N92-13841

The Role of Time and Speed in NASA's SUNLITE Program

Joseph C. Hafele
Research Professor of Physics
Christopher Newport College
Newport News, VA 23606

The SUNLITE program of NASA's Langley Research Center aims to demonstrate lower noise and better frequency stability for CW solid-state lasers in the microgravity environment of space. The program will utilize laser-diode-pumped non-planar-ring-oscillators regulated by ultra-stable high-finesse Fabry-Perot spectrometers to produce light beams with phase rate or frequency variations as low as 3 Hz. This translates into a stability ratio $\Delta f/fo \sim 1E-15$, where Δf = FWHM Line-Width (Hz) and fo = Line Center Frequency (Hz). Achieving this goal would increase previously achieved solid-state laser coherence lengths from 186 miles (1 light-msec, $\Delta f \sim 1$ kHz) to 62,000 miles (0.3 light-sec, $\Delta f \sim 3$ Hz). Such long coherence lengths would permit more efficient transmission of Earth-satellite and intersatellite communications. It would also permit large interferometric gravity wave antennas, and even interferometric detection of relativistic "inertial-drag" effects on low-Earth-orbit satellites.

SUNLITE will use the "period-method" to measure the phase rate and frequency stability of the lasers. The stability of a voltage, v(t), that is periodic in time, t, could be measured by one of two basic methods: *i)* (FT(V)-method) Measure N voltages {v1,...,vN} at N successive times {t1,...,tN}, apply a preconditioning function V(v) to the raw data, and then apply the Fourier Transform to the preconditioned data to get the spectral distribution of frequencies in V(v). If V = v, the FT(v) gives the spectral distribution of frequencies in v(t), from which the line-width Δf and line-center frequency fo could be calculated (Fig. 1). *ii)* (P-method) Measure M successive periods {p1,...,pM} and reciprocate the periods (fi = 1/pi) to get M frequencies {f1,...,fM}. The mean frequency and Allan Variance can then be calculated from this sequence of frequencies. Notice that the P-method does not involve the Fourier Transform.

The P-method was chosen because it requires less memory space for the raw data (M < 3N), because frequencies can be analyzed on-line in real-time simply by reciprocating the periods (fi = 1/pi), and because the mean and variance of the frequencies can be calculated as fast or faster than they can be with the fastest FFT's. Furthermore, for a given signal-to-noise power ratio, the P-method requires less data and less computer time to extract the noise components. Although the P-method does require fast Time Interval Counters, the FT-method requires comparably fast Sampling Volt Meters. For either method, however, time and computer speed play a critical role.

The simple question "What time is it?" may not be as simple as it seems. According to Newton (*cir.* 1700 AD) "Absolute time flows equably, without relation to anything external", and a clock is only a futile attempt to measure the underlying absolute time. But according to Einstein (*cir.* 1900 AD) "Time is that which is indicated by a clock", and the time "here and now" cannot be separated from the clock located at the place "here" that is used to measure the time at the instant "now". The SUNLITE program is actually concerned with the question: "What is the time ti for the i-th voltage vi, and what is the time step, Δti = ti+1 - ti, between the (i+1)-th voltage and the i-th voltage?" A computer clock will be used to measure these times.

Any clock consists of a periodic or oscillating element, and a memory to count the periods of the oscillator. Digital clocks contain an electronic oscillator and a counter/register which serves as the memory (Fig. 2). The performance of a clock can be determined by comparing it with another clock. Most computers contain two independent clocks, a "system clock" that drives the microprocessor, and a "24-hour time and date clock" that stays on all the time. A source code in BASIC that can be used to study the performance of the system clock "TIMER" is listed under Prog. 1.

The speed of a computer can be expressed in IOPS (Integer Operations Per Second) or in FLOPS (FLoating point Operations Per Second). A source code in BASIC to measure the speed for the integer operation K=K+1 is listed under Prog. 2. This program, in both the uncompiled form (interpreter) and in the compiled form (exe code), was used to measure the speed of various PC's. For 6 different PC's, IOPS ranged between 100 and 7000 K-counts per second. Mean values for 3 personal computers are shown in Fig. 3. Clearly, computer speed depends both on the computer model (hardware) and on the programming (software).

The higher the frequency of the oscillator in a clock, the better is the resolution or time interval between "ticks" of the clock. As important as resolution is uniformity in the time interval between successive ticks. Figure 4 is a graph of the K-counts of Prog. 2 for 260 seconds on a COMPAQ PORTABLEIII laptop computer. The "jitter" up-and-down from the general trend is due to phase noise, while the large step down near 70 seconds is caused by "digit rollover" in the reference clock (Fig.2). In this case, digit rollover caused more variance than did jitter. A more severe case of digit rollover is shown in Fig. 5. In this case, rollover caused a clearly visible "staircase" effect in the K-count.

To minimize the noise components from digitization and rollover, and to maximize memory reliability, SUNLITE will use a nonvolatile 32-bit memory to store the period data.

Fig. 1. Various functions V(v) which could be applied to the periodic voltage v(t) before applying the Fourier Transform.

FT(V)	Precond.Func.	Comment
FT(ω)	$V = v$	Identity Function (raw data)
FT(z)	$V = (v-\langle v\rangle)/\sigma$	Normalized to Standard Deviation
FT(y)	$V = (v-\langle v\rangle)/\langle v\rangle$	Normalized to Mean
FT(power)	$V = v^2$	Power Spectrum
FT(τ)	$V = v(t)v(t-\tau)$	Autocorrelation Function

Fig. 2. Schematic diagram for a 24-hour digital clock. Output of the divider has a frequency of 1 Hz. Digit rollover occurs between successive stages of the memory registers.

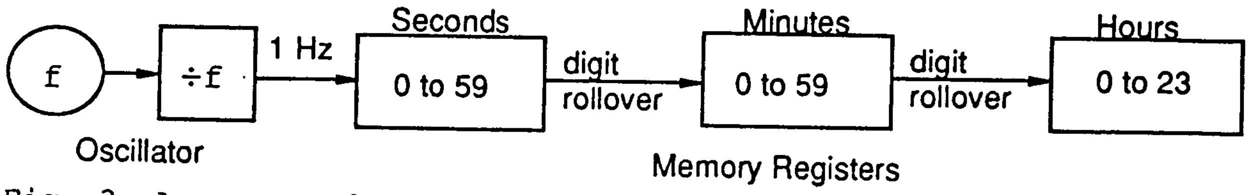
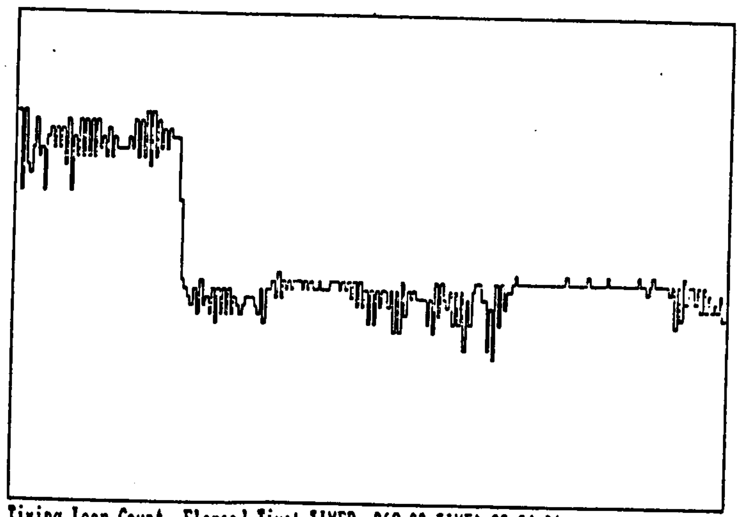


Fig. 3. Average values for the counter operation K=K+1 for various personal computers. The IOPS program listed below (Prog. 2) was used in each case. It was run by means of the GWBASIC interpreter and the QUICKBASIC compiler.

Software	IBM PC AT	MEGA PC AT	COMPAQ PORTIII
GWBASIC(interp.)	500	672	?
QUICKBASIC(comp.)	1292	1720	2512*

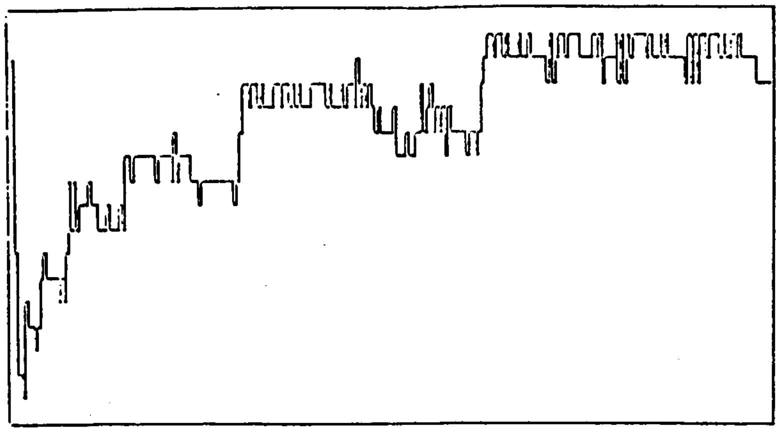
(* A NORTHGATE PC (w/64k nonvolatile cache memory) produced a K-count greater than 7000.)

Fig. 4. Time structure in the IOPS K-count of Prog. 2 for a COMPAQ PORTABLEIII laptop PC. This structure is similar to every other case studied. Total elapsed time (x-axis) = 250 seconds. Minimum K-count = 2501. Maximum K-count = 2529. Mean K-count = 2512.482.



```
Timing Loop Count. Elapsed Time: TIMER  260.89 TIME$ 00:04:21
MaxCount: 2529 MinCount: 2501 Mean: 2512.481927710843
```

Fig. 5. Time structure in the K-count showing an extreme case for digit rollover. This case was measured with an EPSON LQ 510 laptop PC. Elapsed time = 260 seconds. Mean K-count = 626.488. Minimum K-count = 615. Maximum K-count = 630.



```
62486764200.000 = Start          Seconds(AD)          Stop = 62486764460.950
StartDATE$TIME$Set 01-01-1980:00:00:00 NumXvalues= 249 MeanXvalue=  626.4879
MinXvalue =   615    MaxXvalue =   630      ElapsedTime (seconds) =   260.950
```

Prog. 1. Source code in BASIC to study the performance of the system clock TIMER. After setting the 24-hour clock to zero(line 20), the system clock is synchronized to it (line 30). Successive values for TIMER are recorded (line 30) in the array T(I).

```
10    CLEAR:DEFINT I-K:OPTION BASE 1:DIM T(1000)
20    TIME$="0:0:0":T=TIMER
30    WHILE TIMER=T:WEND
30 FOR I=1 TO 1000:T(I)=TIMER:NEXT I
40    FOR J=1 TO 1000
50      PRINT J,T(J)
60    NEXT J
70    END
```

Prog. 2. Source code in BASIC to measure a computer's speed. Line 40 counts the number of K=K+1 operations the computer can execute in successive 1-second time intervals IOPS. FLOPS could be measured by substituting X=X+1 in the WHILE/WEND loop (line 40).

```
10   CLEAR:DEFINT C,I-K:OPTION BASE 1:DIM C(100)
20   TIME$="0:0:0"
30   FOR I=1 TO 100: K=0: T=TIMER+1
40     WHILE TIMER<T: K=K+1: WEND
50     C(I)=K
60   NEXT I
70   FOR J=1 TO 100:PRINT J,C(J):NEXT J
80   END
```