



Real-World Benchmarking

By Dr. T.S. Kelso



November/December 1996

In our previous column on benchmarking (March/April 1996, pp. 80-81), we discussed three issues of particular significance to benchmarking: relevancy, speed, and accuracy. While most users are well aware of the need for benchmarks to report both speed and accuracy, it is also extremely important that the benchmark used be relevant to the type(s) of applications the system supports.

For satellite tracking applications, we showed that fairly simple system benchmarks, such as the Savage benchmark, can give a pretty good idea of the relative performance—both in speed and accuracy—of the system hardware, operating system, and application software (most notably the compiler). A more thorough demonstration of application performance was shown using an SGP4-based benchmark (that is, a benchmark based upon the NORAD SGP4 orbital model) which calculates the positions of a set of ten satellites at one-minute intervals throughout an entire day. The satellites were chosen to be representative of a wide range of orbit types. The time taken to calculate the 14,400 predictions is used to determine the average throughput of the particular system (hardware, OS, and compiler) together with the specific implementation of the orbital model. An even better implementation would not only calculate the throughput but also calculate the average error relative to an expected result.

While this previous discussion of benchmarking served to give us a good idea of system and application performance, it is all a bit antiseptic. That is, we really haven't done anything to show how well a particular application tracks a satellite in the real world. Just as it is important to understand the relative performance factors of various systems, it is often critical to show that the system and application software work together to produce the desired result.

Real-World Data

Before we can set up any real-world benchmarks, there are several things that must be done. First, we must understand what type(s) of data are representative of our application under normal operating conditions. For example, if we need to point an antenna at a satellite for communication, to collect telemetry, or for radar tracking, then we will want to use look angles (azimuth and elevation) to test our application's performance. If we need to point a telescope at a satellite to perform optical tracking, then we would probably want to use right ascension and declination so that the satellite's position could be measured directly against a star background.

The second step in the process is to obtain a good quality data set to test the application against. The better the quality of the data set, the more useful it will be for this type of benchmarking. Because any data set is merely a set of observations—and any such set comes complete with its own errors—it is important that not only the raw data be available, but that there also be some discussion of the observation parameters and the error characteristics.

Good quality data sets can be difficult to come by, but if you know where to look, they can be found. For example, several satellite programs provide high-precision ephemerides (or more simply, tables of position and velocity over time). Satellite navigation programs, such as the Global Positioning System (GPS), or geodetic satellite programs, such as the LAGEOS or TOPEX/poseidon missions, provide very-high-precision ephemerides which can be used to test a wide range of applications and can even be used to test performance between competing high-precision orbital models. Testing high-precision models, of course, requires complete understanding (and modeling) of the specifics of the coordinate systems involved. About the only real drawback to these types of data sets is that they are available for only a limited set of orbit types.

The final requirement to be able to conduct real-world benchmarking of an application is to be able to output data from the application in a format suitable for comparison with the real-world data sets. Many satellite tracking applications have the ability to output ephemerides for use in analysis, but many do not describe the particulars of the coordinate system being used. For those applications that are entirely graphically based, testing anything more than a few data points can be extremely frustrating and time consuming.

A Real-World Case Study

Let's examine an actual case study to see what kind of information is required and how such a benchmark might be applied. As an example, let's look at some visual observing data for two satellites—Landsat 5 and Cosmos 1766—and see how it was used to perform part of the validation effort for the routines that went into my SGP4 Pascal libraries that are used in the TrakStar program.

In the early 1990s, I was approached by the US Navy to help them test the operation of a new S-band antenna that was being built for them. Because of the narrow beamwidth of the antenna, it was important that the antenna be able to track within one-quarter degree of the intended target. Although the antenna was being built to support their FLTSATCOM satellites—which are geostationary and require little, if any, tracking—they wisely chose to test the system performance using a low-earth orbiting satellite (DMSP). The US Navy needed a satellite tracking program which would provide look angles to get the new antenna within the required tolerance.

As the code was developed and **verified** (**verification** is the process of ensuring that the program does what it is expected to do), it was obvious that the US Navy would not accept the program unless it could be **validated** (**validation** is the process of ensuring that the program accurately models the real world). That meant I needed to show that the program would work tracking representative low-earth orbiting satellites within the tolerances the US Navy required. I turned to a set of data I had obtained while working on another project with Contraves Goerz Corporation back in 1987 (Contraves is best known for building the optics and mounts for US Space Command's GEODSS deep-space surveillance network, a system capable of resolving a basketball-sized target at a range of 20,000 miles).

The data was obtained during the process of satellite tracking using an optical instrument called a KINETO (KineTheodolite). The KINETO instruments are used by test ranges from the Far East to White Sands, NM. This particular instrument mounts a pair of lenses, a Zoom lens of instrumentation accuracy (180-1800 mm), and a 100-inch catadioptric lens of Contraves design, with better than 150 line pairs resolution for 75mm film. The instrument is remotely controlled by a set of 11/73 series DEC mini-computers (remember, this was built in the mid-1980s), and uses a hybrid, 8 MHz, real-time video tracking unit developed for a military anti-aircraft unit. The electronics are all mounted in an air-conditioned van, and the whole setup is mobile. With this instrument, you can drive down the road, park, set up in about an hour, and track a target. After the track, you can reduce the data, and obtain output on the target's look angles or position in ECI coordinates—again in less than an hour.

The data was collected in an auto-tracking mode with samples taken every 0.02 seconds. The azimuth and elevation data is accurate to within 5 arc seconds (approximately 0.001 degrees), after having been corrected by a star calibration run. The observation point is known to approximately 1 cm, having been surveyed by GPS, Phase II, and by a USGS benchmark some 1 mile from the site. The details are presented in table 1.

Table 1. Observation Site Coordinates

Geodetic Coordinates		XYZ Offset	
Latitude	40.5018 degrees	X coordinate	-7.8500 m
Longitude	280.1250 degrees	Y coordinate	-17.6940 m
Altitude	280.2710 meters	Z coordinate	2.2910 m

The latitude, longitude, and altitude of the observation site represents the origin of the XYZ coordinate system with the offset denoting the center of the observing system (center of axes of an El-over-Az mount).

An azimuth of zero degrees is true north and an elevation of zero degrees corresponds to the true earth tangent (WGS 72). The targets were tracked with a 0.258-degree field of view optic using a video camera and a digitizing TV tracker. The field of view is 412 pixels wide (x) by 312 pixels high (y)—that is, each pixel represents an angle of less than one millidegree (0.001 degree).

Table 2. Validation Data Sets

Landsat 5 Observations
Cosmos 1766 Observations
TrakStar Observer File
Validation Element Sets

The observations were collected as a set of UTC times with corresponding (corrected) azimuth and elevation. To complete the data collection, it was necessary to obtain NORAD two-line element sets just prior to the observation periods. Both the original observation data and the NORAD two-line element sets are available on the CelesTrak WWW (<http://celestrak.com>). The results of the observations are plotted against the predictions from [TrakStar's](#) look angle output in figures 1 and 2.

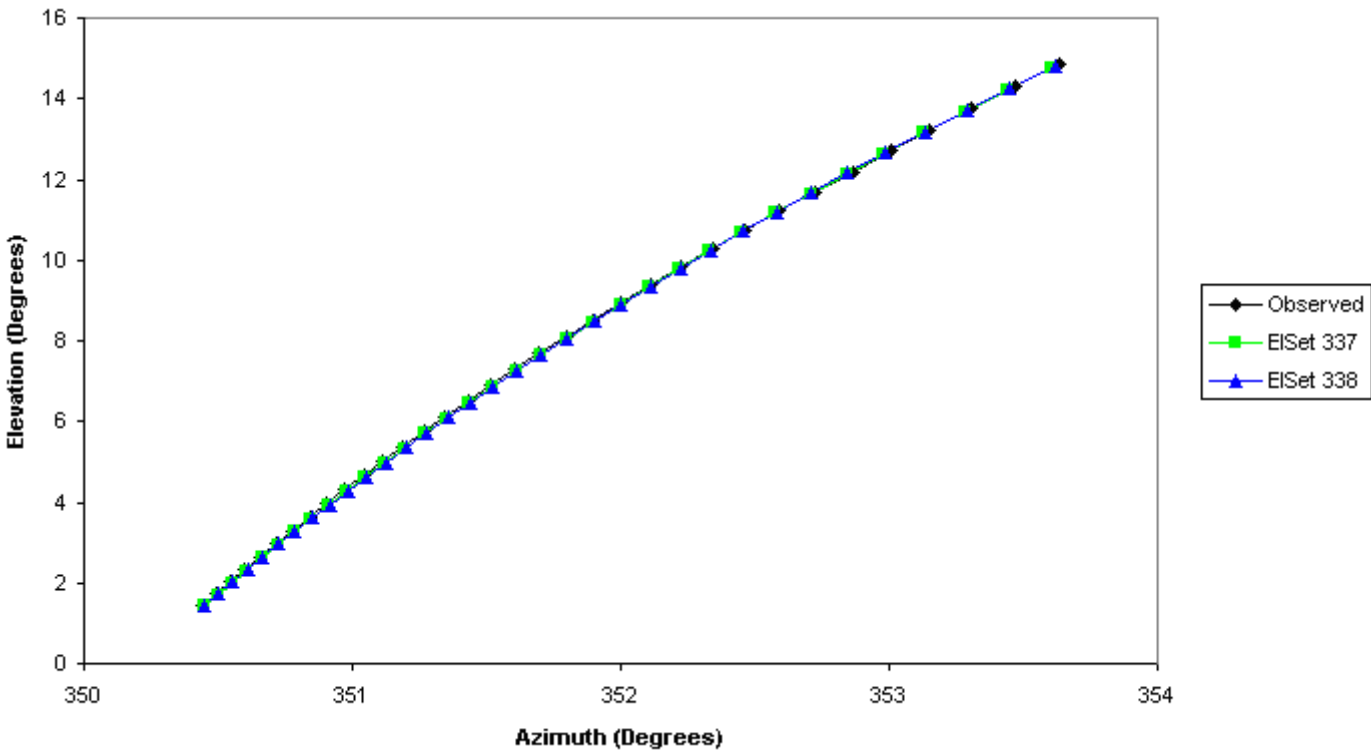


Figure 1. Landsat 5 Look Angles

In figure 1, the observations for Landsat 5 are plotted at five-second intervals along with the predictions from two successive NORAD two-line element sets—one with an epoch almost two days prior to the observations and the other with an epoch three hours afterwards. Landsat 5 is moving toward the horizon during the period of observation.

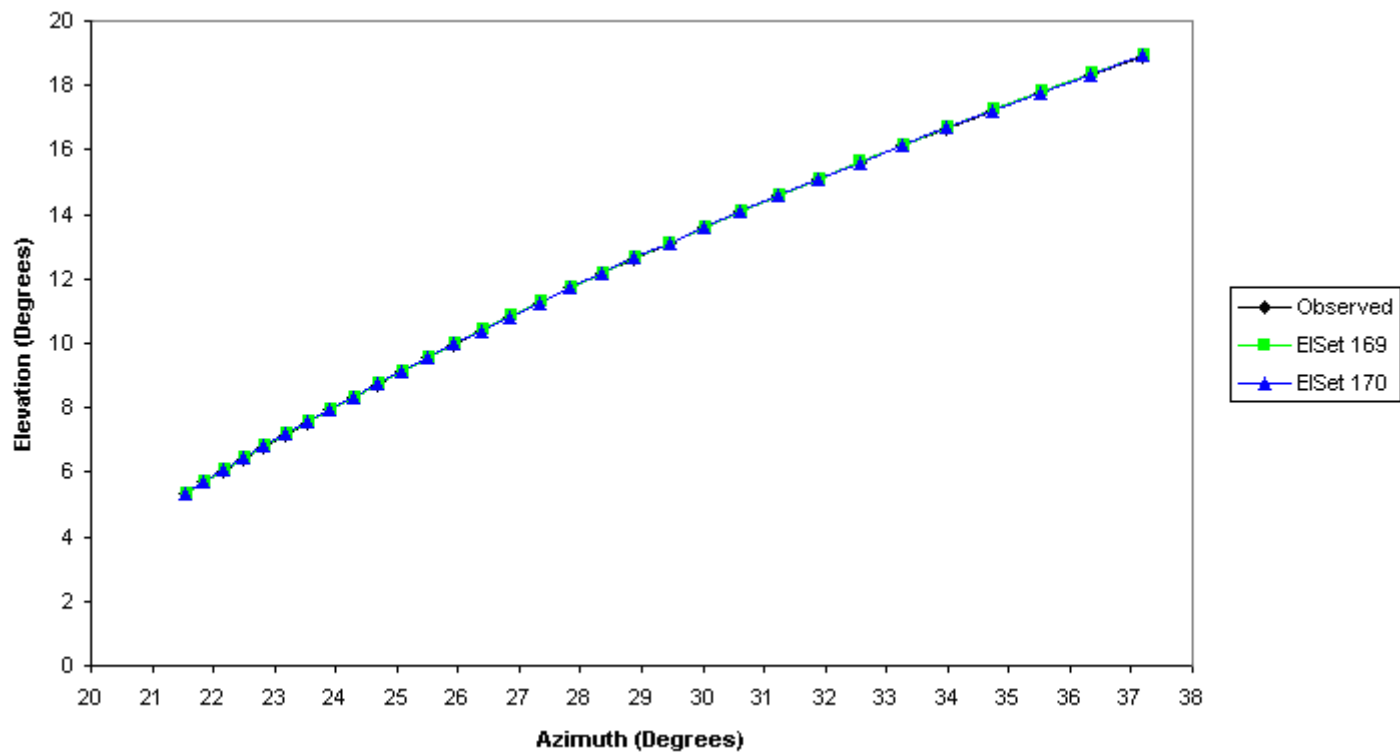


Figure 2. Cosmos 1766 Look Angles

In figure 2, the observations for Cosmos 1766 are plotted at five-second intervals along with the predictions from two successive NORAD two-line element sets—one with an epoch almost two days prior to the observations and the other with an epoch about an hour and a half afterwards. Cosmos 1766 is moving toward the horizon during the period of observation.

Examination of figures 1 and 2 shows very good agreement between the observations and the values predicted by TrakStar using either two-line element set. In fact, the agreement is so good that it is difficult to distinguish any difference on this scale. This is true even at the lower elevations where atmospheric refraction becomes significant because TrakStar calculates this effect (using standard atmosphere) when determining the look angles. In order to accentuate the differences between the observations and the predictions, the error (that is, the observed value minus the predicted value) is plotted for each element set in figures 3 and 4.

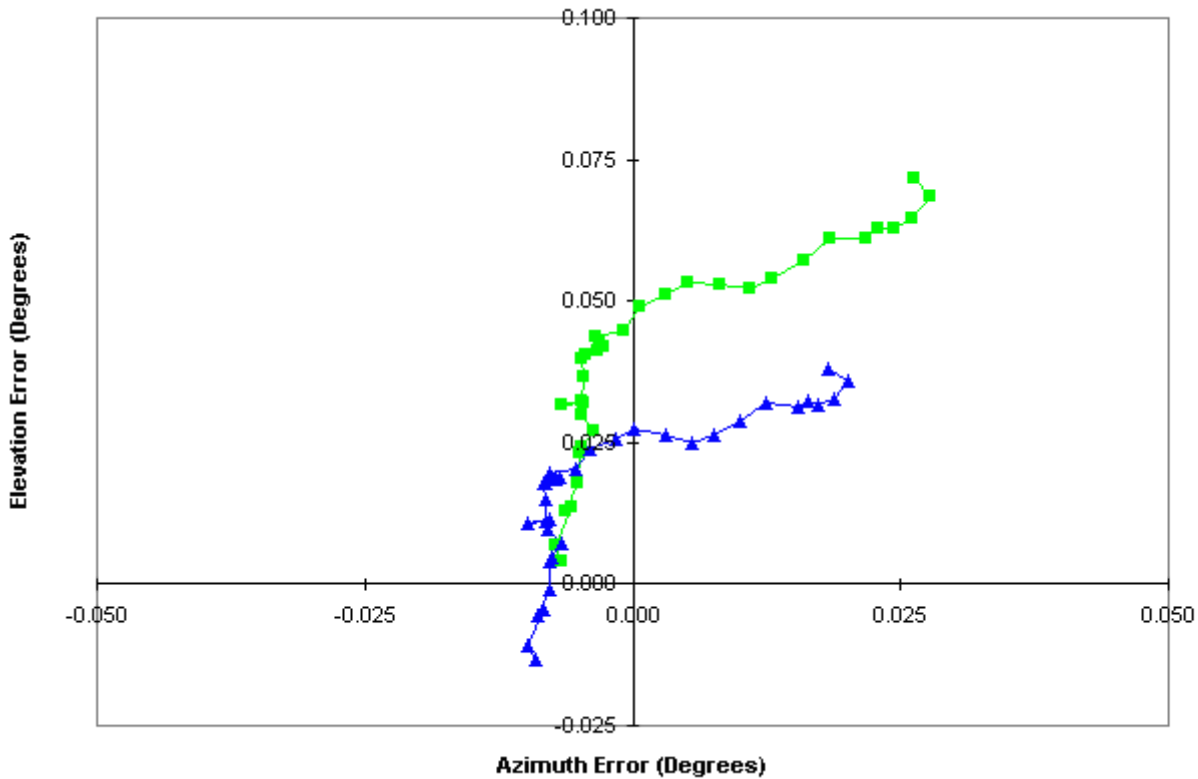


Figure 3. Landsat 5 Errors (Observed Minus Predicted)

In figure 3, the error trend for Landsat 5 moves from the upper right to the lower left (toward the origin) over the course of the observation period. While there does appear to be some systematic trend in the error, it is quite clear from this graph that the error remains well within the quarter-degree tolerance.

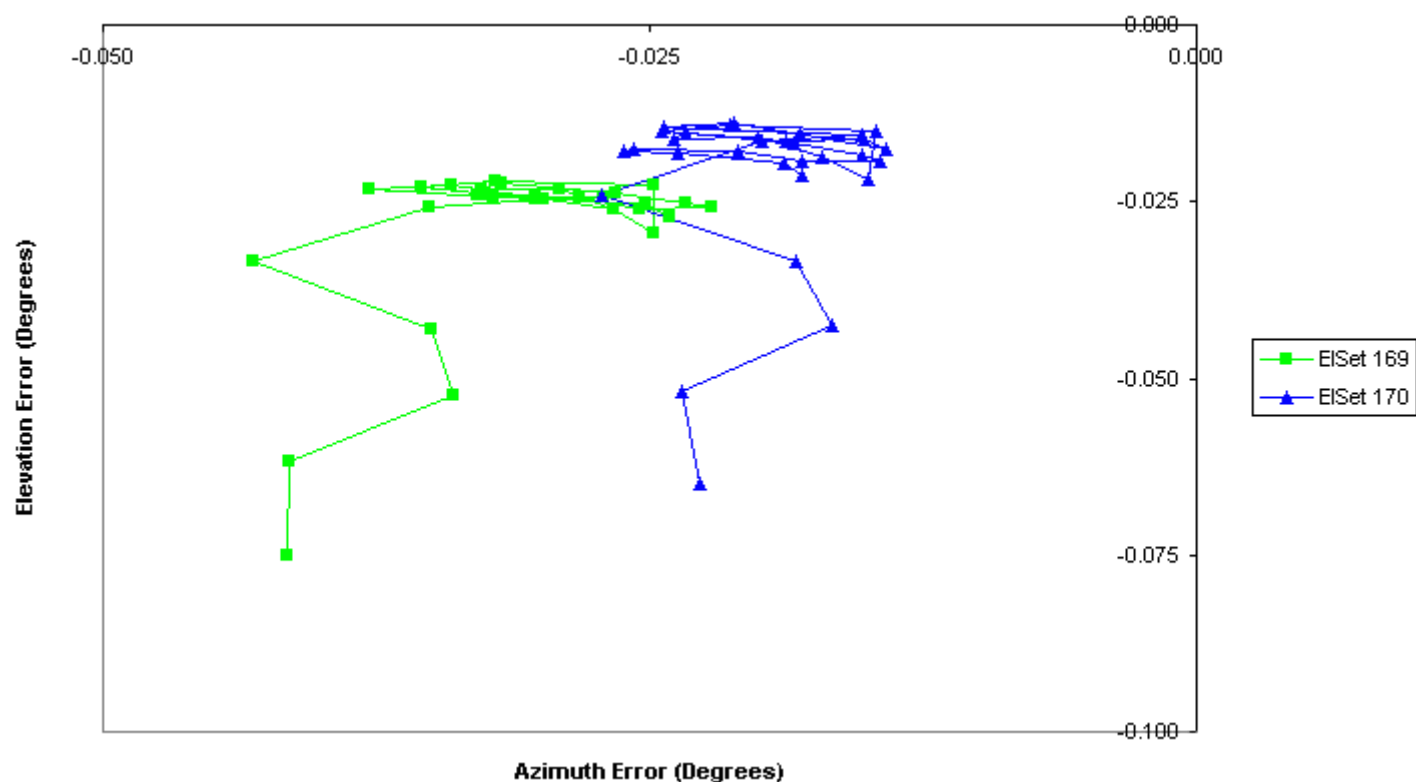


Figure 4. Cosmos 1766 Errors (Observed Minus Predicted)

In figure 4, the error trend for Cosmos 1766 moves from the lower left to the upper right (toward the origin) over the course of the observation period. Again, while there does appear to be some systematic trend(s) in the error, it is evident that the error remains well within the quarter-degree tolerance required by the US Navy. It was just this kind of analysis, using other test cases, that convinced the Navy that the NORAD SGP4 orbital model, together with the NORAD two-line element sets, was capable of generating look angles of sufficient accuracy to allow them to acquire low-earth orbiting satellites using their new S-band antenna.

Other Test Cases

There are many ways to devise useful benchmarks. We have already discussed using state vectors from satellite navigation programs such as GPS or various geodetic satellites to test model accuracy and demonstrated how visual observations can be used in our case study. Other approaches could include auto-tracking using a radio signal (many S-band antenna systems can do this), using Doppler shifts, or even timing AOS and LOS (acquisition of signal and loss of signal). These methods may not be as accurate as the others discussed, but could provide adequate validation for certain types of applications. In a future column, we will show how visual imagery can be used for validation, using a case study analyzing APT (automatic picture transmission) data from the NOAA polar-orbiting weather satellites.

A Proposal

While it is important for any user to understand the limitations of the application they are using for satellite tracking, it can be difficult and time consuming to build an appropriate test suite and conduct the benchmarks for that application themselves. A much more reasonable approach would be for the producer of the application to provide—as part of the application's documentation—an analysis of the application's performance against its intended purpose.

To ensure that such an analysis fairly represents a particular application's performance and that the results are directly comparable to those from other similar applications, it is paramount that such an analysis be conducted against a standardized test suite. Such a test suite could be developed by a technical committee of one of the professional societies of the astrodynamics community, such as the American Institute of Aeronautics and Astronautics (AIAA) or the American Astronautical Society (AAS). Once a draft test suite was developed, it would be made available for community comments—with those comments being incorporated into the final version, as appropriate.

The test suite would consist of a number of dated data sets, together with relevant details on observation conditions, coordinate systems, and how the test should be applied. The test suite should be easily accessible via the Internet. This type of standard is long overdue in our community and would be a welcome addition not only in comparing satellite tracking packages but in improving their overall performance.

If you have any questions or comments regarding this column, please feel free to contact me at TS.Kelso@celestrak.com. Until next time, keep looking up!



[Dr. T.S. Kelso \[TS.Kelso@celestrak.com\]](mailto:TS.Kelso@celestrak.com)
Follow CelesTrak on Twitter @TSKelso
Last updated: 2019 Dec 28 02:28:00 UTC
Accessed 84,711 times
Current system time: 2020 Jun 26 13:00:03 UTC
[Celestrak's Simple Privacy Policy](#)

