Monte Carlo Simulations of Precise Timekeeping in the Milstar Communication Satellite System

J. C. Camparo and R. P. Frueholz
Electronics Technology Center
The Aerospace Corporation
PO Box 92957, Los Angeles, CA 90009

Abstract

The Milstar communications satellite system will provide secure antijam communication capabilities for DoD operations into the next century. In order to accomplish this task, the Milstar system will employ precise timekeeping on its satellites and at its ground control stations. The constellation will consist of four satellites in geosynchronous orbit, each carrying a set of four rubidium (Rb) atomic clocks. Several times a day, during normal operation, the Mission Control Element (MCE) will collect timing information from the constellation, and after several days use this information to update the time and frequency of the satellite clocks. The MCE will maintain precise time with a cesium (Cs) atomic clock, synchronized to UTC(USNO) via a GPS receiver. We have developed a Monte Carlo simulation of Milstar's space segment timekeeping. The simulation includes the effects of: uplink/downlink time transfer noise, satellite crosslink time transfer noise, satellite diurnal temperature variations, satellite and ground station atomic clock noise, and also quantization limits regarding satellite time and frequency corrections. The Monte Carlo simulation capability has proven to be an invaluable tool in assessing the performance characteristics of various timekeeping algorithms proposed for Milstar, and also in highlighting the timekeeping capabilities of the system. Here, we provide a brief overview of the basic Milstar timekeeping architecture as it is presently envisioned. We then describe the Monte Carlo simulation of space segment timekeeping, and provide examples of the simulation's efficacy in resolving timekeeping issues.

Introduction

Figure 1 shows the baseline timekeeping architecture for Milstar as presently envisioned. The constellation will consist of four satellites in geosynchronous orbit[1], each carrying a set of four rubidium (Rb) atomic clocks, though at any one time only one clock will be operational on any given satellite. A satellite's active clock is labeled as either master (MSR), monitor (MON) or slave. The slave clock ties its time and oscillator frequency to the master via timing comparisons performed through the satellite crosslinks using a slaving procedure developed by Lockheed (the Milstar prime contractor)[2]. The monitor clocks are free-running, and are present in order to assess the health of the MSR again via the satellite crosslinks. Several times a day, during normal operation, the Mission Control Element (MCE) collects timing information on the Triplet of free-running clocks (i.e., MSR, MON1 and MON2), and after several days uses

this timing information to update the time and oscillator frequencies of the Triplet. The MCE maintains precise time with a cesium (Cs) atomic clock, which is synchronized to UTC.

In outline, Milstar timekeeping would appear to be straightforward and robust; however, in detail precise Milstar timekeeping is a complex matter. The time comparisons between satellites via the satellite crosslinks, and those using the uplink/downlink between the inview satellite and the MCE, are not perfect: un-accounted for equipment delays can introduce non-negligible timing errors into the system. Moreover, even if the communications links were perfect, there are limits as to the accuracy with which time and oscillator frequency corrections may be applied to the satellite clocks. These limits are a consequence of both the satellite hardware and Milstar operating procedures. Additionally, the diurnal temperature variations that the satellites experience introduce timing errors as a consequence of the Rb atomic clock's (albeit slight) temperature sensitivity[3]. Though individually these processes are straightforward, with regard to system timekeeping they act together in non-obvious ways as part of a "satellite-to-MCE feedback loop": these processes cause time differences between the satellite and MCE, which the MCE attempts to correct periodically. Finally, it must be recognized that even though the satellite Rb atomic clocks will introduce no more than about 4 μs of timing error into the system in a week[4], this requires the MCE to set them perfectly. As a consequence of these considerations, it should be recognized that cursory analyses of timekeeping performance may neglect important subtleties, and could lead to incorrect conclusions.

In order to accurately address system level timekeeping issues, several approaches may be taken. First, one might consider developing a hardware prototype of system timekeeping. This approach is impractical not only because it requires a large capital outlay for the various pieces of equipment, but also because investigations into system timekeeping over periods of months would have to be done in real time. Alternatively, one could attempt to solve the satellite-to-MCE feedback loop equations. This too is an impractical approach, because closed form solutions could not be obtained without significant approximation. Moreover, altering system characteristics slightly (e.g., system algorithms) could force a re-derivation of the entire set of feedback loop equations, requiring significant amounts of additional effort. Our approach to answering system level timekeeping questions has none of the above mentioned drawbacks, as it is based on Monte Carlo simulation[5]. With a Monte Carlo approach, the results are obtained without approximation; years of system timekeeping experience can be built up over the course of several hours, and changing system algorithms requires nothing more than the change of a subroutine.

Figure 2 is a functional diagram of the Monte Carlo concept, illustrating some of the important components of this simulation capability. The studies to be discussed below have focussed on the MCE's management of space timekeeping assets, and the performance of those assets under varied operational conditions. Generally, however, Milstar timekeeping also includes the process of synchronizing Milstar time, which is maintained at the MCEs, to UTC which is maintained by the Naval Observatory for DoD programs. Synchronizing Milstar time to UTC should be straightforward, and hence not require detailed Monte Carlo simulations for the resolution of timekeeping issues.

In the analysis of system timekeeping, we start by generating a time series of random frequency fluctuations for both a satellite and MCE atomic clock[6,7]. Additionally, whenever timing

comparisons take place between clocks, we simulate the appropriate communication link time–transfer noise (i.e., either uplink/downlink or crosslink), and make allowances for any limitations as to timekeeping corrections. Finally, we include in the simulation the diurnal temperature variations that a satellite clock might experience, and the resulting diurnal frequency variations. All of these stochastic and deterministic process realizations are generated in a 486–PC, and frequency variations are integrated and combined with other timing errors. The output of a single simulation is the satellite time error as a function of time, and this can be obtained for any one of the four satellite clocks (i.e., MSR, MON or Slave). By performing thousands of these simulations we generate statistics on Milstar's timekeeping performance.

Simulation of Atomic Clock Noise

The success of a Monte Carlo analysis of system timekeeping requires the accurate simulation of various timekeeping fluctuations, and in this regard one of the most significant challenges is the simulation of an atomic clock's colored (i.e., flicker and random–walk) frequency fluctuations. The approach we employ may be referred to as a "recursive filter" approach[6], and is best described by considering the spectral density of an atomic clock's random processes. Experimentally, if one had white noise, and one wanted to turn this into colored noise, then one would simply pass the white noise through a filter. The filter function would then shape the noise process's spectral density into some desired form. This is essentially the method we employ for simulating colored noise processes as illustrated in Fig. 3[8].

In order to simulate a noise process with a spectral density that is an even function of Fourier frequency f, we start with computer generated random numbers. These numbers have a uniform probability distribution, but may be transformed into random numbers with a normal (i.e., gaussian) probability distribution using the standard Box–Mueller algorithm[9]. At this point, we have a simulation of a gaussian white noise process. These numbers are input to a numerical filter, described by a transfer function $H(f)$, and the spectral density of the filter output is $|H(f)|^2$. Thus, to simulate random–walk noise we just need to choose $H(f) \sim 1/f$.

Simulating a noise process that is an odd function of Fourier frequency is a bit trickier, as $H(f)$ would then have to be a function of Fourier frequency to some fractional power. (If $H(f)$ is a rational function, then the inverse of $H(f)$ can be found by the method of partial fractions.) Since the MCE's Cs atomic clock noise has a flicker noise component, this portion of the simulation is important for properly modeling the MCE's timekeeping capability. Simulating noise processes with [INSERT 3] may be accomplished by cascading filters that are integral functions of Fourier frequency.[6] By a judicious choice of filter functions, the cascade can be made to approximate an overall filter that is not a rational function of Fourier frequency, which in turn yields an [INSERT 4] that is (approximately) an odd function of Fourier frequency

As a final point, it should be mentioned that in deriving the equations for the recursive filter, it is assumed that the filter's operation is in steady–state. This is tantamount to assuming that the filter has been processing data since $t = -\infty$. The fact that the recursive filter must be started at some finite time in the Monte Carlo simulations is called the "Initialization Problem."[10] Though a technical description of this problem and its solution is beyond the scope of the present discussion, suffice it to say that if the Initialization Problem is not handled properly,

the accuracy of system timekeeping simulations would have to be called into question. In the present simulations we include initialization of both the satellite Rb atomic clocks and the MCE's Cs atomic clock.

An example of our capability to simulate colored atomic frequency standard noise is illustrated in Fig. 4. Using the method outlined above, we simulated the frequency fluctuations that are expected for a Milstar satellite Rb atomic clock. We then performed an Allan standard deviation calculation on these simulated frequency fluctuations, and the results are shown as boxes in Fig. 4. The solid line represents the expected Allan standard deviation for the satellite Rb atomic clocks based on clock manufacturer data. Clearly, the agreement between our simulated frequency fluctuations and those truly generated by the Milstar satellite atomic clock is excellent.

Figure 4 represents only one validation test for our Monte Carlo simulation of Milstar timekeeping. However, at every stage in the development of the Monte Carlo simulation, tests were performed to establish the simulation's verity. These tests included an accurate simulation of the MCE's cesium atomic clock, specifically its flicker noise component, and a demonstration that the simulation would generate expected results under well defined, though not necessarily Milstar accurate, conditions.

Applications

The Monte Carlo simulation of Milstar timekeeping outlined above includes the full range of timekeeping processes and elements associated with the MCE's management of Space Segment assets, and it has been extensively exercised to address topics in both the single and multi–satellite environments. In this section we provide examples of those applications. The first of the examples concerns work that was performed several years ago when the question of how the MCE would estimate satellite time and frequency offsets was unanswered. This example will illustrate how various system algorithms can be easily changed and examined for their effect on overall system timekeeping using a Monte Carlo approach. The second example deals with the question of how satellite temperature variations influence precise satellite timekeeping. This latter example illustrates the complicated fashion in which various processes combine to produce a non–obvious dependence of timekeeping capability on system parameters.

A. MCE Estimation Algorithms

As discussed in the general description of Milstar timekeeping, the MCE will determine the time offsets of all the satellites in the constellation via the inview satellite and crosslinked data. This timing information will then be used by the MCE in an estimation algorithm in order to determine the time and frequency corrections that need to be supplied to the various free–running (i.e., Triplet) satellite clocks. One of the major timekeeping questions faced by Milstar system planners in the mid–eighties concerned the form that the estimation algorithm would take.

Figure 5 illustrates an MCE ranging on an inview satellite, and the timekeeping data that the

MCE would collect (i.e., satellite time error as a function of measurement time). The time error collected by the MCE will have the general form:

$$x(\tau) = x_0 + y_0\tau + \frac{1}{2}D\tau^2 + \alpha\int_0^\tau [T(t,\theta) - T_0]dt + \int_0^\tau y_\tau^{\text{sat}}(\tau)dt + \int_0^\tau y_\tau^{\text{MCE}}(\tau)dt + \epsilon(\tau) \quad (1)$$

Here, $cx(\tau)$ is the time offset between the satellite and MCE at some time τ, x_0 is an initial time offset, y_0 is a constant fractional frequency difference between the satellite Rb clock and the MCE Cs clock, D is the fractional frequency aging rate of the satellite Rb clock (parts in 10^{13} per day[11], α is the temperature coefficient of the satellite clock, $T(t,\theta) - T_0$ is the diurnal temperature offset of the satellite clock from some nominal value, T_0, y_τ^{sat} and y_τ^{MCE} represent the random fractional frequency fluctuations of the satellite and MCE clocks, respectively, and $\epsilon(\tau)$ is the measurement error associated with the MCE–to–spacecraft communication link. The parameter θ in the satellite temperature term represents the phase relationship between the satellite's diurnal temperature cycle and the cycle of MCE corrections. The question addressed with our Monte Carlo simulation, was how the MCE could best use the time error data presented in Fig. 5 to periodically correct the satellite time and frequency. In the following, the update interval will be defined as the period of time between MCE corrections of the satellite clock.

On an examination of Eq. (1) for $x(\tau)$, several possibilities for employing the time error data of Fig. 5 present themselves. First, the MCE could restrict its consideration to data collected only at the beginning and end of an update interval. The time error at the end of the update interval would then be the time correction that the MCE needs to apply (δt), while the frequency correction (δy) would come from the estimated rate of time error build up based on the two time error measurements. If T_{update} is the length of the update interval, then the time and fractional frequency corrections to be applied by the MCE are:

$$\delta t = xT_{\text{update}} \quad (2)$$

$$\delta y = \frac{x(T_{\text{update}}) - x(0)}{T_{\text{update}}} \quad (3)$$

This is called the 2–Point estimation algorithm, and has the advantage of being very simple. An alternate procedure would be to take advantage of all the intervening data collected by the MCE during the update interval. The data could then be fit to a straight line in order to determine the appropriate time and frequency corrections:

$$\delta t = \delta y \cdot T_{\text{update}} + t_0 \quad (4)$$

Here, δy and t_0 are the slope and intercept determined by the linear least squares. This is called the Linear estimation algorithm, and it is to be noted that the frequency correction is determined by the slope of the linear least squares fit. Finally, by examining the above equation

for $x(\tau)$, one might expect to do better at correcting the clock by fitting the data to a quadratic, which would essentially be attempting to account for the Rb clock's aging rate:

$$\delta t = t_0 + \hat{y} \cdot T_{\text{update}} + \frac{1}{2}\hat{D} \cdot T_{\text{update}}^2 \tag{5}$$

Here, $\hat{y}$ is the linear coefficient of the least squares quadratic fit, which is essentially the initial frequency offset of the clock, and $\hat{D}$ is the least squares estimate of the aging rate of the clock. This is called the Quadratic estimation algorithm.

Using our Monte Carlo simulation of Milstar timekeeping, we were able to investigate the performance of each of these estimation algorithms[12]. The parameters that were employed in the calculations are collected in Table I. To determine the efficacy of any estimation algorithm, we allowed the MCE to correct the satellite clock several times, essentially letting the system get into a steady state, and then examined the satellite time error after either 3 or 10 days of free–running operation. (Note from Table I that a 3 day free–running period corresponds to the time error the satellite would have just prior to receiving its normal MCE correction.) Hundreds of simulations were performed (each with a different satellite clock aging rate) to generate the statistics of Milstar timekeeping, and the results of that analysis are collected in Table II. In the table, the standard deviation of time error at the end of the free–running period is tabulated for the various estimation algorithms. Since the Linear estimation algorithm minimizes the spread of satellite time error, it is considered to be the best estimation algorithm among these three. Similar results comparing the Linear estimation algorithm against a Kalman Filter estimation algorithm eventually lead to the adoption of the Linear estimation algorithm for the Milstar MCEs due to its simplicity.

The fact that the Linear estimation algorithm is superior to the Quadratic estimation algorithm was initially something of a surprise. Since the Quadratic estimation algorithm more closely models the underlying performance of the satellite Rb atomic clock, one would typically expect it to result in less timing error. After some study of this issue, we found that the poor performance of the Quadratic algorithm derives from the influence of the measurement noise, $\epsilon(\tau)$, and the Rb atomic clock frequency noise, y_τ^{sat}, on the estimated coefficients. Apparently, these noise processes strongly influence the estimated drift coefficient in the Quadratic algorithm, and of course any error in that estimate has a strong influence on timekeeping since it contributes to time error quadratically.

B. Satellite Temperature Variations and MCE Control of the Satellite Clock

As any Milstar satellite orbits the Earth, its temperature will vary in a diurnal fashion, and in the mid–eighties thermal analysis of the satellite payload indicated that the satellite clock would experience peak–to–peak temperature variations of $\sim 20°\text{F}$. The question arose as to how these temperature variations would influence satellite timekeeping, both for the crystal oscillator that would be launched on DFS–1 (the first Milstar satellite) and the Rb atomic clocks that would be launched on subsequent satellites. Specifically, there was interest at the time in knowing

how large the satellite oscillator's temperature coefficient could get without impacting system timekeeping performance.

Clearly, the MCE could choose to set up its cycle of satellite corrections anywhere within the satellite's diurnal temperature cycle. The quantity expressing this relationship in Eq. (1) is θ. For example, the MCE could choose to correct the satellite clock when the satellite temperature is near its largest daily value; this would correspond to a value of $\theta = 0$ in Eq. (1). Alternatively, the MCE could choose to correct the satellite clock when the satellite temperature is near its daily mid–range value; this would correspond to a value of $\theta = \pi/2$ in Eq. (1). (For the reader's general information, analysis has shown that the diurnal temperature variations will be roughly sinusoidal. We note, however, that our calculations employ the expected diurnal temperature variations and not a sinusoidal approximation.) Thus, in order to study the influence of a satellite oscillator's temperature coefficient on system timekeeping, it is necessary to specify θ. Since the actual value of θ for any given satellite is an arbitrary quantity, we performed two sets of analyses, one with $\theta = 0$ and the other with $\theta = \pi/2$.

Parameters for one illustrative study are collected in Table III, corresponding to a satellite clock with characteristics very near those of a crystal oscillator clock. As discussed in the previous example, our method was to allow the MCE to update the satellite clock through several update intervals, essentially reaching a steady–state of timekeeping, and then to calculate the satellite time offset at the end of a free–running period. For the case under discussion, the free–running period was chosen to be 24 hours (i.e., the update interval). Again, hundreds of simulations were performed, which allowed us to generate the statistics of Milstar system timekeeping, and the results are shown in Fig. 6. In the figure, the 2 σ time error at the end of 24 hours is plotted as a function of the satellite clock temperature coefficient. Two curves are shown, one with the diurnal phase angle $\theta = 0$ and the other with $\theta = \pi/2$.

It is clear from the figure that there is a dependence of Milstar timekeeping on θ. Though the strength of this dependence was unexpected, it could be rationalized as a consequence of optimally choosing the data points employed by the MCE's estimation algorithm. More surprising, however, were the specific results for $\theta = \pi/2$, where the satellite time error is actually found to be a decreasing function of clock temperature sensitivity (at least for temperature coefficients less than about $1 \times 1 -^{11}$ /°C). It would appear that for $\theta = \pi/2$, Milstar system performance is enhanced by having a clock with a slightly larger temperature coefficient. This counter–intuitive result indicates that under certain conditions the effects of the diurnal temperature variations on the Linear estimation algorithm can (to some extent) compensate for the frequency aging of the standard. With regard to the question that motivated these studies, the results of Fig. 6 indicate that the satellite clock temperature coefficients can take on values up to $\sim 1 \times 10^{-11}$/°C (for arbitrary θ without significantly changing Milstar system timekeeping. This value is large, and indicates that the Milstar constellation can be made relatively robust to satellite diurnal temperature variations. Moreover, if the MCE judiciously chooses the correction cycle for the satellites under its control, then the diurnal temperature variations might actually be beneficial to Milstar timekeeping.

Taking a broader view of the results shown in Fig. 6, these Monte Carlo simulations demonstrate the complicated interplay among: satellite temperature variations, communication link time–transfer noise, frequency aging rates, and all the other parameters that are important to satellite

timekeeping. The relationship between system–time–error, satellite–oscillator–temperature–coefficient and [INSERT 17] was not obvious prior to the Monte Carlo computations. Even now, knowing that the relationship exists, it is not obvious what the optimum θ value is for the MCE's estimation algorithm. The important lesson to be learned is that intuitive predictions of satellite timekeeping performance must be accepted warily. How all the various timekeeping processes combine to yield the system performance is not always obvious, and in this regard a Monte Carlo simulation of system timekeeping has great value.

Summary

The above discussion has reviewed a Monte Carlo simulation of Milstar timekeeping. Given the complexity of Milstar timekeeping issues, our experience with these simulations has shown that many results are non–intuitive, and that without a Monte Carlo simulation capability accurate predictions of system performance would be exceedingly difficult (if not impossible) to obtain. Though the simulation capability was developed with Milstar in mind, the capability is fairly general, and could easily be applied to timekeeping issues associated with other satellite systems, for example GPS.

Acknowledgment

Over the course of the past six years our efforts have been supported by various organizations within The Aerospace Corporation's Milsatcom Program Offices. The authors would especially like to acknowledge: R. Covey, J. Cox, A. Dubin, A. Grossman, R. Meis, and S. Sokolsky. The encouragement and support of these individuals has been greatly appreciated.

References

[1] J. Fawcette, Milstar: *"Hotline in the Sky"*, High Technology, Nov., 62–67 (1983); J. Schultz, *"Milstar to Close Dangereous C3I Gap"*, Defense Electronics, March, 46–59 (1983).

[2] Greg Hively and George Keirns, *"The Predicted Performance of the Control Algorithms Used to Synchronize the MILSTAR Time and Frequency Standards (U)"*, 15 July 1989 (Presented at the 1989 MILCOM Conference.)

[3] J. Vanier, R. Kunski, P. Paulin, M. Tetu, and N. Cyr, *"On the Light Shift in Optical Pumping of Rubidium 87: The Techniques of "Separated" and "Integrated" Hyperfine Filtering"*, Can. J. Phys. 60, 1396 (1982).

[4] C. Audoin and J. Vanier, *"Atomic Frequency Standards and Clocks"*, J. Phys. E 9, 697 (1976).

[5] See also: P. Kartaschoff, *"Computer Simulation of the Conventional Clock Model"*, IEEE Trans. Instrum. Meas. IM–28, 193 (1979).

[6] M. J. Levin, *"Generation of Sampled Gaussian Time Series Having a Specified Correlation Function"*, IRE Trans. Inform. Theory IT–6, 545 (1960); J. A. Barnes and S. Jarvis, Jr., *"Efficient Numerical and Analog Modeling of Flicker Noise Processes"*, NBS Technical Note 604 (US Government Printing Office, Washington DC, 1971); J. S. Meditch, *"Clock Error Models for Simulation and Estimation"*, Aerospace Technical Report TOR–0076(6474–01)–2, (The Aerospace Corporation, El Segundo, CA, 1975); J. S. Meditch and W. A. Feess, *"Performance Limits in Clock Error Prediction"*, Aerospace Technical Report TOR–0077(2475–10)–1, (The Aerospace Corporation, El Segundo, CA, 1977); S. M. Kay, *"Efficient Generation of Colored Noise"*, Proc. IEEE 69, 480 (1981).

[7] For an alternate technique of simulating clock noise, see: N. J. Kasdin and T. Walter, *"Discrete Simulation of Power Law Noise"*, in Proc. 1992 IEEE Frequency Control Symposium (IEEE, Piscataway, NJ, 1992) pp. 274–283.

[8] J. C. Camparo and P. Lambropoulos, *"Monte Carlo Simulation of Field Fluctuations in Strongly Driven Resonant Transitions"*, Phys. Rev. A 47, 480 (1993).

[9] J. H. Ahrens and U. Dieter, *"Computer Methods for Sampling from the Exponential and Normal Distributions"*, Commun. ACM 15, 873 (1972).

[10] C. A. Greenhall, *"Initializing a Flicker–Noise Generator"*, IEEE Trans. Instrum. and Meas. IM–35, 222 (1986); R. F. Fox, *"Numerical Simulations of Stochastic Differential Equations"*, J. Stat. Phys. 54, 1353 (1989).

[11] J. C. Camparo, *"A Partial Analysis of Drift in the Rubidium Gas Cell Atomic Frequency Standard"*, in Proceedings of the 18th Annual Precise Time and Time Interval (PTTI) Applications and Planning Meeting, Washington D. C. , 1986, pp. 565–588.

[12] Personnel at Lockheed (the Milstar prime contractor) investigated the possibility of employing a Kalman filter as an estimation algorithm.

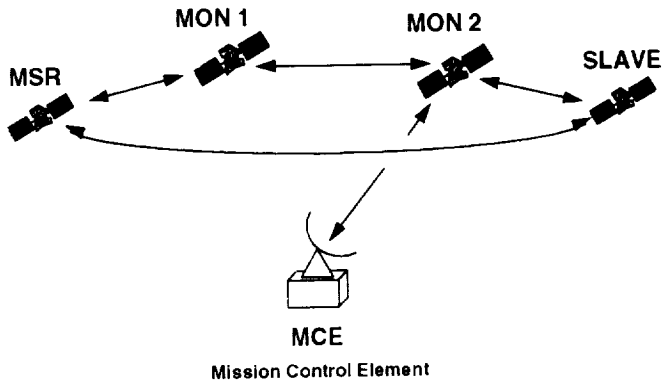

Figure 1: Operational diagram of the baseline Milstar timekeeping architecture. As discussed in the text, the constellation will consist of four satellites labeled: MSR (master), MON (monitor) or slave. The Mission Control Element (MCE) will periodically correct the time and oscillator frequency of the MSR and MONs.

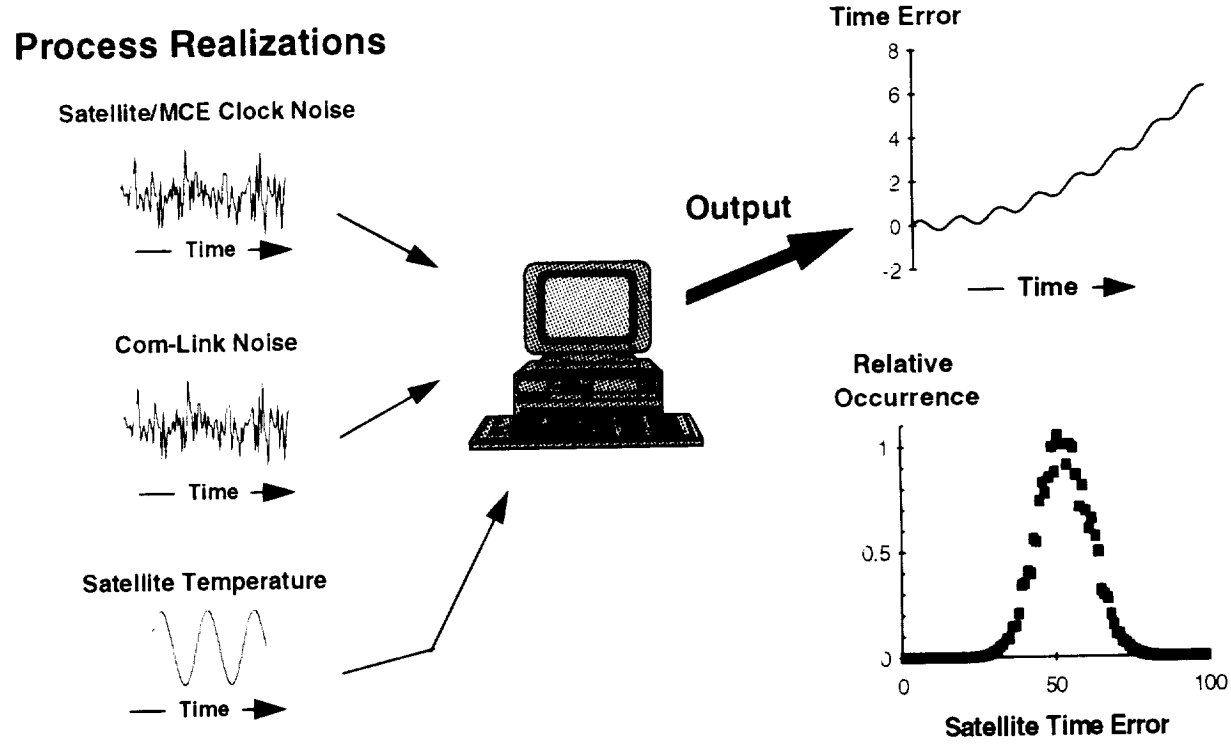

Figure 2: In the Monte Carlo simulation of Milstar timekeeping, realizations of *random* timekeeping processes as well as *deterministic* processes (e.g., satellite temperature variations) are generated. These fluctuations are combined to generate a single realization of a satellite clock's time-error history. By examining thousands of such simulations, the statistics associated with any clock's timekeeping performance may be built up for any set of parameters or operating scenario.

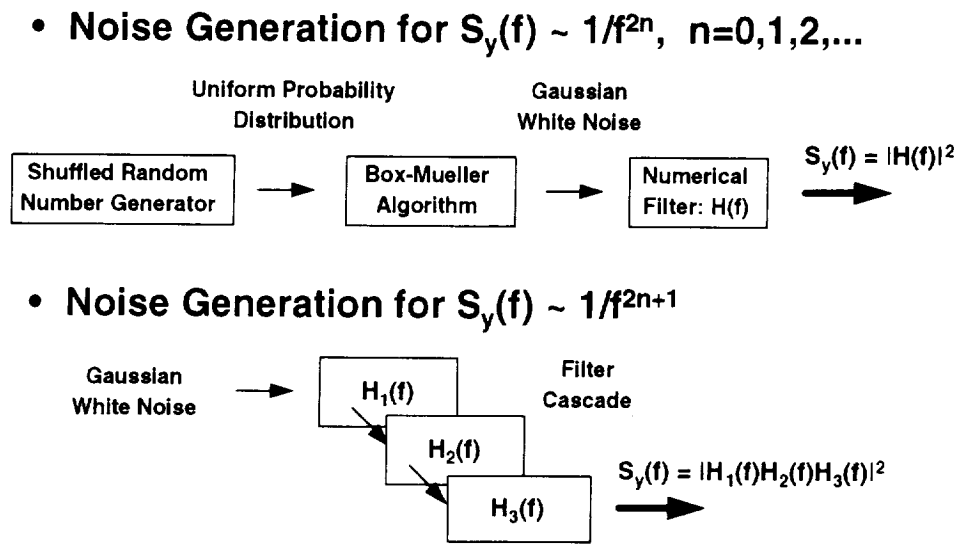

Figure 3: Method of simulating colored atomic frequency standard noise as discussed in the text.

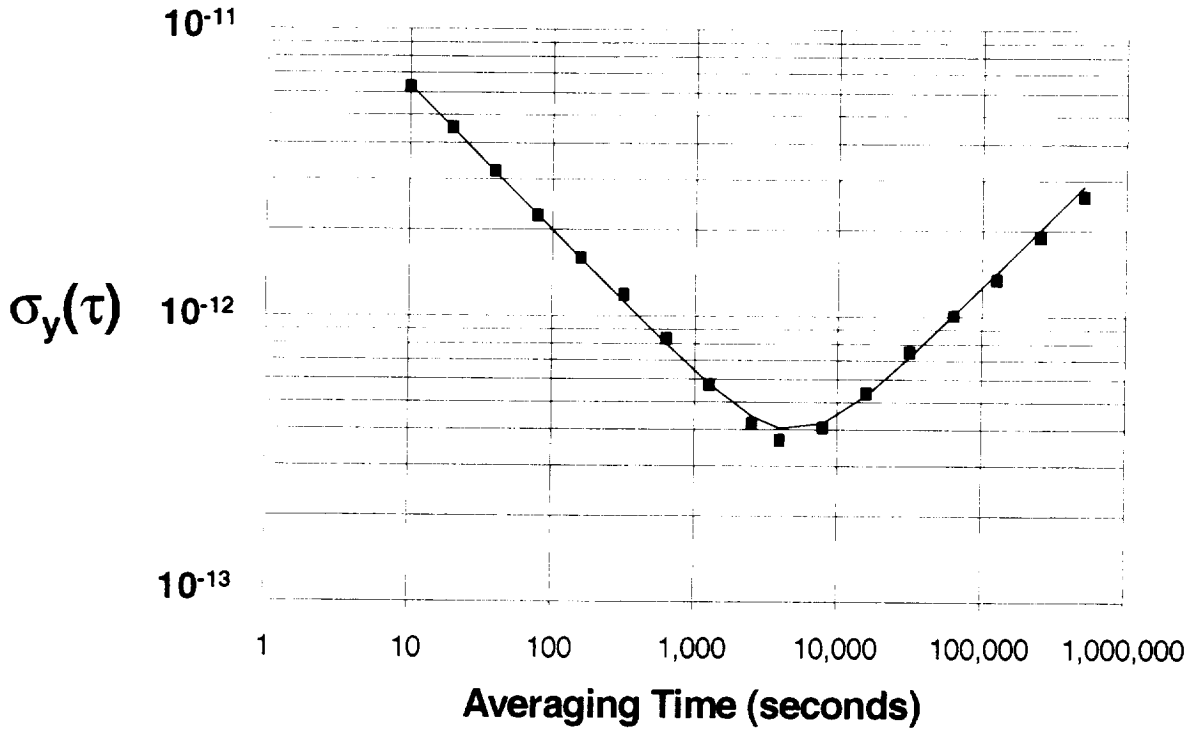

Figure 4: Allan standard deviation plot. The squares correspond to the Allan standard deviation obtained by analyzing the frequency fluctuations simulated by our Monte Carlo program for a Milstar satellite Rb atomic clock. The solid line represents the expected Allan standard deviation based on Milstar clock manufacturer data.

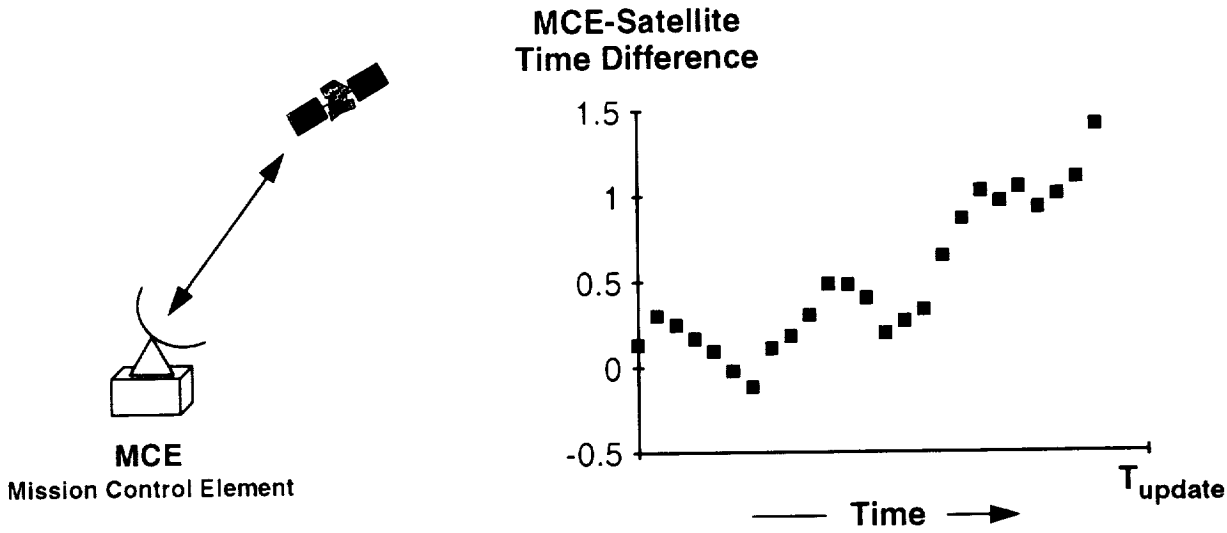

Figure 5: MCE ranging on an inview satellite and collecting time difference information. The MCE-to-Satellite time difference information is used to determine the time and frequency correction that the MCE should apply to the satellite.

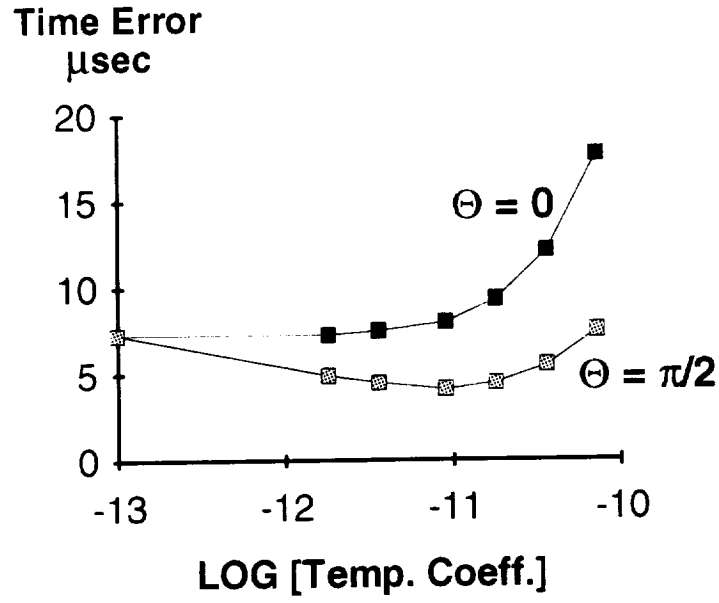

Figure 6: Satellite time error after 24 hours of free-running operation. Temperature coefficients for the satellite clock are per degree Celsius. The two curves labeled $\theta = 0$ and $\theta = \pi/2$ correspond to different phase relationships between the satellite's diurnal temperature cycle and the MCE's satellite correction cycle.

Table I: Parameters used in the Monte Carlo simulation of Milstar timekeeping for the question of which estimation algorithm is best for the Milstar system.

Parameter	Value
Satellite Rb Clock Allan Standard Deviation	$2\times10^{-11}/\sqrt{\tau} + 4\times10^{-15}\sqrt{\tau}$
Satellite Rb Clock Frequency Aging	$0.0 \pm 5.0\times10^{-13}$/day
Satellite Rb Clock Temperature Coefficient	$1.0\times10^{-12}/^{o}F$
Diurnal Temperature Variation Phase Angle, θ	0.0
Update Interval, T_{update}	3 days
MCE-to-Satellite Measurement Interval	8 hours

Table II: Results from Monte Carlo analysis of MCE estimation algorithms. The results show the standard deviation in microseconds of satellite time error at the end of a 3 day and 10 day free-running period.

Estimation Algorithm	3-Day SD	10-Day SD
2-Point	2.5	7.4
Linear	2.2	6.7
Quadratic	4.6	14.3

Table III: Parameters used in the Monte Carlo simulation of Milstar timekeeping for the question of how satellite temperature variations would influence satellite timekeeping.

Parameter	Value
Satellite Clock Allan Standard Deviation	$5\times10^{-13}/\sqrt{\tau} + 5\times10^{-14}\sqrt{\tau}$
Satellite Clock Frequency Aging	$2.0 \pm 0.5\times10^{-11}$/day
Satellite Clock Temperature Coefficient	0.0 to $4.0\times10^{-11}/^{o}F$
Diurnal Temperature Variation Phase, θ	0.0 and $\pi/2$ radians
Update Interval, T_{update}	24 hours
MCE-to-Satellite Measurement Interval	2 hours
MCE Estimation Algorithm	Linear