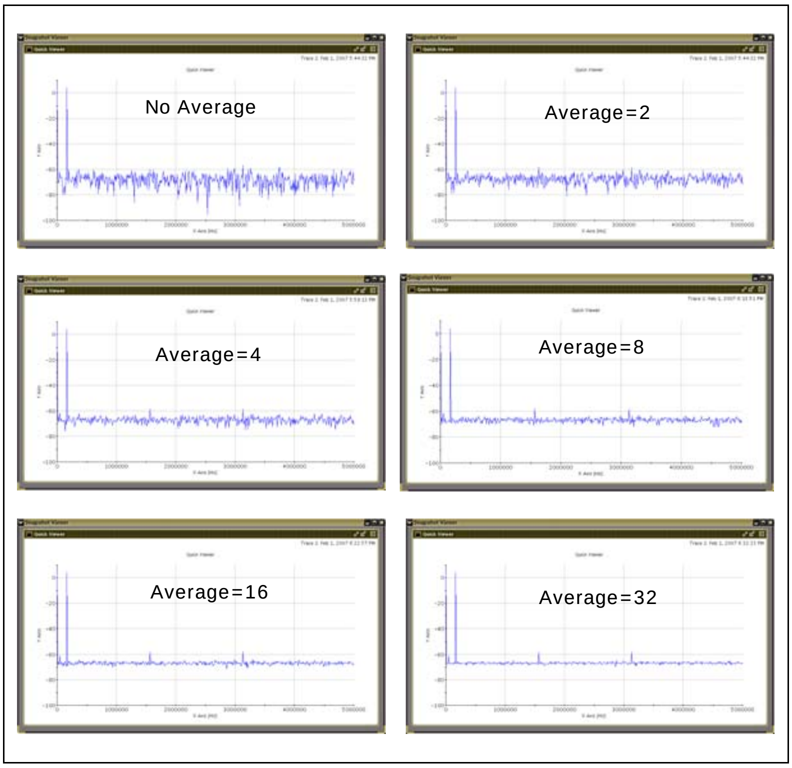

Figure 19: Frequency Domain Averaging

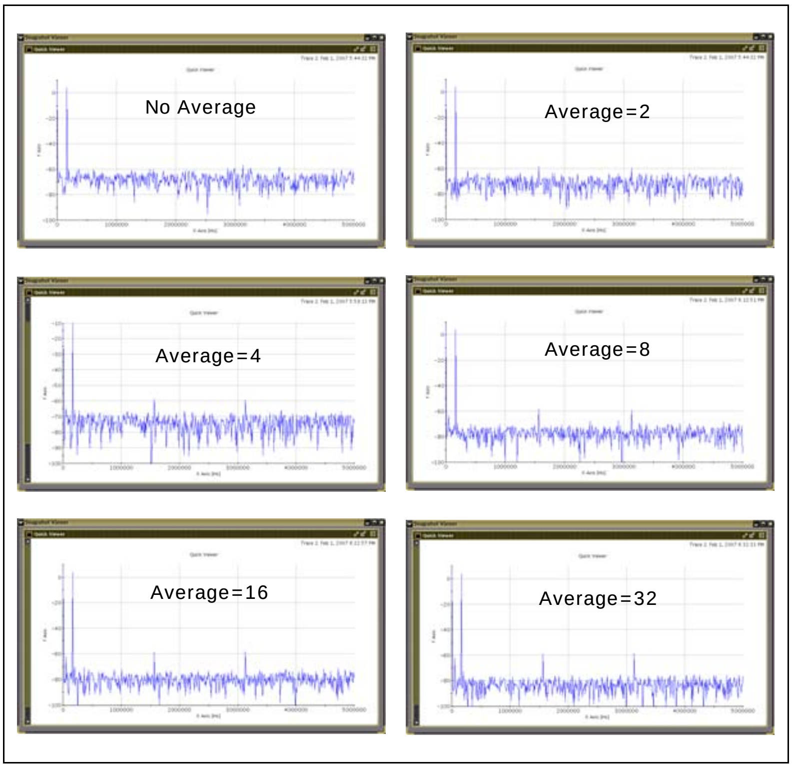

Figure 20: Time Domain Averaging

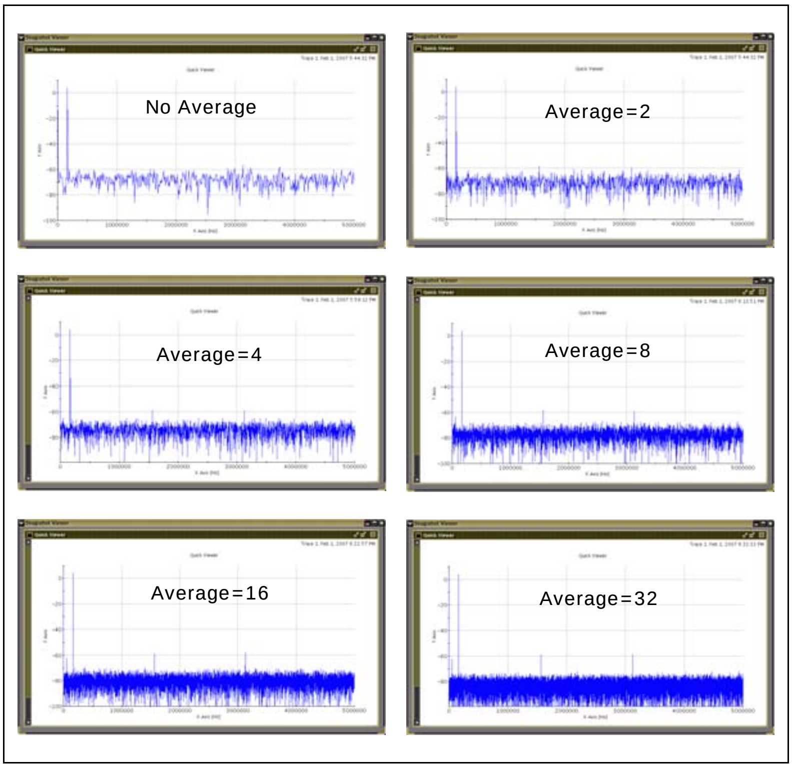

Figure 21: "All Data FFT"

Below table shows the summary of the number of average.

Table 6: Summary of the number of average

Average 2 times					
	Original	Frequency Domain	Time Domain Average	All Data FFT	Unit
NPD	-106.35	-106.43	-109.60	-106.41	dBm/Hz
SNR	43.33	43.42	46.59	43.40	dB
Fundamental	3.98	3.98	3.98	3.98	dBm
Spurious 1	-66.67	-65.57	-68.33	-68.33	dBm
Spurious 2	-60.06	-57.89	-58.19	-58.19	dBm
Spurious 3	-56.76	-58.64	-59.32	-59.32	dBm
Noise Floor p-p	37.39	20.57	29.99	32.17	dBm
Average 4 times					
	Original	Frequency Domain	Time Domain Average	All Data FFT	Unit
NPD:	-106.35	-106.48	-112.22	-106.46	dBm/Hz
SNR	43.33	43.47	49.21	43.45	dB
Fundamental:	3.98	3.98	3.98	3.98	dBm
Spurious	-66.67	-61.70	-64.95	-64.95	dBm
Spurious	-60.06	-58.47	-58.90	-58.90	dBm
Spurious	-56.76	-58.51	-59.02	-59.02	dBm
Noise Floor p-p	37.39	14.10	58.03	59.43	dBm
Average 8 times					
	Original	Frequency Domain	Time Domain Average	All Data FFT	Unit
NPD:	-106.35	-106.48	-114.79	-106.46	dBm/Hz
SNR	43.33	43.47	51.78	43.45	dB
Fundamental:	3.98	3.98	3.98	3.98	dBm
Spurious	-66.67	-61.45	-63.41	-63.41	dBm
Spurious	-60.06	-57.87	-58.36	-58.36	dBm
Spurious	-56.76	-58.41	-59.07	-59.07	dBm
Noise Floor p-p	37.39	9.71	34.49	49.48	dBm
Average 16 times					
	Original	Frequency Domain	Time Domain Average	All Data FFT	Unit
NPD:	-106.35	-106.48	-116.88	-106.46	dBm/Hz
SNR	43.33	43.47	53.87	43.45	dB
Fundamental:	3.98	3.98	3.98	3.98	dBm
Spurious	-66.67	-60.85	-62.43	-62.43	dBm
Spurious	-60.06	-58.03	-58.72	-58.72	dBm
Spurious	-56.76	-57.57	-58.17	-58.17	dBm
Noise Floor p-p	37.39	7.52	37.96	45.86	dBm
Average 32 times					
	Original	Frequency Domain	Time Domain Average	All Data FFT	Unit
NPD:	-106.35	-106.49	-118.82	-106.47	dBm/Hz
SNR	43.33	43.48	55.81	43.46	dB
Fundamental:	3.98	3.98	3.98	3.98	dBm
Spurious	-66.67	-60.96	-62.45	-62.45	dBm
Spurious	-60.06	-58.14	-58.85	-58.85	dBm
Spurious	-56.76	-58.05	-58.68	-58.68	dBm
Noise Floor p-p	37.39	4.99	37.25	54.03	dBm

From these results, it is obvious that only “Frequency Domain Averaging” method can reduce the noise floor peak-to-peak deviation. Below graph shows the relation of the ‘Number of average’ and the ‘Noise floor p-p’ in dB. The Blue line shows the actual simulation results in above table. The linear regression is calculated from these results and the Red line shows the regression result.

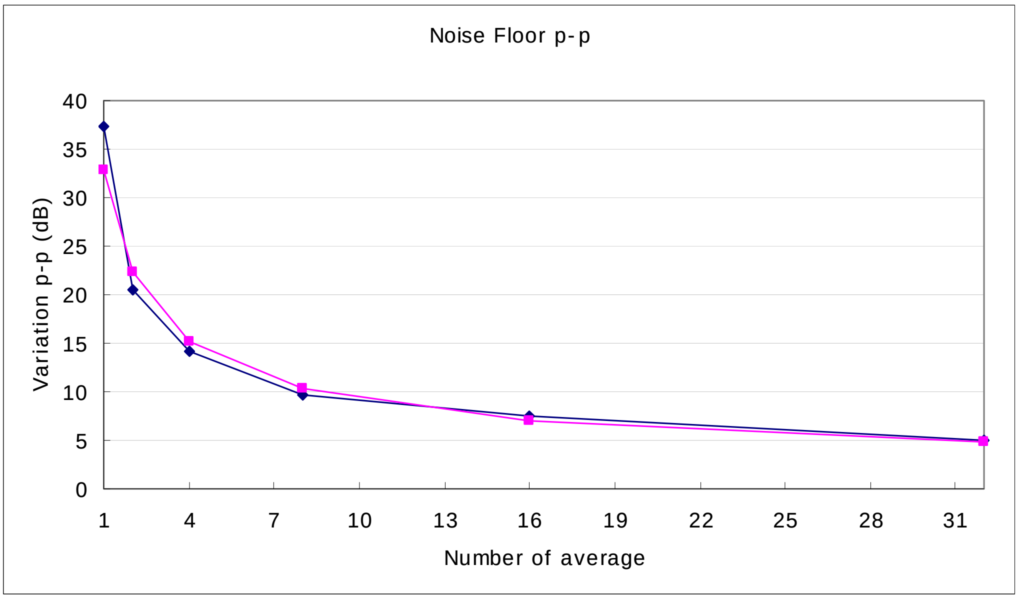

Figure 22: The relation of Deviation p-p in dB and the number of average

The linear regression result using above simulation data is:

$$Deviation(dB) = Deviation\_of\_original \times Number\_of\_average^{-0.554955}$$

From this result and the calculation speed, you can determine the suitable number of average for your test.

Calculation Speed

With my workstation, I measured the calculation speed of each averaging methods.

Table 7: Calculation time

Averaging	32 times	16 times	8 times	4 times	unit
ALL Data FFT	7389	3512	1527	789	usec
Original	141	141	143	140	usec
Frequency domain average	1957	1084	570	321	usec
Time domain average	842	489	300	211	usec

Original has 1024 data points (512 frequency bins)

The measurement time is not included in above data. The measurement time is also increased when the averaging is performed. If the measurement time is critical, it is recommended not to perform the averaging.

Below figure shows the calculation time of each averaging method.

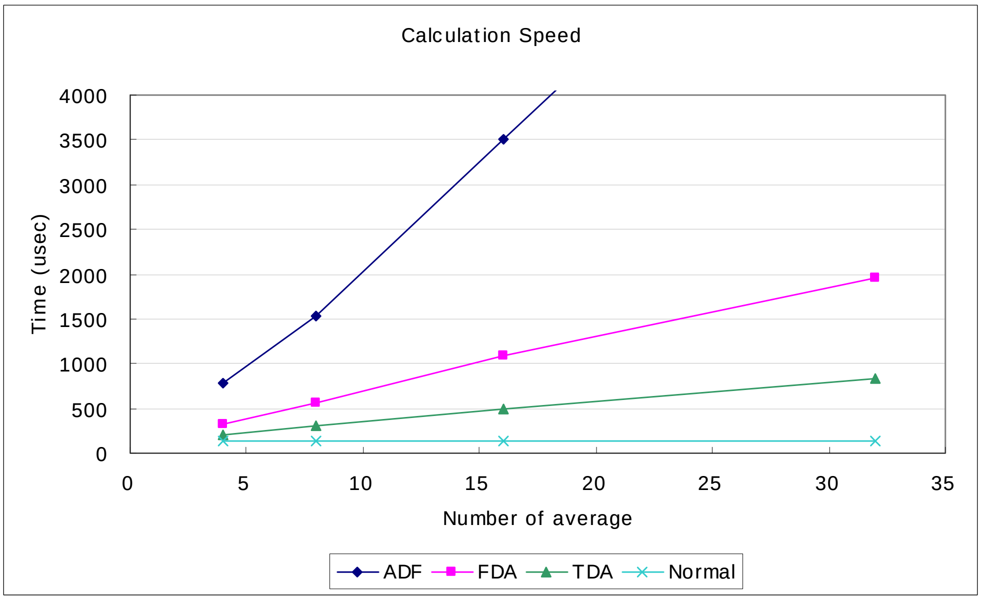

Figure 23: Calculation time of each averaging method

This calculation time data may be different from the actual case because of the workstation performance, workstation configuration and the processes that are running while this calculation is performed.

Below shows the basic workstation data used for this calculation speed measurement.

HP workstation xw4100
Pentium 4 - 3.2GHz
Memory 2GB

Enterprise Linux 2.4.21-40
RedHat Linux 3.2.3-54
SmarTest 5.4.2