Numerical Method for Rapidly Determining Satellite-Satellite and Satellite-Ground Station In-View Periods

J.A. Lawton*
Naval Surface Weapons Center, Dahlgren, Virginia

An algorithm that finds, for low-eccentricity satellite orbits, the rise and set times of in-view periods for satellite-satellite and satellite-ground station comparisons is developed. The algorithm is much faster than comparable brute force methods. The method of the algorithm consists of finding the period P of the visibility function v and stepping through time in fractions of P to locate rise and set times. This procedure is continued until all in-view periods are found. Numerical tests show that the algorithm has a 99.99% success rate for finding in-view periods for both spherical-Earth and oblate-Earth gravity models, with a 95% decrease in computer time over the brute force method. A further test shows the potential for success of the oblate-Earth version in an operational environment.

Introduction

To date, satellite-satellite and satellite-ground station in-view periods have usually been computed using the brute force method. In this method, time is stepped through in a given increment, and for each new time, a simple check is done on position vectors to determine if the components can "see" each other. The brute force method, however, can be undesirable when accurate encounter times are required in short computer turnaround time because it takes much computer time to yield these accurate results.

The present algorithm is much faster than an accurate brute force method. A method related to the present algorithm is described by Parks,[1] where satellite-ground station in-view periods are predicted by referring to Δ, the vertical distance above a plane tangent to the ground station. If the vertical distance is positive, the satellite is in-view; otherwise, it is not. The parameter Δ as a function of time is a nicely behaved sinusoidal function, and iteration methods quickly determine rise and set times. The present algorithm extends this idea of using a visibility function analagous to Δ, and, using numerical methods on the function to determine in-view time parameters, for working with satellite-satellite and satellite-ground station configurations, all with one algorithm. The algorithm yields good results for low-eccentricity orbit satellites ($e \leq 0.1$) for either spherical-Earth or oblate-Earth gravity fields, and yields a savings of about 95% in computer time over a comparable brute force method.

The Visibility Function and Its Manipulation

The visibility function v used in the algorithm is defined by

$$v = \alpha_1 + \alpha_2 - \phi \tag{1}$$

where

$$\alpha_1 = \cos^{-1}\frac{R'_E}{r_1}$$

$$\alpha_2 = \cos^{-1}\frac{R'_E}{r_2}$$

$$R'_E = R_E + h_g$$

R_E = radius of Earth

h_g = grazing altitude

$$\phi = \cos^{-1}\frac{\bar{r}_1 \cdot \bar{r}_2}{r_1 r_2}$$

$\bar{r}_1$ = position vector of satellite 1

$r_1 = |\bar{r}_1|$

$\bar{r}_2$ = position vector of satellite 2

$r_2 = |\bar{r}_2|$

The grazing altitude h_g is used to expand the Earth horizon for sighting over the atmosphere.

The first time derivative of v is given by

$$\dot{v} = \dot{\alpha}_1 + \dot{\alpha}_2 - \dot{\phi} \tag{2}$$

where

$$\dot{\alpha}_1 = \frac{R'_E \dot{r}_1}{r_1^2 \sin\alpha_1}$$

$$\dot{\alpha}_2 = \frac{R'_E \dot{r}_2}{r_2^2 \sin\alpha_2}$$

$$\dot{\phi} = \frac{-(\dot{\bar{r}}_1 \cdot \bar{r}_2 + \bar{r}_1 \cdot \dot{\bar{r}}_2) r_1 r_2 + (\dot{r}_1 r_2 + r_1 \dot{r}_2)\bar{r}_1 \cdot \bar{r}_2}{r_1^2 r_2^2 \sin\phi}$$

The angle ϕ is the angle between the two position vectors (see Fig. 1). If ϕ is less than $\alpha_1 + \alpha_2$, the objects can "see" each other. Therefore, if $v > 0$, the objects are in-view; $v = 0$ gives the rise and set times of the in-view periods.

Received Dec. 16, 1985; presented as Paper 86-0086 at the AIAA 24th Aerospace Sciences Meeting, Reno, NV, Jan. 6-9, 1986; revision received May 30, 1986. This paper is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

*Project Astrodynamicist. Member AIAA.

Notice that v and $\dot{v}$ are functions of R_E, h_g, the position vectors, their magnitudes, and the time derivatives of the latter two. Therefore, the algorithm could be potentially used for any vehicle or object that can be described by position vectors. In the present application, a mixture of low eccentricity satellites ($e \leq 0.1$) and ground stations are tested for all possible combinations of in-view periods.

A typical plot of v for two low eccentricity orbits is found in Fig. 2, from which its sinusoidal nature is evident. The reason for limiting the satellites to low-eccentricity orbits can be seen in Fig. 3, which shows v for one $e = 0.7$ satellite and one nearly circular satellite; here, v has lost its sinusoidal nature, and finding its zeros becomes difficult and time consuming. For the low-eccentricity case, P, the period of v can be found through the Fast Fourier Transform and time can be stepped through in fractions of P to bracket the time of maximum v, t_{max}. Then, t_{max} is found by the regula-falsi iteration method. If $v(t_{max})$ is negative, this maximum does not indicate an in-view period. If $v(t_{max})$ is positive, time is incremented backwards in fractions of P to bracket and iterate to the rise time; similarly, time is incremented forward from t_{max} to find the set time. From the set time, a new search is initiated for the next in-view period. This process is continued until the end of the time span of interest is attained.

The Algorithm

The following is a step-by-step description of the algorithm. Figure 4 contains a structure flow chart of the description, and subsequent sections describe in greater detail the major parts of the algorithm. Note that the $f_1, f_2, \ldots f_6$ values are the fraction factors of P, which will be detailed in a later section.

1) Start processing at the beginning of the time span of interest.

2) Use the Fast Fourier Transform to compute the period of v every 32,000 s, as described n the next section.

3) Find the region where $\dot{v} > 0$:

a) If $\dot{v} > 0$, the region is found.

b) If $\dot{v} \leq 0$ and $v < 0$, then $t = t + f_1 \cdot P$.

c) If $\dot{v} \leq 0$ and $v > 0$, then $t = t - f_2 \cdot P$.

d) Repeat this process until $\dot{v}(t_P) > 0$, where t_P is a time in this region.

4) Bound the time of maximum v, t_{max}:

a) Start at time of step 3, $t_L = t_P$.

b) Step in time, $t_R = t_L + f_3 \cdot P$.

c) If $\dot{v}(t_R) > 0$, then set $t_L = t_R$, and repeat substep b.

d) Otherwise, if $\dot{v}(t_R) \leq 0$, then t_L and t_R bracket the maximum.

5) Iterate to t_{max} with the regula-falsi method.

6) If $v(t_{max}) \leq 0$, skip down to step 11.

7) Bound the rise time:

a) Start at $t_R = t_{max}$.

b) Step in time, $t_L - f_4 \cdot P$.

c) If sign $[\dot{v}(t_R)] \neq$ sign$[\dot{v}(t_L)]$, then iterate toward t_{min} with the regula-falsi method; then, if $v(t_{min}) < 0$, $t_L = t_{min}$.

d) If $v(t_L) \geq 0$, then set $t_R = t_L$, and repeat substep b.

e) Otherwise, if $v(t_L) \leq 0$, then t_L and t_R bracket the rise time.

8) Iterate to the rise time (t_{rise}) with the regula-falsi method.

9) Bound the set time:

a) Start at $t_L = t_{max}$.

b) $t_R = t_L + f_4 \cdot P$.

c) If sign$[\dot{v}(t_L)] \neq$ sign$[\dot{v}(t_R)]$, then iterate toward t_{min} with the regula-falsi method; then, if v $(t_{min}) < 0$, $t_R = t_{min}$.

d) If $v(t_R) > 0$, then set $t_L = t_R$, and repeat substep b.

e) Otherwise, if $v(t_2) < 0$, then t_L and t_R bracket the set time.

10) Iterate to the set time (t_{set}) with the regula-falsi method.

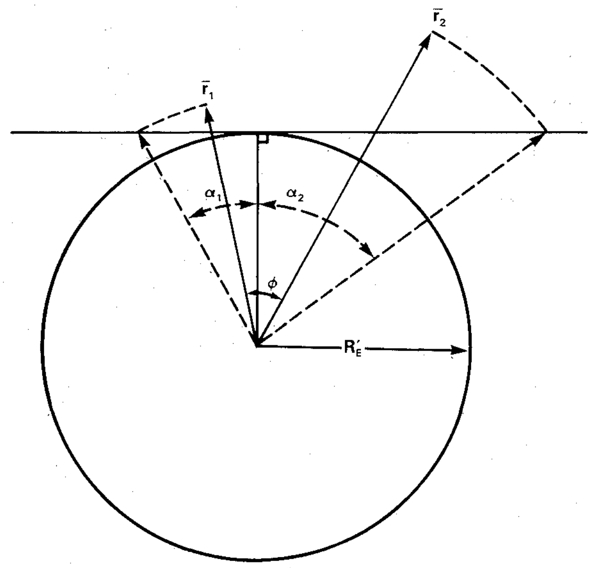

Fig. 1 Orientation of the angles describing v.

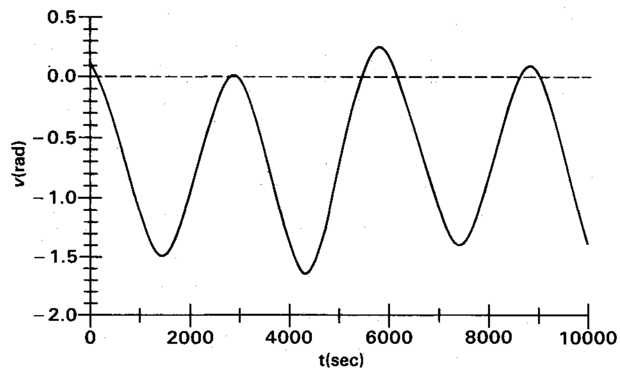

Fig. 2 A typical visibility function for two low Earth satellites with $e \leq 0.1$.

11) If t_{set} is less than the end of the time span of interest, set $t = t_{set} + f_5 \cdot P$, and start again at step 3 to find the next in-view period. (For the case where t_{max} produced no in-view period $[v(t_{max}) \leq 0]$, then $t = t_{max} + f_6 \cdot P$.)

The Fast Fourier Transform

The following procedure describes the estimate of P, the period of v, as found through the Fast Fourier Transform.

1) Find $v(t_k)$, $k = 1, 2, \ldots n$

2) Compute

$$C_j = \sum_{K+1}^{n} v(t_K) e^{-ik\omega}, \qquad j = 1, 2, \ldots n$$

$$= \sum_{K=1}^{n} v(t_K) e^{-ikj\Delta\omega}, \qquad j = 1, 2, \ldots n$$

$$= \sum_{K=1}^{n} v(t_K) e^{-(2\pi ijk/n)}, \qquad j = 1, 2, \ldots n$$

The time span is scaled to $(0, 2\pi)$. Note that $i = \sqrt{-1}$.

3) Find $j = j_{max}$, where C_{jmax} is the C_j with the largest modulus.

4) If two major C_j peaks occur, pick the one with the shorter period. This is done through the empirical relation that if $|C_j| > 0.25\, |C_{jmax}|$ for $_{j > jmax}$, then this value of j is chosen and becomes the new $_{jmax}$.

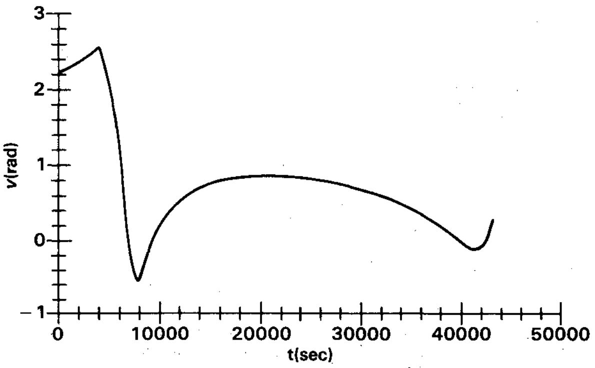

Fig. 3 Visibility function for two large semimajor axis orbits, one with $e = 0.7$.

5) $$\omega_{j_{max}} = \frac{2\pi(j_{max} - 1)}{n}$$

6) $$P = \frac{2\pi}{\omega_{j_{max}}}$$

The time span used to find P must be at least as large as P, and n must be large enough to obtain a good sample of the frequency of the function. Furthermore, n should be a power of 2 for the Fast Fourier Transform to yield the fastest results. The procedure gave good results in the cases tested with a time span for sampling v of 32,000 s and $n = 64$.

Step number 4 is necessary because v sometimes displays both short- and long-period nature. Figure 5 depicts v for a case like this; Fig. 6 depicts the associated $|C_j|$ values from the Fast Fourier Transform. In Fig. 5, one long period and

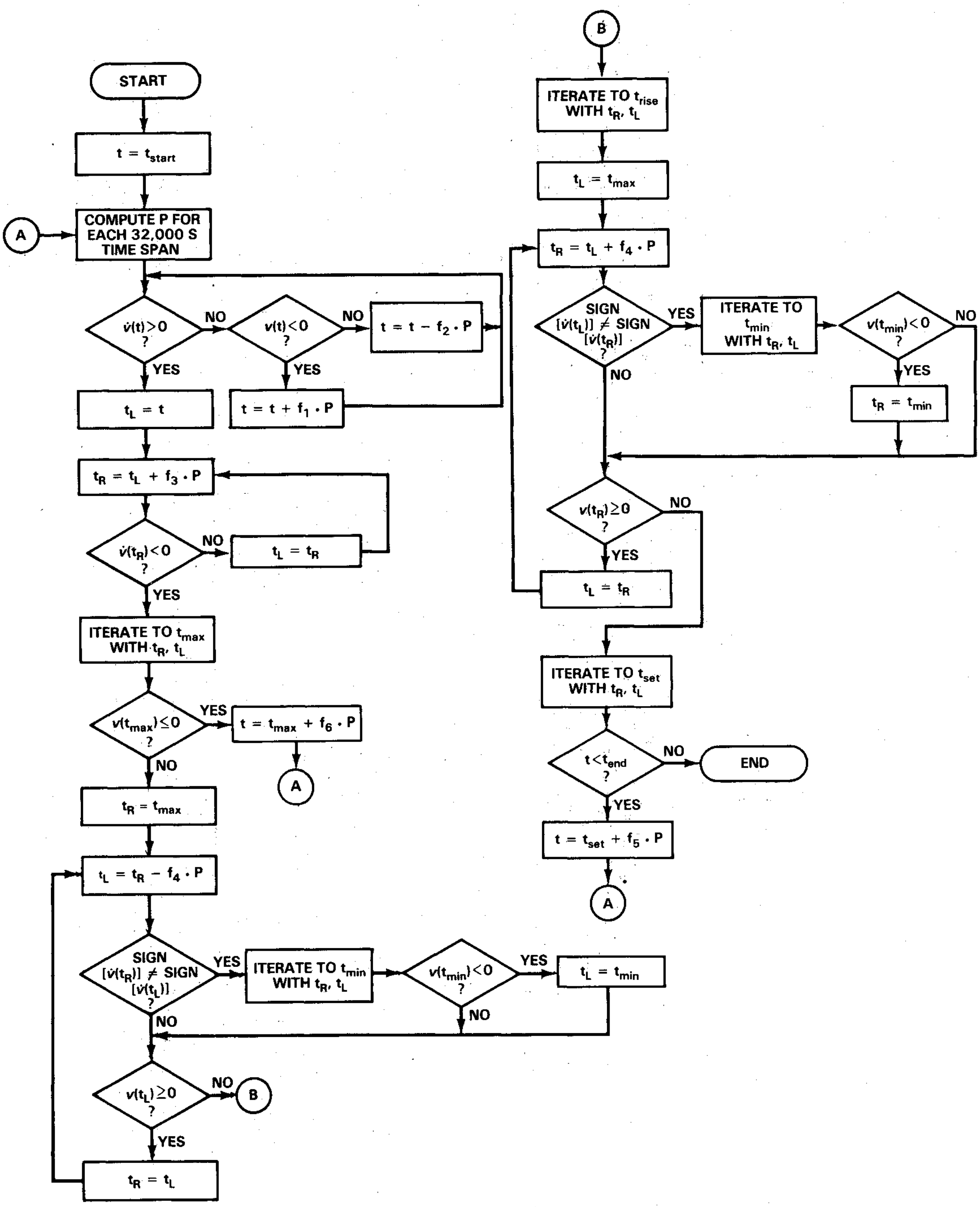

Fig. 4 Flowchart of the in-view algorithm.

Table 1 Orbital elements of satellites in tesseral gravity comparison

Satellite	a, km	e	i(°)	Ω(°)	ω(°)	M_0(°)
1	8500	0.1	80	45	135	0
2	7500	0.07	30	0	44.9	0

about two short periods are shown. The short period is desired because it is the actual period of v, while the long period is the variation of the center of the sinusoid in time.

Fractions of P as Time Increments

The techniques for the manipulation of the visibility function are to first find P, and then to step through the function in increments of time that are fractions of P. The key to this latter part is to find the proper fraction set, $f_1, f_2, .. f_6$, which will not miss any maximum, rise, or set times. The simplest means of picking the fractions consists of considering an ideal sinusoidal function and breaking it up into quarters of the period, as was done by Castro,[2] and is shown in Fig. 7. The first fraction in the preceding algorithm, f_1, must equal at most 0.5 because the starting time could be in region 1 or region 2 (as defined in Fig. 7), with $v<0$. If $v>0$ at the starting time, then time must be decremented by $f_2 \cdot P$ until $v(t)>0$. In order to not overshoot t_{max} by a large number and having it iterate back to t_{max}, $f_2=0.25$ appears to be a good choice. To find a region where $v(t)<0$, choose $f_3=0.25$; this should bound t_{max} without large overshoot. Similar reasoning for the rest of the fractions yields $f_4=0.125$, $f_5=0.5$ and $f_6=0.75$. Since the sinusoidal nature of v is not ideal (that is, the amplitude and period are not constant), these fractions should be trimmed to minimize overshooting desired points. The new empirical fractions that yield good results over many test cases are: $f_1=0.2$; $f_2=0.1$; $f_3=0.1$; $f_4=0.1$; $f_5=0.2$; and $f_6=0.3$. For the cases investigated, the results with thee fractions were very good, locating 99.99% of the in-view period and yielding a savings in computer time of 95% over a brute force method.

Comments on Methods of Zero Finding

Attempts were made to find the maximum v, rise, and set times by approaching the values through approximation methods and converging on the exact values with a Newton-Raphson iteration scheme. This approach, although fast, was too unstable to find a reasonable precentage of the in-view periods. For this reason, methods that bracket the zero of a function were used to guarantee convergence. Brent's's[3] method, a very successful and usable algorithm, was tried, but it was found that the regula-falsi method actually took slightly less computer time. The probable reason for this is that Brent's method does many numerical checks in order to guarantee convergence for a wide range of poorly behaved functions, while the regula-falsi method converges quickly for the relatively nicely behaved visibility function.

Earth Oblateness Gravity Perturbations

Earth oblateness effects (J_2 gravity pertubations) were added to the satellite trajectory models to yield more realistic in-view period information for actual implementation. The secular effects of the disturbing force on Ω, ω, and M were implemented through the following equations:

$$\langle\dot{\Omega}\rangle = \sqrt{\frac{\mu}{a_0^3}}\left\{-\frac{3}{2}J_2\left(\frac{R_E}{P_0}\right)^2 \cos i_0\right\} \tag{3}$$

$$\langle\dot{\omega}\rangle = \sqrt{\frac{\mu}{a_0^3}}\left\{\frac{3}{4}J_2\left(\frac{R_E}{P_0}\right)^2(-1+5\cos^2 i_0)\right\} \tag{4}$$

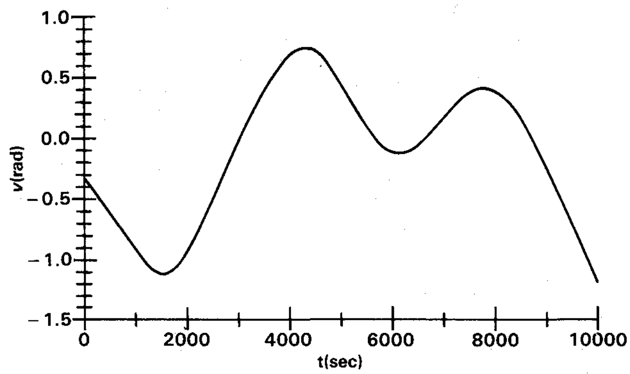

Fig. 5 Typical visibility function showing short-and long-periodic nature of some cases.

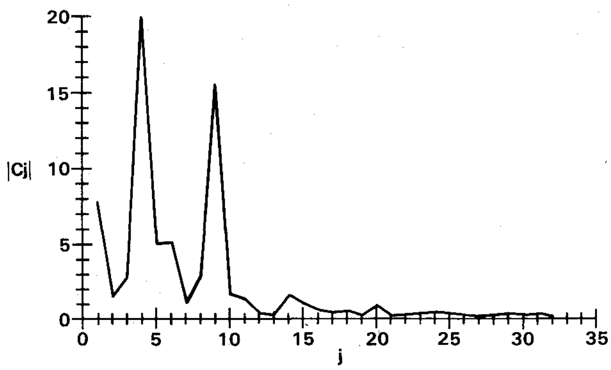

Fig. 6 Results of the fast Fourier transform for the case of Fig. 5.

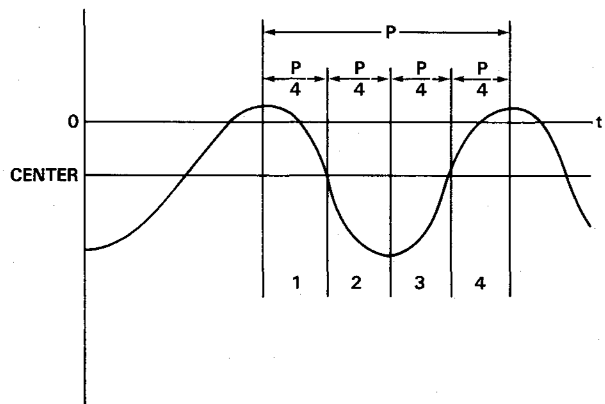

Fig. 7 Fractions of P, the period of v.

$$\langle\dot{M}\rangle = \sqrt{\frac{\mu}{a_0^3}}\left\{\frac{3}{4}J_2\left(\frac{R_E}{P_0}\right)^2\sqrt{1-e_0^2}(-1+3\cos^2 i_0)\right\} \tag{5}$$

The subscript (°) denotes mean orbital elements.

The algorithm starts with the mean elements of the orbit and models the effect of J_2 perturbations with these equations. These additions to the algorithm, of course, added more computer run time (an increase of approximately 7.5%);

nevertheless, the algorithm yielded consistent results when compared to a brute force method containing the same secular perturbation equation.

The in-view algorithm with J_2 gravity perturbations was also compared to a brute force model that obtains its position and velocity vectors from ephemerides generated by a 14 by 14 tesseral gravity computer program. The purpose of this test was to ensure that the algorithm can be used with real satellite data to accurately predict in-view periods. The comparison was done with two satellites over a time span of one day. The orbital elements for the satellites are shown in Table 1. The standard deviation of the difference between the J_2 in-view algorithm's and the tesseral brute force program's rise and set times was 4.1 s. Therefore, the data produced by the J_2 gravity pertrubations version of the in-view algorithm could potentially be used in an operational environment.

Numerical Tests

For the spherical-Earth case, the major numerical tests that were done consisted of 4,046 satellite-satellite comparisons (119 satellites in one set compared to 34 in the other set) and 68 satellite-ground station comparisons (2 ground stations compared to 34 satellites), over a time period of one day. The results of the algorithm were weighed against the same data run in a brute force algorithm, and had a vector-check time step of 5 s. The satellites used had eccentricities ranging from 0.01 to 0.1, and altitudes ranging from low-Earth orbits to geosynchronous orbits. Out of the 35,915 in-view periods located by the brute force method, the present algorithm found 35,914 of them (99.997%), with a savings in computer time of 95%.

The same numerical data was used for the oblate-Earth gravity field case as was used for the spherical-Earth gravity field. Out of the 36,133 in-view periods located by the comparable brute force method, the algorithm found 36,130 of them (99.992%), with a savings in computer time of 95%.

Conclusions

The in-view algorithm previously described has been developed for low eccentricity satellites to yield accurate in-view information for satellite-satellite and satellite-ground station comparisons over time spans of interest; this is much faster than an accurate brute force method. Also, the algorithm is not limited to range constraint comparisons. The algorithm was tested with both spherical-Earth and oblate-Earth gravity fields, yielding a savings of 95% in computer time over a comparable brute force method, and finding 99.9% of the in-view periods. Furthermore, the oblate-Earth version yelded accurate results when compared to a tesseral gravity brute force method, showing its utility for applications in an operational environment.

Since the visibility function is only a function of Earth physical constants, position vectors and magnitudes, as well as the first-time derivatives of these quantities, the in-view algorithm could easily be extended to include other vehicles with well-behaved position time histories. Potential candidates include nonaccelerating aircraft, ships, and ground vehicles. Furthermore, for work with satellites, the algorithm is not limited ot the secular J_2 perturbation method described above; almost any method of obtaining accurate vectors could be used. For example, for each satellite, position and velocity vectors could be read from a precalculated tesseral gravity ephemeris instead of being computed internally. Therefore, the in-view algorithm described herein has been proved successful for the applications decribed in the body of the paper, and also has the potential for success in other diverse applications.

References

[1]Parks, A.D., "A Mathematical Description of the Station Alert Predictor," NSWC TR 82-387, Naval Surface Weapons Center, Dahlgren, VA, Sept. 1982.

[2]Castro, H.E., "A New Method for Finding Station Prediction Alerts," NSWC/DL TM K-1/63, Naval Surface Wapons Center, Dahlgren, VA, Aug. 1963.

[3]Brent, Richard P., *Algorithms for Minimization Without Derivatives,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, pp. 47-60.