

Accelerator Timing Systems Overview

J. Serrano, P. Alvarez, M. Lipinski, T. Włostowski (CERN, Geneva)

Abstract

Timing systems are crucial ingredients for the successful operation of any particle accelerator complex. They are used not only to synchronize different processes but also to time-stamp and ensure overall coherency of acquired data. We describe fundamental time and frequency figures of merit and methods to measure them, and continue with a description of current synchronization solutions for different applications, precisions and geographical coverage, and some examples. Finally, we describe new trends in timing technology and applications.

INTRODUCTION

Control and data acquisition systems for particle accelerators often need to ensure time-coherent behavior in a distributed environment. This is usually achieved by a dedicated timing network whose purpose is to distribute a common notion of time from one master to many receiving nodes. Time distributed in this way can be universal – such as Coordinated Universal Time (UTC) – or locally generated in the facility. The former is more convenient for cases where data tagged using many time sources must be correlated, and the latter is typical of cases where a beam-synchronous clock signal is used for time-keeping. Once a common sense of time is present in all nodes, time-based control and data acquisition strategies can be used [2], instead of or in addition to event-based strategies. To illustrate the difference between these two paradigms, let's look at an example. Imagine somebody wants many kicker magnets to fire all at midday exactly:

- In an *event-based* system the timing master would wait for UTC=12:00 and send a message using the timing network, to which the timing receivers concerned would react by producing pulses driving the kickers.
- In a *time-based* system, the timing master would broadcast a message instructing the receivers to output a pulse at midday. This message would be sent some time before the deadline of 12:00 UTC, and the receivers would use their internal notion of time – derived from the same timing network using techniques we will describe shortly – to compare it with the requested time and generate the output pulse when there is a match.

Real-life accelerator timing systems often combine the two approaches. Event-type behavior is very popular for driving actuators, especially when the master needs to react to

external conditions with low latency. On the other hand, having a solid timebase everywhere is very practical for time-tagging acquired data. Also, for actuators, it should be noted that a time-based system in which the deadlines are very close to the emitting time is in effect an event-based system.

TIMING CONCEPTS

The basic idea of an event system is very simple, and its implementation – i.e. deciding which events to be broadcast when – is very accelerator-specific. The rest of this paper will look instead at how to distribute a common notion of time from a central location to many distant receivers. This notion of time is embodied by a clock contained in each one of the receivers. A clock can be simplistically thought of as the combination of a clock signal (a repetitive square wave of a given frequency) and a counter counting ticks of that clock signal from some arbitrary instant. If we ensure that two nodes hold the same count at any given moment, they are said to be synchronized. As we will see in the following sections, there are techniques to evaluate the delay δ of transmission between a master and a slave node. If we can send a clock signal from master to slave and phase-shift this signal by δ seconds in the receiver we should in principle obtain a copy of the master clock signal in the slave. But this assumes that the clock signal is perfectly periodic, which is never the case. The following section deals with departures from this ideal case and ways to partially mitigate them.

The imperfect clock signal

We will take the clock signal to be a sine wave without any loss in generality, since a square wave is nothing else than a sum of sine waves, even a finite one because of bandwidth limitations. Real-world clock signals present imperfections [9] in both amplitude and phase as expressed in eq. 1.

$$a(t) = A(1 + \alpha(t)) \sin(\omega t + \varphi(t)) \quad (1)$$

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 $\varphi(t)$ term, usually called phase noise. Ignoring amplitude

modulation, eq. 1 can be re-written as

$$a(t) = A \sin \left(\omega \left(t + \frac{\varphi(t)}{\omega} \right) \right) \quad (2)$$

showing that the $\frac{\varphi(t)}{\omega}$ term, which has dimensions of time, represents the time deviations in zero-crossing between the perfect and the imperfect periodic waveforms. $\varphi(t)$ is a random signal whose rms value is in principle a good indicator of clock quality. Dividing that rms value by ω gives the clock jitter.

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 $\varphi(t)$, 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 [7]:

$$\int_{-\infty}^{+\infty} |\varphi(t)|^2 dt = \int_{-\infty}^{+\infty} |\Phi(f)|^2 df \quad (3)$$

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

$$\Phi_T(f) = \int_{-T/2}^{+T/2} \varphi_T(t) e^{-j2\pi ft} dt \quad (4)$$

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

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

Since the LHS of eq. 5 is clearly a measure of the power of the signal, the term $\frac{|\Phi_T(f)|^2}{T}$ in the RHS can be interpreted as a Power Spectral Density (PSD). In fact, the Wiener-Khintchine theorem [8] tells us that

$$S_{\varphi}^{II}(f) = \lim_{T \rightarrow \infty} \frac{1}{T} |\Phi_T(f)|^2 \quad (6)$$

where $S_{\varphi}^{II}(f)$ 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:

$$S_{\varphi}(f) \approx \frac{2}{T} \left\langle |\Phi_T(f)|^2 \right\rangle_m \quad (7)$$

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

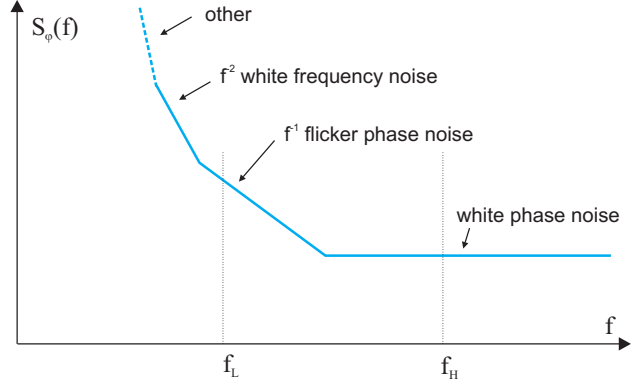


Figure 1: One-sided PSD of phase noise for a typical oscillator.

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 the 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.

Phase noise and jitter can be measured with dedicated instruments or with an oscilloscope in infinite persistence mode. It should be borne in mind that the definition of phase noise involves comparing a noisy waveform with an ideal non-existent sine wave, while typical setups using an oscilloscope measure one edge of the noisy clock signal with respect to another edge happening later [4]. Sometimes the clock signal recovered in a timing receiver is used to clock an Analog to Digital Converter (ADC) [1]. Figure 2 shows the conversion happening in that case between phase noise and signal amplitude noise, resulting in a decrease of the ADC's Effective Number Of Bits (ENOB).

Phase-locked loops

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

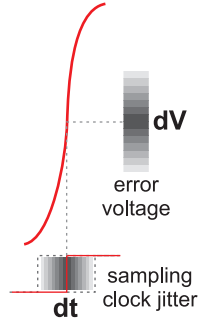


Figure 2: Conversion between clock phase noise and signal amplitude noise in an ADC.

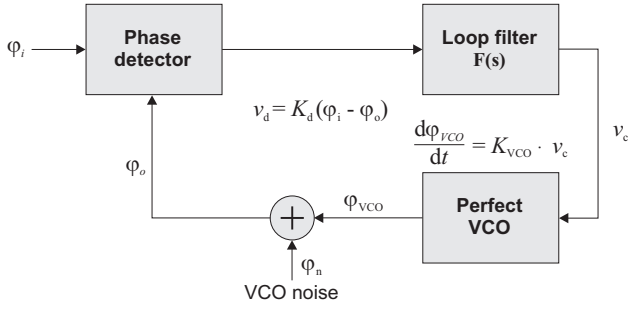


Figure 3: Block diagram of a phase-locked loop.

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

$$V_d(s) = K_d(\Phi_i(s) - \Phi_o(s)) \quad (8)$$

The next block after the phase detector is the loop filter, which outputs the control signal for the Voltage-Controlled Oscillator (VCO):

$$V_c(s) = F(s) \cdot V_d(s) \quad (9)$$

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:

$$\Phi_{VCO}(s) = \frac{K_{VCO} \cdot V_c(s)}{s} \quad (10)$$

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

$$\Phi_o(s) = H(s) \cdot \Phi_i(s) + E(s) \cdot \Phi_n(s) \quad (11)$$

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

$$H(s) = \frac{K_{VCO}K_dF(s)}{s + K_{VCO}K_dF(s)} \quad (12)$$

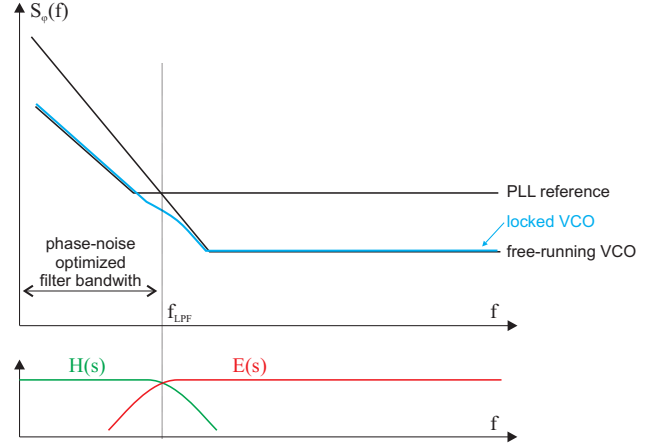


Figure 4: Optimal choice of PLL bandwidth for jitter-cleaning applications.

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

$$E(s) = 1 - H(s) = \frac{s}{s + K_{VCO}K_dF(s)} \quad (13)$$

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 φ_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 φ_o will come from the reference φ_i and the high-frequency noise will come from φ_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 φ_i and φ_n it is the task of the designer to choose a cut-off frequency that will minimize overall area under the φ_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.

TIMING TECHNOLOGIES

In this section we present currently available technologies that allow users to achieve synchronization at different levels of accuracy and precision. By accuracy we mean the degree to which the time base in a receiver matches that of the master on average. Precision refers to the amount of jitter in the receiver's clock signal, i.e. the fluctuations around the average deviation indicated by the accuracy. Very often in accelerator environments a constant time deviation is inconsequential as long as the constant is known and compensated for either in hardware or software (by e.g. correcting time tags). This compensation of fixed delays can

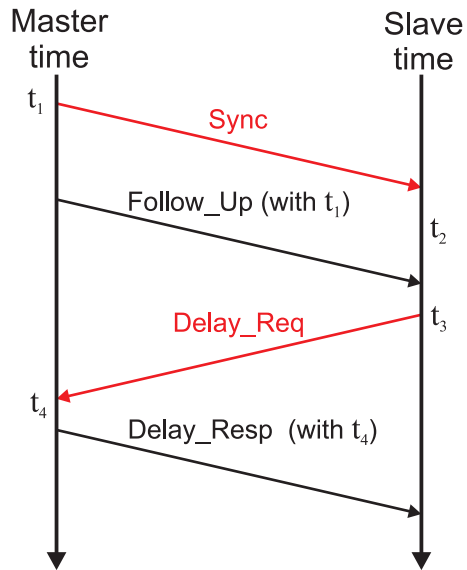


Figure 5: Transmission delay estimation using a two-way scheme.

take the form of tweaking programmable delay generators in the receiver (to produce delayed pulses driving a piece of equipment) and observing the beam until a satisfactory situation is achieved. In other cases, a precise compensation of fixed delays, along with tight jitter control, are needed in the timing network itself. If the master and slave nodes have the capability of time-tagging messages as they emit them and receive them, a scheme such as the one depicted in figure 5 can be used to determine transmission delay.

Knowing time tags t_1 , t_2 , t_3 and t_4 , the slave node can evaluate the transmission delay δ (see eq. 14) and apply a shift to the broadcast time to compensate for that delay and align with the master.

$$\delta = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \quad (14)$$

This scheme assumes that the link is perfectly symmetric. The extent to which that assumption is true, along with the precision of the time tags, results in different precisions for different implementations.

Millisecond timing

The Network Time Protocol (NTP) is typically used to keep UTC time in general-purpose computers. NTP uses messages over the Internet to synchronize NTP clients (timing slaves) to one or more NTP servers (timing masters). Precision is affected by the lack of symmetry arising from two main factors:

- The time-tagging actually happens in software, through a call like `gettimeofday()` in user space, subject to the variability in scheduling latency of typical operating systems.

- The routing of packets through the network can be completely different for packets going from master to slave and vice versa. Also, traversal of routers and switches exhibits non-deterministic latencies.

Due to these limitations, NTP software must use a very powerful statistics artillery to average many measurements. Typical synchronization precisions range from under 1 ms in a well-controlled Local Area Network to around 10 ms through the Internet. These figures are satisfactory for the needs of e.g. workstations in typical control rooms of accelerator centers.

Microsecond timing

The Precision Time Protocol (PTP, IEEE1588) uses the same ideas as NTP but improves on both of its shortcomings:

- Packets can be time-stamped in hardware, using dedicated PTP network cards which sniff the packets as they are emitted or received and freeze a counter to generate a precise time tag at a well-specified moment within the packet.
- Although PTP can work with standard switches, special PTP switches have been developed which do not introduce any meaningful loss in precision due to their variable transmission latencies.

Through the use of these techniques, PTP can give sub-microsecond precisions in a well-controlled environment. Performance is limited by the fact that typical implementations of the clock in the slave are based on a free-running local oscillator whose frequency offset with respect to that of the master has to be compensated for continuously. Even a fixed frequency offset results in a linearly increasing phase difference, so this phase drift has to be compensated by frequent exchanges of time-tagged packets over the network. However, for applications which do not need better than microsecond precision, PTP is perfectly adequate.

Nanosecond and picosecond timing

To palliate the problem of free-running oscillators in the slave, some timing networks recover the clock signal in the slave from the data stream generated by the master. In this scheme, the master uses the clock signal to be distributed as encoding clock signal for the data stream. This clock signal is then used for local counting and time-stamping in the receiver. With the advent of multi-Gb/s data links, the clock signal recovered by the physical layer components must be of very high quality, otherwise the link would not function properly. Using that recovered clock signal ensures that there is no frequency offset with respect to the master, and very infrequent PTP-type exchanges of packets are enough to account for the changes of transmission delay, typically varying only due to thermal – i.e. slow – effects. Two examples of such networks are the beam-synchronous

MRF [6] system used in many light sources and the UTC-synchronous White Rabbit [11] network currently being designed by CERN, GSI and others. Precisions of under 10 ps are relatively easy to achieve using this method (and the jitter minimization techniques presented in the PLL section), and accuracies of around 1 ns are realistic, the limiting factor being hard-to-determine non-symmetric delays in the nodes and the transmission medium.

Another important timing application in the nanosecond realm is the time transfer between laboratories for neutrino oscillation experiments [10]. In these experiments it is important to be able to discriminate neutrinos coming from the emitting lab from those coming from the Sun and other sources. A precision of 1 microsecond is typically enough, but nanosecond precisions open the way to interesting neutrino time-of-flight measurements. Time transfer systems for these experiments typically use the same techniques as national metrology labs use for the manufacturing of UTC time itself. A local atomic clock time base is continuously compared with time received through a GPS receiver. GPS time is noisy in the short term due to perturbations in the atmosphere and other noise sources, but averaged over 24 hours using e.g. a local Cesium clock, it achieves very good accuracy. This averaging can consist e.g. in fitting a straight line with a given slope through the constellation of time-of-day points taken from the Cesium every GPS second. The slope accounts for the frequency offset between the Cesium and the GPS system. The smoothed GPS time base is then considered common for both labs and used to transfer time tags between the two Cesium clocks. Distribution of the Cesium time base inside the labs can make use of the timing networks described above.

Femtosecond timing

Free Electron Laser (FEL) installations typically require the generation and distribution of a precise microwave signal to many destinations with a precision in the tens of femtoseconds. This precision requires a leap from the electronics to the optics realm. Current implementations can be roughly divided into two families: pulsed [5] and Continuous Wave (CW) [12] systems. Both of these systems transmit only a phase-compensated clock signal. They are not data links.

In the case of the pulsed system, the source is typically a mode-locked laser tightly synchronized with the reference microwave oscillator. A partially reflecting mirror is used to bounce back part of the light at the receiving end. Then an optical cross-correlator is used as the two-way delay detection element, and the delay thus measured is used to act on a fiber stretcher in order to keep transmission delay constant. CW systems normally modulate a CW laser with the microwave signal and measure optical phase delay through the fiber using a heterodyne interferometer between the optical signal at the emitter and that bounced back from the receiver, using a partially reflecting mirror as well. While pulsed systems stabilize group delay directly (thanks to the

optical cross-correlator), CW systems measure phase delay and rely on a model to stabilize group delay. Both types of systems have achieved performances well below 50 fs of jitter for fiber lengths of several hundred meters.

CONCLUSIONS

We have explored timing figures of merit and some of the technologies available to fulfill varying requirements in accuracy and precision. In choosing a timing technology, the user should start with an accurate assessment of the needs. This includes, among other things, not only jitter but also the frequency range of interest in the phase noise PSD plot. Additional considerations to bear in mind include whether the system should be UTC or beam-synchronous and the possible need for real-time transmission delay compensation. After the needs are clearly specified in terms of objective and measurable figures of merit, the user can choose an appropriate technology to fulfill them. In this paper we have presented solutions going from software-based millisecond-range synchronization to optical femtosecond-range systems. Each one of them has uses in the field of control and data acquisition systems for particle accelerators.

REFERENCES

- [1] B. Brannon. *Sampled Systems and the Effects of Clock Phase Noise and Jitter*, Analog Devices AN-756, 2004.
- [2] J.C. Eidson. *Measurement, Control and Communication Using IEEE 1588*, Springer, 2006.
- [3] F.M. Gardner. *Phaselock Techniques*, Wiley, 3rd edition, 2005.
- [4] D.A. Howe and T.N. Tasset. *Clock Jitter Estimation based on PM Noise Measurements*, Proceedings of the 2003 IEEE International Frequency Control Symposium and PDA Exhibition.
- [5] J. Kim et al. *Long-term femtosecond timing link stabilization using a single-crystal balanced cross correlator*, Optics Letters, Vol. 32, No. 9, May 1, 2007.
- [6] Micro-Research Finland Oy, <http://www.mrf.fi/>
- [7] A.V. Oppenheim, A.S. Willsky and S.H. Nawab. *Signals and Systems*, Prentice Hall, 2nd edition, 1996.
- [8] A. Papoulis and S.U. Pillai *Probability, Random Variables and Stochastic Processes*, McGraw Hill, 4th edition, 2002.
- [9] E. Rubiola. *Phase Noise and Frequency Stability in Oscillators*, Cambridge University Press, 2009.
- [10] J. Serrano, P. Alvarez, J. Lewis, D. Autiero. *Inter-Laboratory Synchronization for the CNGS Project*, EPAC 2006, THPCH125.
- [11] White Rabbit Project, <http://www.ohwr.org/projects/white-rabbit>
- [12] R. Wilcox et al. *Stable transmission of radio frequency signals on fiber links using interferometric delay sensing*, Optics Letters, Vol. 34, No. 20, October 5, 2009.