N87-28794

A Method for Using a Time Interval Counter to Measure Frequency Stability

C. A. Greenhall
Communications Systems Research Section

This article shows how a commercial time interval counter can be used to measure the relative stability of two signals that are offset in frequency and mixed down to a beat note of about 1 Hz. To avoid the dead-time problem, the counter is set up to read the time interval between each beat note upcrossing and the next pulse of a 10 Hz reference pulse train. The actual upcrossing times are recovered by a simple algorithm whose outputs can be used for computing residuals and Allan variance. A noise-floor test yielded a $\Delta f/f$ Allan deviation of $1.3 \times 10^{-9}/\tau$ relative to the beat frequency.

I. Beat-Frequency Method

In the beat-frequency or single-mixer method of frequency stability measurement, the two sources to be compared must have a small frequency offset, typically in the neighborhood of 1 Hz. The two sources at frequency f_0 are mixed down to a sinusoidal beat note at the offset frequency f_b. This sine wave is passed through a zero-crossing detector, which produces a square wave at the same frequency. The relative time deviation or fractional frequency deviation of the two sources is equal to f_b/f_0 times that of the square wave or, more precisely, its stream of upcrossings, which are spaced approximately one second apart [1]. The improvement to be discussed below deals only with the measurement of these upcrossing times; the analog front end of the system remains the same.

II. Current Measurement System

In the current system used at the JPL Interim Frequency Standards Test Facility (IFSTF), the square wave beat note goes to a custom-built digital module, the Stevens-Sydnor Machine, which latches the readings of a free-running 1 MHz 30-bit counter at the upcrossings and writes them to a 7-track tape, which is processed off-line by a Univac 1100 computer at the Information Processing Facility (Fig. 1). Actually, each Stevens-Sydnor Machine can handle three independent input channels and multiplex them onto the tape. Of course, the Univac software has to correct for the counter rollover, which happens about every 17.9 minutes. In this way, the upcrossing times are captured with a resolution of 1 μs. Otoshi and Franco [2] have been using a system similar to the IFSTF's, but with a 10 MHz counter, to measure Deep Space Station stability.

Although the IFSTF system has given good service, it has several disadvantages. The counter resolution is too coarse for certain applications whose f_b/f_0 ratio is not small enough to put the quantization noise floor below the frequency instabilities to be measured. The 7-track incremental tape drives are obsolete, expensive to rent, and unreliable. (The author had to write a special software routine to recover from unreadable tape blocks.) The Stevens-Sydnor Machines have a problem with input crosstalk; consequently, only one channel at a time can be used. The cumbersome off-line tape processing and graph plotting causes a turnaround time of several hours between running a test and seeing graphical results.

III. An Improved System—The Picket Fence

Because the frequency stability test operation is soon to be moved to the new Frequency Standards Laboratory, this is a good time to find a better way to carry out the tests and get timely results. High-quality commercial equipment for the job is already at hand; namely, a Hewlett Packard 1000 computer and several HP 5334A universal counters, which can measure time intervals with a 1 ns resolution and interface with the computer on an IEEE-488 bus. The problem has been that these are interval counters with dead time between measurements; unaided, such a counter can measure at most every other period of the stream of beat-note upcrossings. This dead-time limitation has now been overcome by the introduction of one more element, a 10 PPS (pulses per second) reference signal, called the Picket Fence, provided by a frequency standard and a divider.

A. Test Setup and Procedure

The new setup is shown in Fig. 2. The beat note square wave goes into Input A of the counter, the picket fence into Input B. The same frequency standard should drive both the counter and the divider to keep the picket fence coherent with the counter. In a multichannel system, each beat note has its own counter and there is only one picket fence signal going into all the B inputs.

To carry out a test, the Period A function of the counter is used to make a preliminary measurement of the period p of the square wave. It is permissible to use an increased gate time or the 100-reading average function when doing this measurement. Having measured the nominal period, the counter is switched to its Time Interval A to B function, and records all subsequent readings. Each reading is the time interval between an upcrossing and the next picket fence pulse. As long as the periods are not too short, the counter has time to reset itself between readings, and hence no upcrossing is missed. From these raw data, the actual upcrossing times are recovered in software by an algorithm discussed below. Figure 3 shows the time evolution of the measurement process.

B. Data Processing

Let d be the picket fence period (100 ms), p the initial period measurement, and $\nu_0, \nu_1, \nu_2, \ldots$ the sequence of time interval data. Let $t_0, t_1, t_2, \ldots$ be the actual upcrossing times relative to some time origin, perhaps one of the picket fence pulses (Fig. 3). Each t_n differs from the corresponding $-\nu_n$ by an unknown integer multiple of d, and we would like to resolve the ambiguities.

Let Δ denote the backward difference operator, e.g., $\Delta t_n = t_n - t_{n-1}$. The following assumptions are made about the beat note:

(1) The first period Δt_1 differs from p by less than $d/2$. Any two successive periods Δt_{n-1}, Δt_n differ by less than $d/2$. This guarantees that the 100 ms ambiguities can be uniquely resolved.

(2) Each period Δt_n is greater than $d + g$, where g is the maximum dead time of the counter. This just guarantees that no upcrossing is missed.

Since the t_n increase quickly and may contain important information in their least significant bits, they are awkward to compute, store, and use. Accordingly, the algorithm actually computes the sequence of time residuals defined by

$$x_n = t_n - t_0 - np, \qquad n = 0, 1, 2, \ldots \tag{1}$$

Figure 4 shows the relationship between the t_n and the x_n. The core of the algorithm is the signed residue function Smod(x, m), which is defined to be x minus the closest integer multiple of m to x. For example,

$$\text{Smod}(3, 5) = \text{Smod}(-7, 5) = -2$$

If x is halfway between two integer multiples of m, then it doesn't matter whether Smod(x, m) is defined to be $m/2$ or $-m/2$.

The algorithm that generates the x_n is given in Fig. 5. This version incorporates a mild error check (lines 10 and 13) to prevent one bad input from spoiling all the subsequent outputs. Appendix A gives a proof that the numbers defined by Eq. (1) are identical to the outputs x_n generated by the basic version of the algorithm, that is, with lines 10 and 13 removed. This is accomplished by forcing the second difference $\Delta^2 x_n$ to be less than $d/2$ in absolute value (in accordance with assumption 1) and also equal to $-\Delta^2 \nu_n$ modulo d. Nevertheless, experience has shown that some protection against bad data is needed. Table 1 shows what can happen to the basic version if there is one error with magnitude slightly greater than $d/4$. The starred value of ddx ($n = 3$) has been converted by Smod from $-0.52d$ to $0.48d$. As a result, the x_n start to increase linearly. This can cause computational problems for a program that analyzes the x_n. The error check with threshold $d/4$ anchors the current bad data to the last good data. If Δx_m is the anchor dxa at time n, and the condition in line 10 is satisfied at times n and $n + 1$, then Δx_{n+1} is within $d/2$ of Δx_m. In the example, the anchor stays at

Δx_1 for $n = 2$ and 3. At $n = 4$, it is pulled up again. As a result, the x_n values go back to 0 after time 2. Time will tell whether this simple expedient is adequate in practice.

IV. Noise-Floor Test

In order to evaluate the technique, a single-channel hardware and software system was set up (Fig. 2). The HP 1000 collects the data in real time and stores the outputs of the unfolding algorithm on disk. At the same time, a user can process any portion of the test into residuals and Allan deviation. Results can be printed, plotted on screen, or plotted on a pen plotter.

To measure the ultimate noise floor of the technique and to test the integrity of the measurement system, an almost perfect beat note square wave was simulated by a low-rate pulse generator, which was stable at the nanosecond level, and whose period could also be programmed to the nearest nanosecond. The chosen period was

$$0.938196601 \text{ s} = (10 - r)d$$

where $d = 0.1$ s, the picket fence period, and r is the Golden Ratio, $(\sqrt{5} - 1)/2$. This period guarantees a good mix of counter readings ν_n [3, pp. 510, 511, 543]. If a conveniently available 1 PPS signal had been used, the counter would have always been reading the same value. The same frequency source, a cesium standard, was used for driving the counter, the picket fence, and the pulse generator. Thus, the results include instabilities and errors of the pulse generator, picket fence divider, and counter, but not of the measurement time base that drives these components.

A test of duration of 108,600 s was carried out. The accumulated time residuals, with the mean frequency offset removed (about 0.3 ns/s), remained within a 6 ns band over the entire run. The Allan deviation, shown in Fig. 6, is approximately $1.3 \times 10^{-9}/\tau$ for τ between 0.94 s and 11,500 s. This shows that all the equipment maintained time coherence at the nanosecond level and that the counter met its specifications.

To see what this means for an actual test of frequency sources, recall that these numbers must be scaled down according to the source and beat frequencies. For example, if two 1 MHz sources with a 1 Hz offset were being tested, the digital part of the measurement system would contribute a $\Delta f/f$ Allan deviation of $1.3 \times 10^{-15}/\tau$.

V. Conclusions

The technique described in this article is a method for measuring the stability of the square wave produced by mixing two offset frequency sources and passing the low-frequency sine wave through a zero-crossing detector. In contrast to the current IFSTF system, which uses custom digital hardware and has microsecond resolution, this technique uses commercial hardware of moderate cost (under $3000 per counter) with an IEEE-488 interface, and offers nanosecond accuracy.

The author has recently learned of the existence of a unit that can latch the readings of a free-running counter with a precision of 1 ns for several input channels. The counter rolls over every 2^{24} ns, about 16.8 ms, so that in effect the counter makes its own picket fence with that period, and the same unfolding algorithm applies. A multichannel frequency stability measurement system built around this unit might be smaller and less expensive than one built around several interval counters in separate chassis.

Acknowledgment

I would like to give credit to G. J. Dick and R. E. Taylor for conversations that led to the ideas used in this article.

References

[1] C. A. Greenhall, "Frequency stability review," *TDA Progress Report 42-88*, October-December 1986, pp. 200-212, Jet Propulsion Laboratory, Pasadena, Calif., February 15, 1987.

[2] T. Y. Otoshi and M. M. Franco, "DSS frequency stability tests performed during May 1985 through March 1986," *TDA Progress Report 42-86*, April-June 1986, pp. 1-14, Jet Propulsion Laboratory, Pasadena, Calif., August 15, 1986.

[3] D. E. Knuth, *The Art of Computer Programming*, vol. 3. Reading, MA: Addison-Wesley, 1973.

Table 1. State of the unfolding algorithm after line 9 (*d* = 1) with one bad input and no error handling

n	*v*	*v*1	*x*1	*dua*	*dxa*	*du*	*ddx*	*dx*	*x*
0	0	line 9 not applicable							0
1	0	0	0	10	0	0	0	0	0
2	−0.26	0	0	0	0	0.26	0.26	0.26	0.26
3	0	−0.26	0.26	0.26	0.26	−0.26	0.48*	0.74	1.00
4	0	0	1.00	−0.26	0.74	0	0.26	1.00	2.00
5	0	0	2.00	0	1.00	0	0	1.00	3.00
6	0	0	3.00	0	1.00	0	0	1.00	4.00

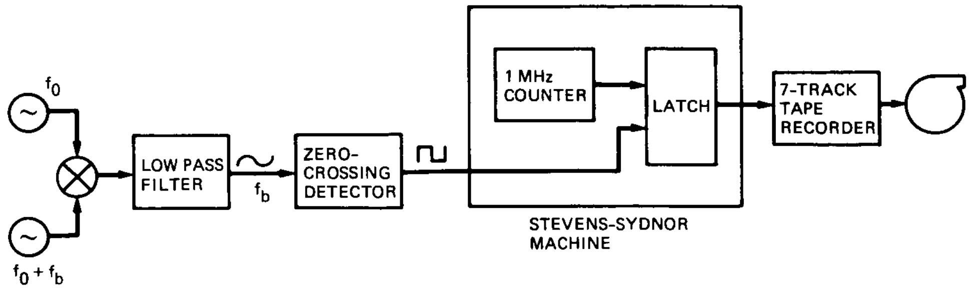

Fig. 1. Current IFSTF frequency stability measurement system

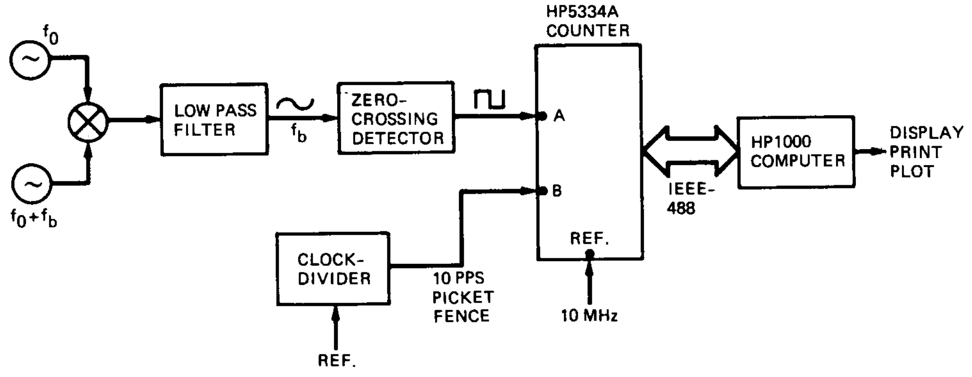

Fig. 2. Picket fence measurement system

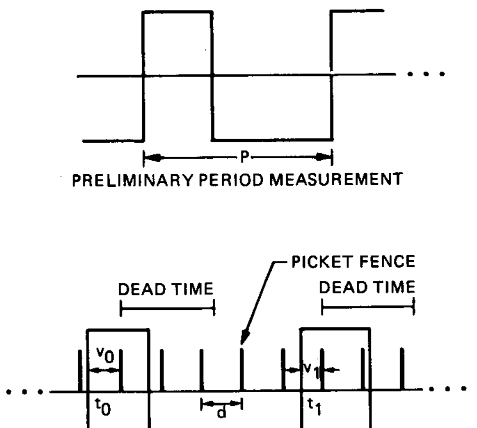

Fig. 3. Time evolution of the measurement process

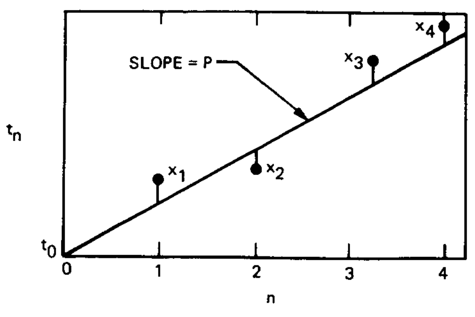

Fig. 4. Zero-crossing times and residuals

```
! Global parameters: p, d

Do for n = 0, 1, 2, ...
     Read vn
     Call Unfold (n, vn, xn)
     Output xn
Enddo
___________________________

Subroutine Unfold (n, v, x)

! Inputs: n, v
! Output: x
! Local variables
!    du : current Δu = -Δv
!    dx : current Δx
!    dua: du anchor
!    dxa: dx anchor
!    ddx: current dx - dxa or second difference
!    v1 : previous v
!    x1 : previous x

1.   If n = 0 then
          ! Initialize
2.        dua = p
3.        dxa = 0
4.        x = 0
5.   Else
6.        du = v1 - v
7.        ddx = Smod (du - dua, d)
8.        dx = dxa + ddx
9.        x = x1 + dx
          ! Error handling
10.       If |ddx| < d/4 then
               ! Data OK, drag anchor along
11.            dxa = dx
12.            dua = du
13.       Endif
14.  Endif
15.  x1 = x
16.  v1 = v
17.  Return
```

Fig. 5. Unfolding algorithm

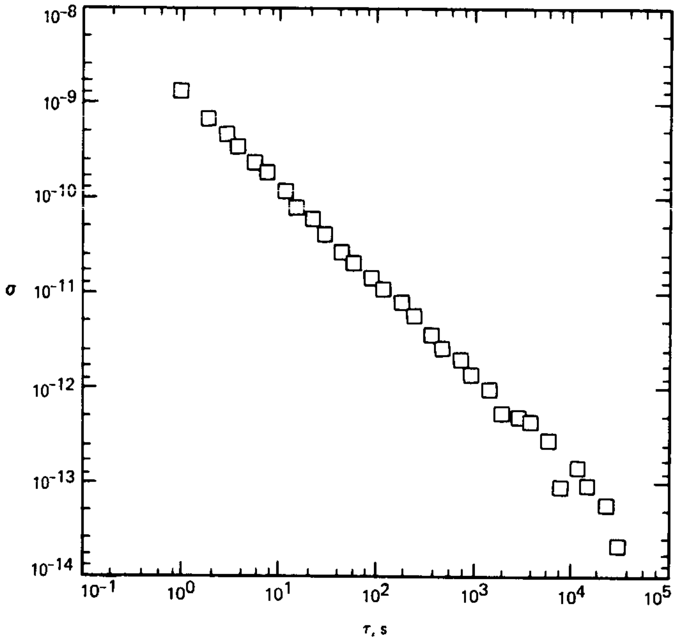

Fig. 6. Allan deviation for a noise floor test on a pulse train with period 0.94 s

Appendix A

Proof of Correctness of the Unfolding Algorithm

To avoid confusion let the output of the algorithm be denoted by X_n, while x_n is still defined by Eq. (1). The error correction lines 10 and 13 are deleted. We shall prove by induction that $X_n = x_n$ for all n. For $n = 0$, both quantities are zero. For $n = 1$ we have

$$\begin{aligned} du &= -\Delta \nu_1 \equiv \Delta t_1 \pmod{d}, \\ ddx &= \mathrm{Smod}(\Delta t_1 - p, d) = \Delta t_1 - p \text{ (assumption 1)} \\ &= x_1 . \end{aligned}$$

Therefore, $X_1 = x = x_1$ since $dxa = x1 = 0$. For $n > 1$ assume the algorithm is correct up to $n - 1$. Then

$$\begin{aligned} dua &= -\Delta \nu_{n-1} \equiv \Delta t_{n-1} \pmod{d}, \\ du &= -\Delta \nu_n \equiv \Delta t_n \pmod{d}, \\ ddx &= \mathrm{Smod}(\Delta^2 t_n, d) = \Delta^2 t_n \text{ (assumption 1)} \\ &= \Delta^2 x_n , \\ dxa &= \Delta x_{n-1}, \ x1 = x_{n-1}, \\ dx &= \Delta x_{n-1} + \Delta^2 x_n = \Delta x_n , \\ X_n &= x = x_{n-1} + \Delta x_n = x_n . \end{aligned}$$