Some considerations on clock signal quality

J. Serrano, T. Włostowski

July 16, 2012
1 The imperfect clock signal

The goal of a timing system is to provide a common notion of time in a distributed environment. This notion of time is usually the result of counting ticks of a clock signal from an arbitrary instant. The clock signal is ideally of perfect periodicity and stability. Real-world clocks, however, present imperfections [8] in both amplitude and phase as expressed in eq. 1.

\[ a(t) = A (1 + \alpha(t)) \sin(\omega t + \phi(t)) \]  

(1)

In our case, most of these clock signals are output by digital gates with hard amplitude limiters. These square signals do not suffer from amplitude modulation, so we will ignore the \( \alpha(t) \) term from now on. The random variations in the zero-crossing of the pseudo-periodic signals arise from the \( \phi(t) \) term, usually called phase noise. Ignoring amplitude modulation, eq. 1 can be re-written as

\[ a(t) = A \sin \left( \omega \left( t + \frac{\phi(t)}{\omega} \right) \right) \]  

(2)

showing that the \( \frac{\phi(t)}{\omega} \) term, which has dimensions of time, represents the time deviations in zero-crossing between the perfect and the imperfect periodic waveforms. \( \phi(t) \) is a random signal whose rms value is in principle a good indicator of clock quality. Dividing that rms value by \( \omega \) gives the clock jitter. Why is phase noise so important? Because this imperfect clock is typically distributed to many receivers, where local counting is done and the common notion of time is generated. In order to compensate for delays in cables and fibers, a constant correction is applied to the local time base, but this assumes the clock is a perfect copy of itself \( T \) seconds ago, where \( T \) is any multiple of the clock period. If this is not the case, as in all real-life clocks, the delay compensation mechanism does not fully achieve its goal.

2 Phase noise and jitter

Unfortunately, all clocks ultimately diverge in phase and even frequency, in such a way that the rms calculation of jitter gets bigger and bigger as the averaging time grows. In order to tackle this problem, it is useful to work in the frequency domain. The Fourier transform of \( \phi(t) \), noted \( \Phi(f) \) has the same energy as the time-domain signal. This result, expressed mathematically in eq. 3, is known as Parseval’s theorem [5]:

\[ \int_{-\infty}^{+\infty} |\phi(t)|^2 dt = \int_{-\infty}^{+\infty} |\Phi(f)|^2 df \]  

(3)

The units of the left-hand side (LHS) of eq. 3 are \( rad^2 s \). A real-life signal would be bounded in time. If we call \( \phi_T(t) \) a signal which is non-zero only between times \( \frac{-T}{2} \) and \( \frac{T}{2} \), its Fourier transform is:

\[ \Phi_T(f) = \int_{-T/2}^{+T/2} \phi_T(t) e^{-j2\pi ft} dt \]  

(4)
Re-writing eq. 3 with the truncated signal and dividing both sides by $T$ we have:

$$\frac{1}{T} \int_{-T/2}^{+T/2} |\varphi_T(t)|^2 dt = \int_{-\infty}^{+\infty} \frac{\Phi_T(f)|^2}{T} df$$  \hspace{1cm} (5)

Since the LHS of eq. 5 is clearly a measure of the power of the signal, the term $|\Phi_T(f)|^2$ in the RHS can be interpreted as a Power Spectral Density (PSD). In fact, the Wiener-Khintchine theorem \cite{6} tells us that

$$S^{II}_\varphi(f) = \lim_{T \to \infty} \frac{1}{T} |\Phi_T(f)|^2$$  \hspace{1cm} (6)

where $S^{II}_\varphi(f)$ is the two-sided PSD of the random process $\varphi(t)$. Multiplying by two, we get the one-sided PSD which is the most usual measure of oscillator phase noise. It is also customary to average $m$ finite-time measurements to get an approximation of the one-sided PSD:

$$S_\varphi(f) \approx 2 \langle |\Phi_T(f)|^2 \rangle_m$$  \hspace{1cm} (7)

Taking the square root of eq. 5 we would have the phase noise rms value, and dividing the result by the nominal frequency gives the jitter. The problem, as we said, is that increasing the integration limits results in bigger and bigger measured jitter.

In real life, however, an application – as we shall see – is only sensitive to jitter generated between two finite limits in the PSD curve. Figure 1 shows a typical plot of one-sided PSD ($S_\varphi(f)$) of the phase noise for an oscillator. Integration limits are set between $f_L$ and $f_H$. Phase noise below $f_L$ corresponds to variations which are so slow as to be common mode for all timing receivers under all circumstances. For example, if accelerators at CERN change beam every 1.2 seconds, phase noise below say 1 mHz will give an almost constant contribution during the 1.2-second span and therefore will not affect the performance of the timing system. Reasons for establishing an upper limit in integration stem mainly from the inability of some systems to react to such fast variations, i.e. to limitations in bandwidth. These limitations can be in electronics, such as the bandwidth of the input stage of a digital gate, or in electro-mechanical systems such as an RF accelerating cavity. It is important to justify lower and upper integration limits for a given application based on both requirements and an intimate knowledge of the system.

Figure 1 also illustrates different types of noise, which can be identified by the different slopes of their PSDs in a log-log graph \cite{8}. White phase noise dominates the high frequency area and has a flat distribution. Moving towards lower frequencies, we find flicker (pink) phase noise, which is characterized by a PSD scaling as $f^{-1}$. Since frequency is the derivative of phase, white frequency noise – arising from white noise in the frequency-setting elements of an oscillator – features an $f^{-2}$ slope in the phase noise PSD diagram. Higher order $f^{-n}$ terms can also be present. This low-frequency area of the graph will feature quick divergence under integration, and corresponds to the problematic long time-spans mentioned earlier for the time-domain representation.
2.1 Sampled phase noise considerations

In some situations, we actually have samples of phase noise and it then becomes desirable to calculate overall jitter using a computer program. Given a sampled version of $\varphi(t)$, called $\varphi[n]$, and its Discrete Fourier Transform (DFT), $\Phi[k]$, Parseval’s theorem becomes

$$\sum_{n=0}^{N-1} |\varphi[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\Phi[k]|^2$$

(8)

where $N$ is the number of acquired points. Dividing eq. 8 by $N$ and taking the square root will give an estimate of rms phase noise. The $\Phi[k]$ values can be multiplied by the sampling period $T_s$ to generate samples of $\Phi_T(f)$, the Fourier transform of the analog $\varphi(t)$ signal, and from them plot $S_\varphi(f)$ using eq. 7.

2.2 Relationship with oscilloscope jitter measurements

In many situations, the parameter of interest is the time-domain first-difference jitter between a certain rising edge of a clock signal and another rising edge $k$ clock periods later. This is typically measured with an oscilloscope triggered off the first edge of interest, observing on the screen $k$ ticks later in infinite persistence mode. Oscilloscope time measurements, however, are inherently less precise than frequency-domain measurements of phase noise as described in 3. Fortunately, one can take the first-difference jitter problem and cast it into phase noise language so that first-difference jitter information can be extracted from phase noise measurements in the frequency domain [3]. Let us first define a discrete time deviation signal which measures the deviation between the zero-crossing of a real oscillating signal and that of an ideal one. From eq. 2, discretizing, we have

$$x[n] = \frac{\varphi[n]}{\omega}$$

(9)

where $x$ is a time deviation and $\omega$ is the nominal frequency of the oscillating signal (which is also the sampling frequency of this discrete-time system). The oscilloscope measurement is equivalent to applying a filter to the sequence $x[n]$
and measuring the rms value at the output of the filter. This filter implements the difference equation

\[ y[n] = x[n] - x[n-k] \]  

(10)

whose transfer function in the Z domain is:

\[ H_k(z) = 1 - z^{-k} \]  

(11)

The transfer function in the frequency domain can be obtained by evaluating the Z transfer function in the unit circle:

\[ H_k(e^{j\omega_d}) = (1 - \cos \omega_d k) + j \sin \omega_d k \]  

(12)

where \( \omega_d \) is now a digital frequency constrained between \(-\pi\) and \(\pi\). This frequency can be converted to the analogue frequency \(f_a\) domain using

\[ \omega_d = 2 \pi \frac{f_a}{f_s} \]  

(13)

where \(f_s\) is the sampling frequency. The filter response in the \(f_a\) domain will of course be periodic with period \(f_s\) as for any digital filter. One possible method to calculate first difference jitter from phase noise PSD data is to filter that data with \(|H_k(e^{j\omega_d})|^2\) and then calculate the rms value of the output using the resulting scaled PSD. One remarkable feature of the first difference filter is that it will attenuate phase noise at frequencies close to its zeroes, and in particular very low frequency phase noise will not show up in the result. This is a confirmation of the fact that in many applications, this type of noise is irrelevant because it shows up as common mode in situations where we only care about differences.

3 Measuring phase noise

For low-noise conditions, we can re-write eq. 1 as

\[ a(t) = A (1 + \alpha(t)) \sin \omega t + A \varphi(t) \cos \omega t \]  

(14)

where we have dropped second-order terms. The amplitude and phase noise contributions are now modulating carriers in quadrature, so it is simple to design a device that will discriminate between them for measurement purposes. Figure 2 depicts the internal structure of such a device.

The role of the loose quadrature phase-locked loop (PLL) is to track the incoming signal with very low bandwidth, filtering all the high-frequency phase
Figure 3: Block diagram of a phase-locked loop.

noise we are interested in studying. More on this filtering action will be explained in section 4. Phase noise of lower frequency than the bandwidth of this loose PLL cannot be measured with this device\(^1\). In addition, this PLL has to be set up to generate a signal in quadrature with \(a(t)\) i.e. \(\cos \omega t\) and not \(\sin \omega t\) so that at the output of the mixer and low-pass filter we are left with a baseband signal proportional to \(\varphi(t)\). Digitization of this signal followed by Fourier analysis results in a plot of phase noise PSD vs. frequency.

For cases where very low-noise sources have to be characterized, the noise of the loose PLL can represent a significant portion of the overall measured noise, therefore affecting the precision of the measurement. Some commercial devices feed the signal to two parallel PLL + mixer + low-pass + ADC branches and perform a cross-correlation of measurements. Noise which is uncorrelated between the two branches is attributed to the PLL and other imperfections and discarded before displaying the PSD.

One last important item to bear in mind when measuring phase noise with commercial instruments is that measurements often get reported as \(L(f)\) which is defined as \(\frac{S_\varphi(f)}{2}\). This is recommended in IEEE standard 1139.

4 Phase-locked loops

Phase-locked loops [2] are an invaluable tool in cleaning up the jitter of clocks, among many other possible applications. Figure 3 depicts their internal structure.

The phase detector (PD) block generates an output voltage \(v_d\) proportional to the phase difference between the input and output of the PLL. In Laplace space, its output is therefore

\[
V_d(s) = K_d (\varphi_i(s) - \varphi_o(s)) \tag{15}
\]

The next block after the phase detector is the loop filter, which outputs the

\[
\frac{d\varphi_{\text{VCO}}}{dt} = K_{\text{VCO}} \cdot v_c
\]
control signal for the Voltage-Controlled Oscillator (VCO):

\[ V_c(s) = F(s) \cdot V_d(s) \] (16)

The VCO outputs a signal with a frequency proportional to its input voltage. Since frequency is the derivative of phase, this means that the phase of the signal at the output of the VCO is proportional to the integral of the VCO control voltage:

\[ \Phi_{VCO}(s) = \frac{K_{VCO} \cdot V_c(s)}{s} \] (17)

Since there are no perfect VCOs, we have included a VCO noise source in the diagram, contributing phase \( \varphi_n \). Calculating the output phase \( \varphi_o \) from the two sources in the diagram (reference input phase \( \varphi_i \) and VCO phase noise \( \varphi_n \)) again in Laplace space gives

\[ \Phi_o(s) = H(s) \cdot \Phi_i(s) + E(s) \cdot \Phi_n(s) \] (18)

where \( H(s) \) is called the system transfer function, defined as

\[ H(s) = \frac{K_{VCO} K_d F(s)}{s + K_{VCO} K_d F(s)} \] (19)

and \( E(s) \) is the so called error transfer function, defined as

\[ E(s) = 1 - H(s) = \frac{s}{s + K_{VCO} K_d F(s)} \] (20)

In typical clock-cleaning applications, \( H(s) \) is a low-pass filter, while \( E(s) \) is high-pass. Cut-off frequencies are dictated by PLL parameters, and most importantly the loop filter \( F(s) \). The PSD of the phase noise of \( \varphi_i \) will be filtered by \( |H(s)|^2 \) while the phase noise PSD of the VCO will be filtered by \( |E(s)|^2 \). This means that the low frequency noise in the PSD of \( \varphi_o \) will come from the reference \( \varphi_i \) and the high-frequency noise will come from \( \varphi_n \). The transition from one noise source to the other will be at a frequency determined by the loop parameters. After careful study of the PSDs of \( \varphi_i \) and \( \varphi_n \) it is the task of the designer to choose a cut-off frequency that will minimize overall area under the \( \varphi_o \) PSD curve, and consequently time-domain jitter. In typical systems – like the transmission of a very stable clock over a channel which adds high-frequency noise – the VCO is worse than the reference at low frequencies and better at high frequencies. The point in frequency where the two PSD plots (reference and VCO) cross is in that case an optimum setting for PLL bandwidth, as shown in figure 4.

In figure 3 the phase detector is shown as perfect, with no noise added to it as for the VCO. In practice, phase detector noise is also a concern, but mathematically it is equivalent to reference noise, so that the above formalism can be applied, replacing reference noise by reference + phase detector noise.

**Add comments about correlated and non-correlated low-frequency noise**

One important aspect to bear in mind for applications in which many PLLs are cascaded is that of peaking in the \( H(s) \) Bode plot. Peaking is a rise of the magnitude of \( H(j\omega) \) before it starts its descent at the cut-off frequency (see figure 5. Peaking is typical of under-damped systems [1]. If many PLLs with
Figure 4: Optimal choice of PLL bandwidth for jitter-cleaning applications.

Figure 5: Illustration of transfer function peaking in a Bode plot.
the same (or similar) $H(s)$ transfer function are cascaded, the peaking can be exacerbated and affect phase and gain margin, ultimately resulting in potential instabilities of the overall system. For this reason, in cascaded PLL applications, it is better to adjust the PLL parameters so that the resulting PLL is slightly over-damped.
References


