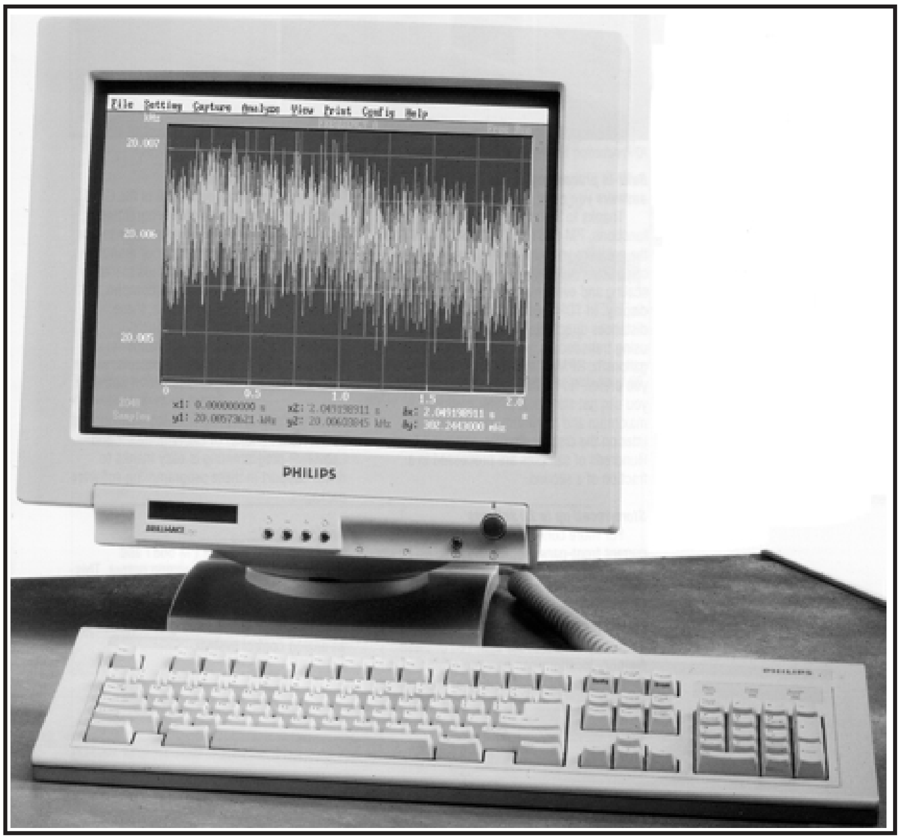
ABC of Modulation Domain Analysis



Background

Why are oscilloscopes such popular instruments? They do not measure voltages very accurately. Even the lowest cost DVM can produce more accurate results when measuring *static* voltages. Yes, that's it: Oscilloscopes let you view *dynamic* voltages, voltages that varies over time. This ability makes oscilloscopes very suited to view and analyze most types of dynamic signals. And the absence of this dynamic signal view, makes voltmeters limited in use, except for verification and calibration of static voltages.

Dynamic signal analysis of amplitude and frequency

Amplitude and frequency contents are the two most important signal properties of any signal. *Oscilloscopes* are used to analyze changes in amplitude but not to analyze changes in frequencies.

The traditional tool for analyzing the frequency contents of a signal is the *Spectrum Analyzer*. This can find static (fixed) frequency components or give a statistical (averaged) picture of dynamic (changing) frequencies.

To view also changing frequencies a third type of tool is needed; the *Modulation Domain Analyzer (MDA)* sometimes also called *Time & Frequency Analyzer*.The three basic signal analysis tools.

To analyze *all* dynamic properties of a signal, three basic tools are needed;

- Oscilloscope
- Spectrum or FFT-analyzer
- Time & Frequency Analyzer

An Oscilloscope lets you view voltage variation over time, a Spectrum or FFT-Analyzer lets you view voltage distribution over frequency and a MDA lets you view frequency variation over time. See fig.1

All three tools are needed for a com-plete signal analysis that covers all three axis; Voltage, Frequency and Time.

TimeView™ - a MDA solution

The Time & Frequency Analyzer TimeView from Pendulum consists of three parts:

- Fast sampling front-end (CNT-81)
- Standard PC with GPIB-interface
- TimeView control and analysis SW

The signal to be characterized is connected to the front-end input (CNT-81 timer/counter/analyzer). All setting controls are made from the PC. Graphs can be printed on the PC-printer and settings and results are stored as ASCII-files, that are easily imported in various programs.

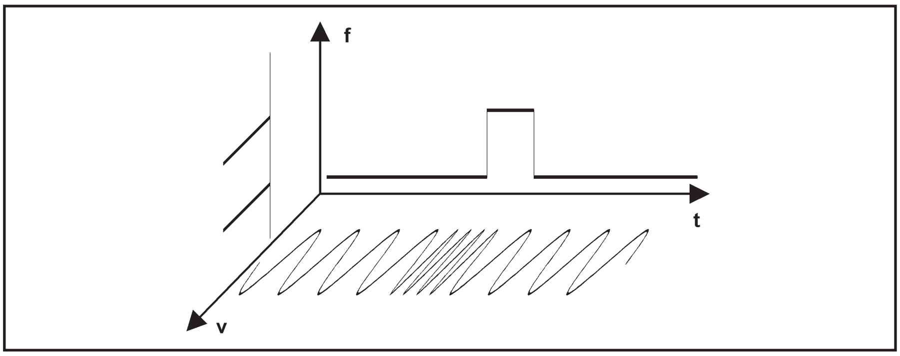


Figure 1 *A sinewave signal with squarewave FM as shown on an oscilloscope (V vs. t), a spectrum analyser (V vs. F) and a modulation domain analyser (f vs. T). All these instruments are needed for a complete view of the signal properties*

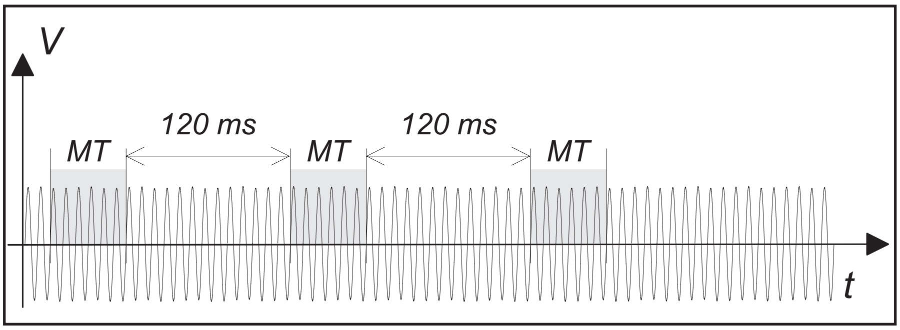


Figure 2 *Free-running frequency samples are taken during a certain measuring time (MT) and with a 120 μs dead-time between measurements. The length of MT is determined by the required resolution of the frequency samples*

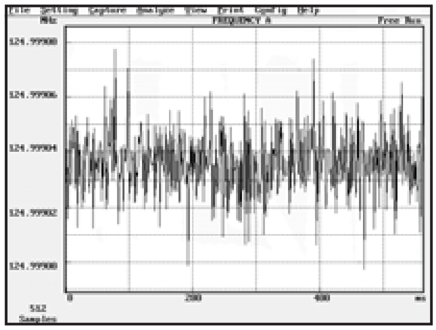


Figure 3 *Free-running frequency samples from a fairly stable signal source, shown in the modulation domain*

Capturing single-shot events (free-run capture)

Single-shot events occur just once. Or the repetition rate may be so low, that you only want to measure on one cycle (e.g. temperature cycling of oscillators). To characterize frequency variations over time, the front-end (CNT-81) makes repeated frequency measurements that are stored in its internal memory.

Sample rate

There is a “dead-time” between measurements of 120 μs. See figure 2. With a *Measuring Time MT*, for each frequency measurement, the sample rate is:

$$Sample\ rate = \frac{1}{(MT + 120\mu s)}$$

An *MT* of 100 μs (TimeView default value) gives each frequency sample a 6-7 digits resolution. TimeView will then take frequency samples, every 220 μs (100+120 μs) or approx 4500 times/s.

Time Stamping

CNT-81 will “time stamp” each measurement in a block., via a separate *time stamp clock* (125 ns resolution). The value of this clock is read and stored together with the start of all subsequent measurements.

When data capture is finished, the values, and the corresponding time stamps, are transferred to the PC as a two-dimensional array and TimeView takes care of the display and analysis. In this way TimeView knows both actual measurement values and the point of time when the measurement was made.

The time stamping feature is especially important for non-continuous signals like when measuring frequency in bursts, or the pulse width of random pulses.

Capturing repetitive events (repetitive sampling)

While free-run capture has a sample rate of well over 8000 values/s, this is not enough in some applications. Consider for example the measurement of output settling time of a VCO or a synthesizer. Here you could expect settling times of a few microseconds. To measure this, you need to improve the time scale, so it corresponds to millions of measurements/s. TimeView does that on periodic repetitive events, in a way called *repetitive sampling*.

With this capture method TimeView measures not once, but several times in subsequent cycles. Each measurement is somewhat delayed in the cycle, with respect to the previous measurement. When enough samples are taken, these are put together to show a picture of the fast frequency transient. See figure 4.

The delay between subsequent measurements can be set down to 100 ns steps. This corresponds to a virtual sampling rate of 10 MSa/s. As with repetitive sampling in a DSO, there must be an external synchronization signal available, or a unique trigger point some-where in the signal.

An example is shown in fig 5 where the frequency response of a VCO is shown. The VCO is controlled via a repetitive pulse with a fast risetime. The input voltage toggles between two levels (high/low voltage) and consequently the output frequency should switch between two frequency values (high/low frequency). The actual frequency response (f vs t) is recorded via TimeView’s repetitive sampling. In the graph, cursor measurements show that the frequency swing is approx 29 MHz (from 433 to 462 MHz) and the “rise time” between cursor positions is 10.7 μs.

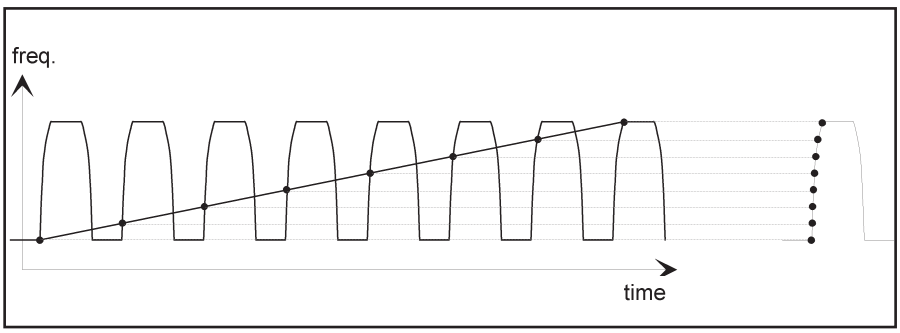


Figure 4 *With repetitive sampling, many measurements are made at fairly long intervals and put together to show a fast frequency transition. Each measurement can be delayed down to 100 ns with respect to the previous measurement.*

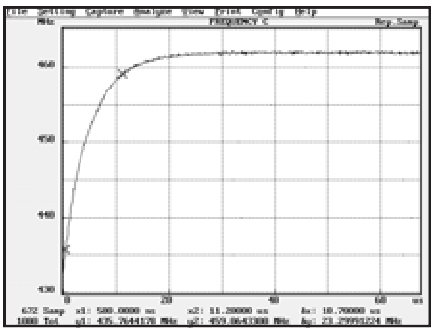


Figure 5 *Repetitive sampling of a fast changing frequency output from a UHF VCO*

Viewing frequencies that varies with time

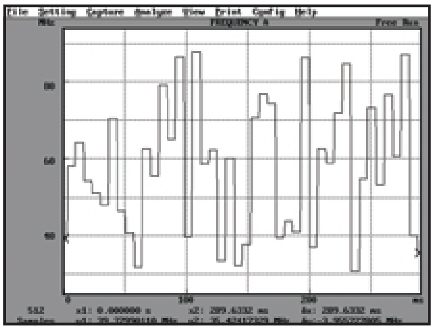


Figure 6 *Frequency hopping is used in e.g. military frequency agile communication*

There exist a great variety of frequency sources. Some are rock stable, and some vary the output frequency a lot. Examples of varying frequencies are found in *frequency hopping communication*. In figure 6, a military frequency agile communication carrier is shown, with pseudorandom frequency changes every 7 ms. The purpose of this rapid change of carrier frequency is of course to prevent “the enemy” from listening.

Another example is the frequency hopping in spread-spectrum communication, found e.g. in noisy industrial environments or in wireless LAN:s. Here the purpose is to supply safer communication, with less interruptions.

Yet other examples of varying frequency are various frequency sweep signals. These are found in LF audio testing of consumer electronics equient as well as in very high frequency radar “chirps”.

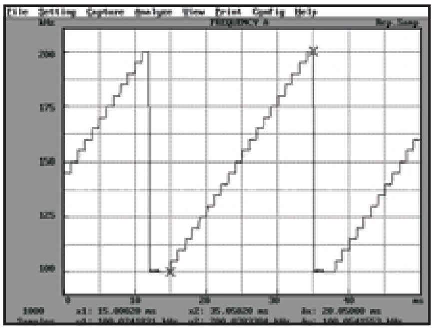


Figure 7 *A frequency sweep made by a digital synthesized generator*

Figure 7 shows an example of a frequency sweep from 100 to 200 kHz, made by a modern function generator. This generator uses digital techniques to synthesize the output frequency. That is why the frequency is changing in 20 discrete steps during the sweep period. An old-fashioned analog sweep generator would give a straight line instead, as shown in figure 8.

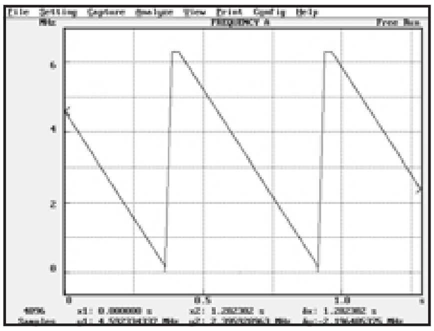


Figure 8 *A frequency sweep made by an analog generator*

Needless to say, the visualization of frequencies that change over time can only be made by a MDA, not an oscilloscope nor a spectrum analyzer.

Measuring jitter and frequency noise

In todays computer and digital telecommunications systems, it is more essential than ever before to keep control over system jitter. But what is *jitter*?

Jitter is the cycle-to-cycle variations of a periodic event; be it period, pulse width or time interval variations. Examples are period variations of a computer clock oscillator or clock-to-data jitter in a communications system. A big amount of jitter can sometimes be detected on analog scopes, seen for example as a fuzzy edge of a pulse. See figure 9.

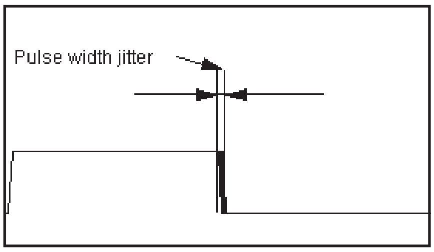


Figure 9 *Jitter can be viewed on an oscilloscope as a “fuzzy edge”, but not quantified*

To *measure* the jitter, you need to do a lot of single pulse width measurements, and statistically process the samples to get the max, min and standard deviation values from the samples. The Δ(max-min) is called the *peak-to-peak jitter*, but normally the most important measure is the *Rms-jitter* (standard deviation).

An oscilloscope can indicate peak-to-peak jitter but not rms-jitter, whereas TimeView can accurately calculate both types of jitter, and also display the distribution of the actual measurements in a distribution histogram. Such a histogram may help to reveal the “nature of jitter”. A random jitter

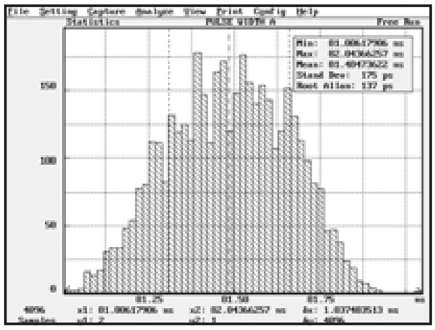


Figure 10 *Random jitter gives a Gaussian distribution*

gives a gaussian distribution, see figure 10.

Jitter caused by a sine modulation, gives a histogram that looks like a bathtub, like in figure 11.

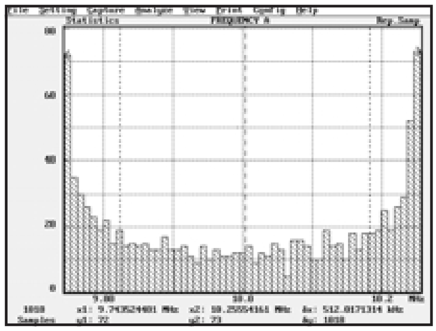


Figure 11 *Jitter caused by sinusodial frequency modulation gives a “bathtub” distribution*

Jitter caused by a square wave modulation on the other hand, gives a histogram with two distinct bars at the maximum respectively the minimum value. See figure 12.

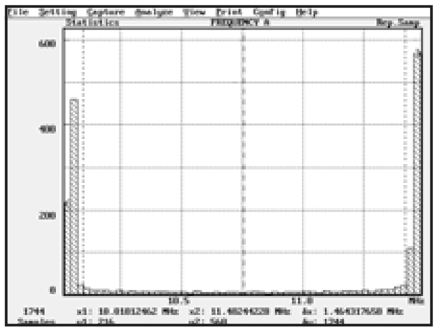


Figure 12 *Jitter caused by a squarewave frequency modulation gives a “2 bar” distribution*

Frequency modulation

A frequency modulated (FM-) signal is difficult to characterize with a normal oscilloscope. The frequency varies and thus the period is changing too. You can not get a stable triggering, and to figure out the nature of the modulating signal is a guesswork.

TimeView can characterize FM easily. Simply because a MDA displays frequency that varies over time, and that is exactly what FM is all about.

A representation of a frequency modulated carrier, in a frequency vs time graph, is shown in figure 13.

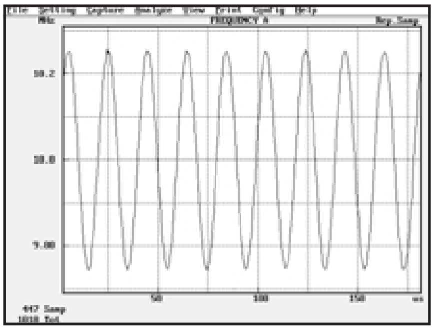


Figure 13 *FM shown in the modulation domain (f vs t). The modulation signal shape is revealed*

From fig. 13, you can quickly conclude that the carrier is approx 10 MHz, having a frequency deviation of approx 2% (0.2 MHz). By looking at the time axis, you see that the modulation is periodic and sinusoidal, having a frequency of approx 50 kHz (20 μs modulation cycle). So, at one quick glance, we have an indication of all three important frequencies in an FM-signal; that is:

- Carrier frequency f
- Frequency deviation f
- Modulation frequency f

FFT-analysis

To analyze the modulation in more detail, you can use the built-in FFT-function. When applied on the signal in fig. 13, that is “frequency vs time”, it will produce the graph showed in fig. 14, that is “frequency vs frequency”.

Just as a “normal” FFT-operation on a voltage vs time graph will show the spectral contents of the original signal, the FFT-graph shows the spectral contents of the frequency vs time graph.

In fig. 14 the found modulation frequencies are shown along the X-axis, just as in a “normal” FFT on voltage vs time. Along the Y-axis we find the carrier and the frequency deviations from the carrier caused by modulation.

In fig.14 you find two cursors, shaped as “X”. The left cursor (x1 in the graph) tells

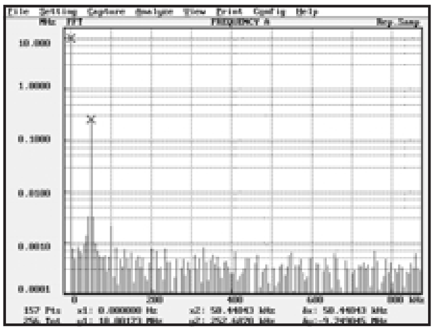


Figure 14 *FFT-processing of the modulation domain graph shows carrier, modulation frequency and deviation*

us that the carrier is 10 MHz. The right cursor (x2) shows that the modulation frequency is 50 kHz causing a deviation of the carrier of 250 kHz.

Distribution histogram

Also the statistical distribution histogram can give valuable information about the modulation, see fig.15. From the shape of the distribution histogram, we can conclude that the modulation is sinusodial (bathtub shape). We can also read the maximum frequency deviations as well as the carrier frequency (mean frequency over N modulation cycles).

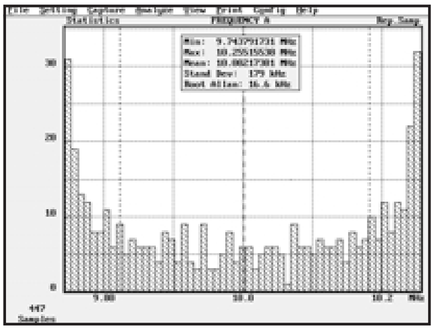


Figure 15 *The distribution histogram shows frequency deviation and indicates sinusodial modulation*

Finding very small unwanted modulation sources

TimeView is an excellent tool for *frequency stability analysis*, and an ideal complement to a spectrum analyzer, whose strength is amplitude stability analysis.

Furthermore TimeView can be used for troubleshooting designs in order to track down causes for noise or interference.

Look at figure 16 that shows the output frequency from a pulse generator, showing a certain amount of jitter.

The jitter seems to be of random nature (distribution histogram in fig.17).

Fig. 18 shows however that there exists a dominant 100 Hz modulation source, i.e. the power supply, that causes FM noise on the output signal.

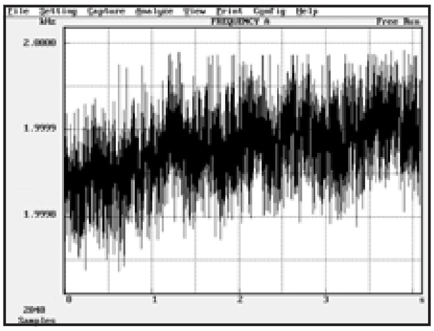


Figure 16 *Frequency vs time from a pulse generator output*

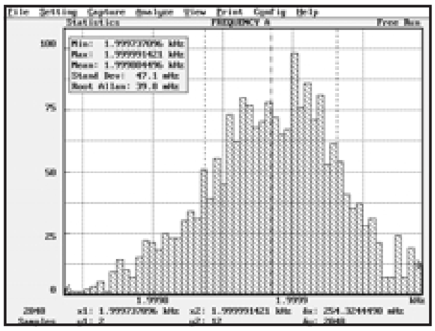


Figure 17 *The histogram of frequency vs time data indicates random noise*

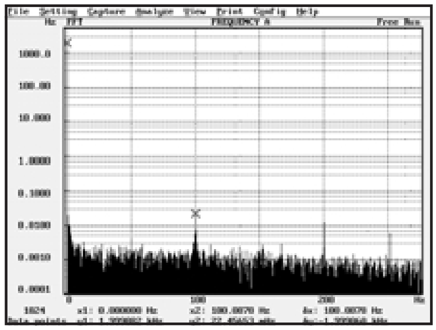


Figure 18 *FFT-processing of the modulation domain graph shows tha the power supply (100 Hz modulation) causes the noise*

Summary

TimeView is a tool designed for:

- Showing dynamic frequency variations (frequencyscope)
- Analyzing noise and jitter
- Analyzing modulation