GEOTracker

Tracking Antenna Control Program

Program Description
and
Operations Manual

January 17, 2003

NASA Glenn Research Center
Cleveland, Ohio

Developer: Russ Jirberg

LAKE LOGIC SYSTEMS

2830 Bonny Blvd., Parma, OH 44134
Phone / Fax: 440.884.1019 E-mail: Russ@Jirberg.com

TABLE OF CONTENTS

FIGURES

TABLES

ACKNOWLEDGEMENTS

The *GEOTracker* program was developed for the NASA Glenn Research Center, Cleveland, Ohio under Contract No. C-71662-T. Its development was accomplished in coordination with team members of the NASA Ka-band Propagation Studies team at Glenn. Significant technical contributions were made by team members Dr. Roberto J. Acosta (Team Leader), Sandra K. Johnson, Walber Feliciano, David R. Kifer and Peter Harbath. Special thanks go to Sandra K. Johnson for her development of the tracking files and for her project administration.

GEOTracker

Tracking Antenna Control Program

1.0 OVERVIEW

GEOTracker is a custom designed program that facilitates the collection of data for Ka-band propagation studies. It does so by automating the task of tracking geosynchronous (GEO) satellites which undergo orbital variations that result from a lack of adequate station-keeping. It provides an open-loop mode of control wherein the antenna pointing information is obtained from user-generated tracking files. The data in these files specify the azimuth and elevation coordinates of the satellite relative to the tracking antenna. These coordinates are calculated based on the ephemeris data for the satellite and on the location of the tracking antenna.

GEOTracker was designed specifically for use with the Research Concepts, Inc. model RC2000 antenna positioner. *GEOTracker,* which is PC-based, interfaces to the positioner through the RC2000 Control Unit. The RC2000 positioner has two motorized lead screw actuators that adjust the azimuth and elevation angles of the antenna. *GEOTracker* converts the azimuth and elevation coordinates obtained from the tracking files (in degrees) to a corresponding pair of rotational coordinates (i.e. turns or *counts*) that govern the position of the two actuators. *GEOTracker* provides a user interface screen that enables the user to control its auto-track and manual modes of operation as well as its built-in system calibration capabilities.

2.0 INSTALLATION

The major components of the NASA Ka-band GEO Propagation Terminal are shown in Figure 1. The outdoor equipment is comprised of a 1.2m reflector antenna, the Research Concepts dual-axis antenna positioner, a Global Positioning Satellite (GPS) receiver, and a Ka-band radiometer / beacon receiver. The antenna positioner is connected to the model RC2000 Antenna Controller which is located indoors along with the Control Computer. The GPS receiver, which provides the system with an accurate long-term time base, is also connected to the Control Computer. A second computer is used to process the radiometry and beacon signals. The use of a second computer is an expedient that allows the propagation terminal to continue using the long-established DOS-based radiometry / beacon program entitled DACS.exe.

The Control Computer should be a Pentium-class PC running Win98/Me/2K/XP. It must have a minimum of 128 KB of RAM, a minimum bus speed of 160 MHz, and a VGA monitor with 1024 by 768 pixel resolution. Additionally, it must have two comm ports: Com1 and Com2. Com1 connects the computer to the RC2000 Antenna

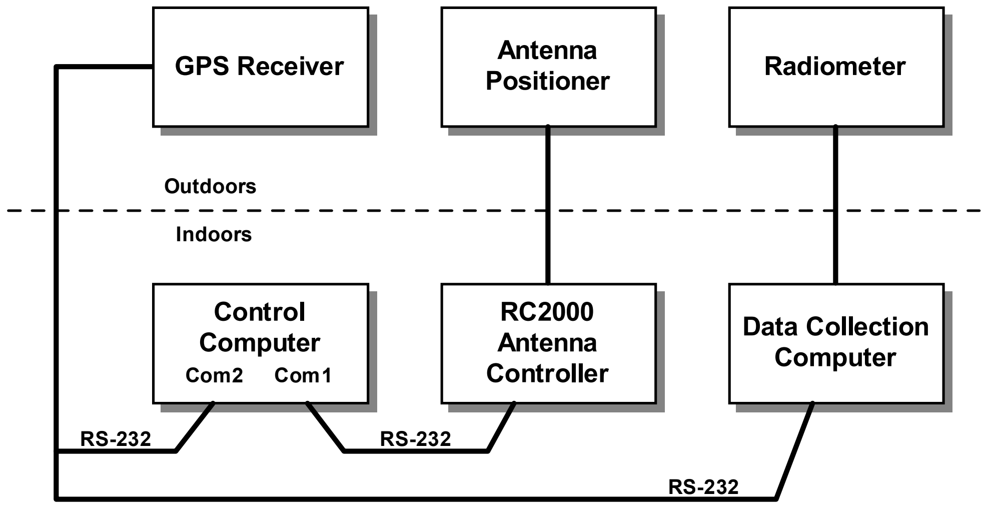

Figure 1. NASA Ka-band Propagation Terminal, block diagram.

Controller. Com2 connects the computer to the GPS receiver. A Zip drive is the prescribed means of loading the tracking files into the computer. Finally, the complement of software in the Control Computer must include LabVIEW™ version 6.1 and *NMEATime* version 5.2. *NMEATime* is a shareware program that is used here to synchronize the internal clock of the PC to the time base of the GPS network.

2.1 *GEOTracker* Program File Installation

GEOTracker is a LabVIEW application. It was developed to run under Win98, but is readily transportable to other Windows, Linux, and Mac operating systems. The program consists of 20 files as listed in Appendix A. *GEOTracker* is intended to be installed on the computer's C: drive in a folder named *C:\GEOTracker.* Simply create the folder, then copy the *GEOTracker* files from the program disk to that folder. Next, create a subfolder *C:\GEOTracker\TracFiles*. During operations the program will automatically copy the tracking files from the Zip drive to the *TracFiles* subfolder on the C: drive as explained in Section 5.3. Note that the default folders specified above may be changed if necessary so long as corresponding changes are made to the program code as described in Section 4.1.

To simplify the process of starting the program, create a Desktop shortcut to the *GTmain.vi* file. Then right-click on the icon, select the Properties option, and browse to change the icon to the *GT1.ico* file included among the *GEOTracker* files.

2.2 *NMEATime* Clock Synchronization Program Installation

A copy of the *NMEATime* program, version 5, is supplied on the *GEOTracker* program disk. *NMEATime* runs in the background periodically re-synchronizing the computer's internal clock to that of the GPS network. By default, *NMEATime* installs on the C: drive in the folder: *C:\Program Files\Vgps\NMEATime\NMEATime*.

After installing the program, configure it by right-clicking on the *NMEATime* icon and selecting "Properties." From the Properties popup window, configure *NMEATime* as indicated in Table 1. Next, again right-click on the *NMEATime* icon and select "show clock." Finally, right-click on the clock itself and select "Digital Clock." Then select "Clock settings" and check the following parameters:

- ✓ Always on top
- ✓ UTC time
- ✓ 24 hour time
- ✓ Show Day of Year

Complete the installation of the *NMEATime* program by connecting the GPS receiver to Com2 on the Control Computer. Insure that the transmit signal from the GPS receiver is connected to the receive pin (i.e. pin 2) of the DB9 connector on the computer. Verify that the GPS signal is being received by noting that the Rx indicator on the "GPS Configuration" tab of the *NMEATime* Properties popup window is flashing. Finally, set the time zone for the clock in the PC to GMT (alternatively referred to as UTC).

Tab	Selection / Setting
Time	n/a
GPS Signal Quality	Indicates GPS lock and operational status.
Update PC Clock	Set update interval to 15 minutes.
Position	n/a
GPS Configuration	Settings: Com Port 2, 4800 baud, NMEA GPRMC msg., delay = 0 ms., Uncheck GPS Quality, 0 satellites.
About *NMEATime*	Enter the registration number.
License	n/a
IRIG Configuration	n/a
Network Time	n/a
System Configuration	Select: "Set PC clock using GPS."

Table 1. *NMEATime* Configuration Parameters.

Note that the GPS receiver is also connected to the Data Collection computer. Further note that the data flow between the Control Computer and the GPS receiver is uni-directional (i.e. data is transferred only *from,* not *to,* the receiver), whereas the data flow to/from the Data Collection computer is bi-directional. Thus at the Control Computer, make **no** connection to the Tx output (i.e. pin 3) of the DB-9 connector. Connect only the Tx signal from the GPS receiver to the Rx pin (i.e. pin 2) of the DB-9 connector, plus the signal return line to pin 5.

At the Data Collection computer, connect the Tx and Rx signal lines from the GPS receiver to the Tx and Rx pins (i.e. pin 3 and pin 2, respectively), and connect the signal return line to pin 5. The transposition of the Tx and Rx signals is handled internally in the RC2000 Controller.

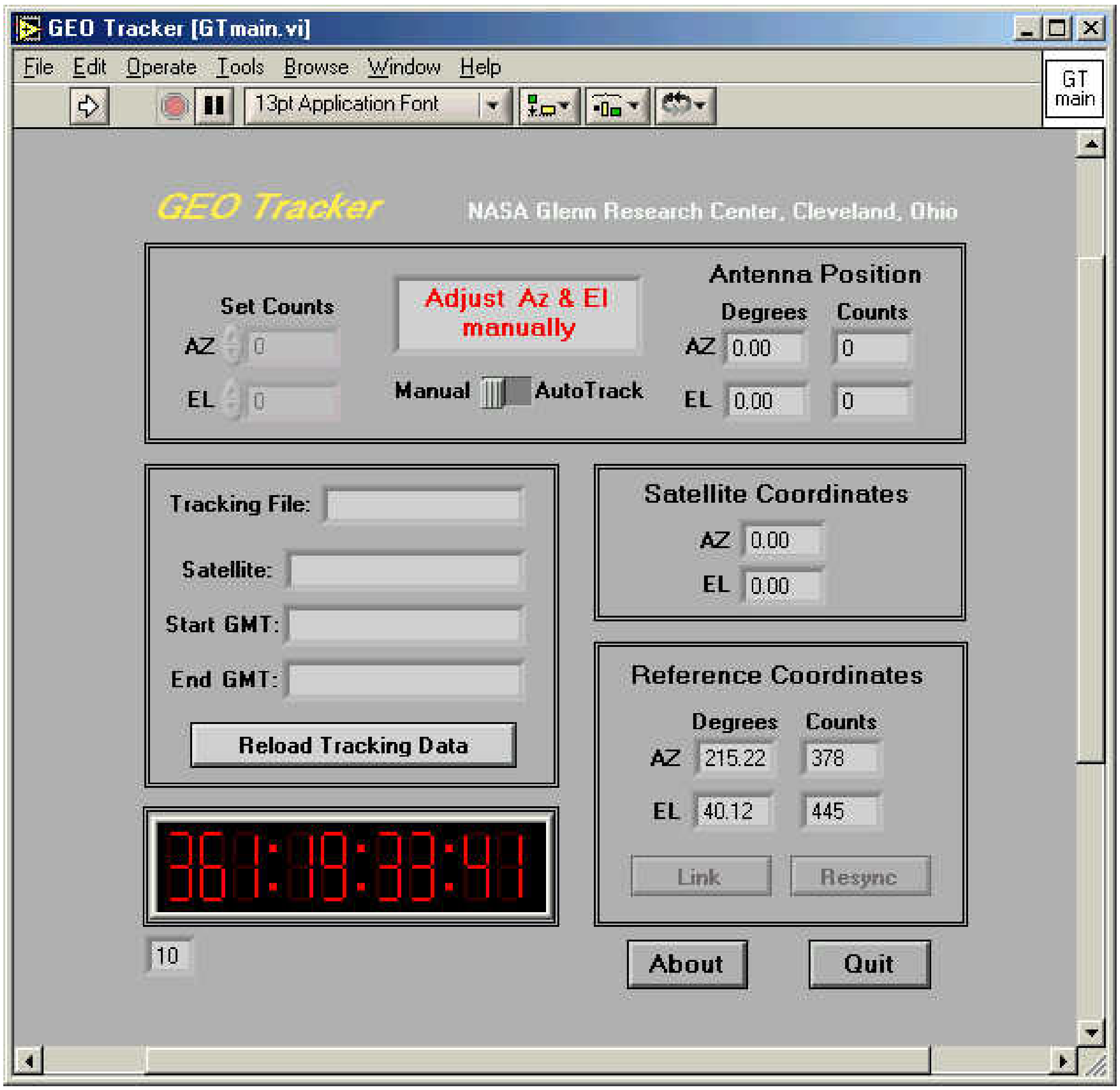

Figure 2. User Interface

3.0 User Interface

The User Interface screen is shown in Figure 2. It is the front panel of the *GTmain* vi. From this screen the operator can control and monitor all aspects of the program.

3.1 Functional Groups

The User Interface is comprised of five major functional groups of controls and indicators. The function of each of these groups is as follows:

1. The upper-most group includes a readout panel that indicates the operational status of the program and a mode selection switch that allows the operator to put the program in either the *Manual* mode or the *AutoTrack* mode. In the *Manual* mode the antenna is moved to the *count* coordinates entered by operator in the *Set Counts* controls located in the upper left-hand corner of the screen. In the *AutoTrack* mode the antenna follows the tracking profile defined by the data in the tracking file. The *Manual* mode is used primarily to perform the calibrations described in Section 6.0. In either mode the actual position of the antenna (in *degrees* and *counts*) is displayed in the set of four indicators located in the upper right-hand corner of the screen.

2. The center-left group provides information relating to the selected tracking file. That includes (a) the name of the tracking file, (b) the name of the satellite, (c) the starting date and time for the file data, and (d) the ending date and time for the file data. The group also includes a switch that allows the operator to reload the tracking data from the tracking file. This serves two purposes: (a) it allows the operator to select another satellite for tracking, and (b) it allows the operator to load data from a new tracking file even though the previously loaded data has not expired.

3. The center-right group consists of a pair of indicators that display the current azimuth and elevation coordinates (in *degrees*) obtained from the tracking file. They are updated by default every 5 minutes. The values displayed in these indicators are calculated by interpolation from the data obtained from the tracking file.

4. The lower-left panel is an empty frame that is intended to contain the *NMEATime* digital clock readout. Once the *GEOTracker* program is installed and run, the *NMEATime* clock can be sized and positioned in this panel.

5. The lower-right group is associated with the calibration of the azimuth and elevation actuators. The *Resync* button and *Link* buttons initiate the calibration processes described in Section 6.1 and 6.3, respectively. Clicking on the *Resync* button re-initializes the azimuth and elevation shaft encoders. Clicking on the *Link* button grabs the current azimuth and elevation coordinates (in both *counts* and *degrees*), stores them as the *reference* coordinates, and posts their values to the user interface. Clicking on these buttons should be done only when it is necessary to perform one of the actuator calibrations described in Section 6.1 or 6.3.

Along the bottom of the user interface screen are three additional controls: (a) a *Frame Number* indicator that is included for diagnostic purposes (see Table 2), (b) an *About* button that posts a help screen, and (c) a *Quit* button that terminates the program.

3.2 Satellite Selection

At startup, an additional pop-up dialog window will appear from which the operator can select one of several pre-designated satellites for tracking. If no selection is made within 30 seconds, the program will default to the last satellite that was selected. It then proceeds to look for a current tracking file for the selected satellite. This same pop-up also appears whenever the program runs out of tracking data. When this happens the operator can then select another satellite if so desired, or proceed by loading a new tracking file into the Zip drive.

3.3 Locking/Unlocking the Screen

Normally the security option for the *GTmain* vi should be set to "Locked (no password)." This keeps the various indicators and controls on the user interface locked in place so that they cannot be inadvertently moved. It also hides the vi's block diagram, which contains the program code. In the event that the code must be changed, the front panel needs to be unlocked. This is done by (a) selecting the File menu item at the top-left of the screen, (b) selecting VI Properties…, (c) selecting the Security category, and (d) selecting "Unlocked (no password)." After making the necessary changes, restore the locked status to once again lock down the User Interface controls and indicators.

4.0 PROGRAM DESCRIPTION

The main (i.e. top-level) vi is *GTmain*. Its structure is that of a foundation While loop within which is a Case structure. The result is a *state machine* type of architecture. The *GTmain* state machine has eleven states numbered 0 through 10 wherein the eleven states correspond to the eleven frames of the Case structure. See Table 2. The program branches from one state to another as dictated by the logic contained within each state.

4.1 Initialization

At startup the program first calls the *NMEATime* subvi. That starts the *NMEATime* clock synchronization program. The program then enters the state machine beginning in State 0 (i.e. Frame 0). State 0 is used to initialize the various local and global variables, and to open the Com1 port for communications with the Antenna Controller. Among these variables are the *path* variables which carry the file locations throughout the program. If it is necessary to install the files in some folder other than the default folders, these path variables in Frame 0 will have to be changed accordingly.

Also at startup the program reads the *GTref.dat* file to determine which satellite was last tracked. It then initializes the actuator *calibration* and *reference* parameters to the values they had when that particular satellite was last tracked. The actuator calibration parameters are read from the satellite data array that is situated outside the foundation While loop. The *reference* parameters are read from the *GTref.dat* file. See Appendix B for details on the structure of the *GTref.dat* file.

From Frame 0 (i.e. State 0) the program branches to Frame 1.

Frame	State	Function
0	Initialization	Initializes the program parameters, serial port, and the local and global variables.
1	Load Tracking Data	Allows the operator to select the satellite and loads the tracking data.
2	Event Processor	Dwells here awaiting a timed or operator-induced event, then branches accordingly.
3	About	Provides info regarding version no., connections, file format, and tech support.
4	Status update	Periodically updates the satellite and antenna positions on the user interface.
5	Auto Move	Issues repositioning commands to the Ant. Controller, and updates the user interface.
6	Link Degs to Counts	Stores the current Az and El coordinates as the *reference* coordinates.
7	Resync	Directs the Ant. Controller to run the Resync process.
8	Manual Move	Positions the antenna to the manually entered coordinates (in *counts*).
9	Load new file	Posts the out-of-data alert and prompts operator to a load new tracking file.
10	Quit	Releases the Com1 port and terminates the program.

Table 2. Program States of the *GTmain* State Machine.

4.2 Satellite Selection and Tracking File Search

In Frame 1 a popup window is generated that prompts the operator to select one of several pre-designated satellites for tracking. If after 30 seconds no selection is made, the program will proceed with the previously selected satellite.

The program then searches for a current tracking file for the selected satellite. The search begins with the *TracFiles* folder on the C: drive. If it finds a file there with data that spans the current date and time, it loads the data from that file into the data array in *GTmain.* If it does not find a current file there, it then looks in the Zip drive in the *Z:\GEOTracker\TracFiles* folder, where Z is the drive letter of the Zip drive. If it finds a current file there, it then copies that file to the *TracFiles* folder on the C: drive and loads the data from that file into the data array as previously indicated. If it cannot find a current file on the Zip drive, the program will cease searching and branch to Frame 9 for operator intervention. If the search was successful, the program will branch to Frame 2.

The satellite selection and file search process described above occurs at startup and then again whenever the program runs out of data. The later occurs normally at midnight GMT. **To keep the program running continuously, the operator must be sure to insert a disk containing a new tracking file into Zip drive prior the midnight GMT changeover.**

4.3 Event Processor

Frame 2 serves as the program's base state. The program dwells in Frame 2 awaiting any one of nine possible events. These can be either timed events or operator-induced events as enumerated in Table 3. When one of these events occurs, the program will branch to the indicated frame. After completing the task associated with that frame, the program will then branch back to Frame 2 to await the next event.

Event	Description	Stimulus	Branch to
0	No event	n/a	Frame 2
1	Quit	Operator input	Frame 10
2	About	Operator input	Frame 3
3	Reload	Operator input	Frame 1
4	Link	Operator input	Frame 6
5	Resync	Operator input	Frame 7
6	Move To	Operator input	Frame 8
7	Load new	Out-of-data	Frame 1
8	AutoMove	Timed event	Frame 5
9	Status	Timed event	Frame 4

Table 3. State Machine Branching Events.

4.4 The About Screen

Frame 3 posts information about the program to the monitor. It serves as a quick reference to the program's version number, the serial data connections, the tracking file naming convention and data format, and developer / technical support information. From Frame 3 the program branches back to Frame 2.

4.5 Antenna Status Update

Frame 4 updates the User Interface. In particular it updates the antenna's actual position as sensed by the Antenna Controller, and the satellite's position as derived from the tracking file data. The position of the satellite is calculated by interpolation in the *GetAzEl* subvi. Additionally in Frame 4 the program looks for any loss of communications with the Controller or any Positioner faults that may be included in the status message returned from the Controller. Status updates are made regularly by default every 5 seconds. From Frame 4 the program branches back to Frame 2.

4.6 AutoTrack Moves

Frame 5 positions the antenna to coordinates derived from the tracking file data. The coordinates are calculated by interpolation within the *GetAzEl* subvi. The program converts the interpolated coordinates to *counts*, assembles them into a command string, and sends that command string to the Antenna Controller. The program then reads back the antenna's new position in the status message from the Controller and updates the User Interface accordingly. These autotrack moves are made regularly by default every 5 minutes. From Frame 5 the program branches back to Frame 2.

Note that because the antenna is directed to a pair of *interpolated* coordinates, the autotrack moves need not occur coincident with the times specified in the tracking file. In fact, the autotrack moves may be made at any arbitrarily chosen interval and still position the antenna on the track defined by the tracking file data.

4.7 The *Link* Process

Frame 6 stores the current azimuth and elevation coordinates as the *reference* coordinates. The *reference* coordinates are involved in the conversion between *degrees* and *counts*. See Section 6.3 for the basis of this conversion. From Frame 6 the program branches back to Frame 2.

4.8 The *Resync* Process

Frame 7 initializes the shaft encoders that indicate the position of the antenna. The encoder initialization process (aka the *Resync* process) needs to be done on only two occasions: (a) when the antenna is first set up, and (b) in the unlikely event that one or both of the actuators were to loose count, as for example would occur if an actuator were driven to one of its mechanical stops. The *Resync* process can be run when the mode selector switch is set to the *Manual* mode. See Section 6.1 for additional detail on the *Resync* process.

In the event that the *Resync* process hangs (due for example to an alarm condition on either axis), the process can be terminated by clicking again (i.e. resetting) the Resync button.

From Frame 7 the program branches back to Frame 2.

4.9 Manual Positioning

Frame 8 serves to manually position the antenna when the mode selector switch on the User Interface is set to *Manual*. The antenna is repositioned to the coordinates manually entered into the *Set count* controls on the User Interface. The program dwells momentarily in Frame 8 until the antenna actually attains the position specified in the *Set count* controls. From Frame 8 the program branches back to Frame 2.

4.10 Prompt for New Tracking Data

Frame 9 is called from within Frame 1 in the event that a current tracking file cannot be found. The operator is given the opportunity to load a new tracking file in the Zip drive, or to exit the program. From Frame 9 the program branches either to Frame 1 or to Frame 10 depending on whether the operator clicked on the Continue button or the Quit button.

4.11 Exit

Frame 10 releases the Com1 port, saves the satellite reference number and reference coordinates to the *GTref.dat* file, and then terminates the program.

5.0 TRACKING FILES

The tracking files provide the positional information needed by the *GEOTracker* program to track a particular GEO satellite from a given propagation terminal on the ground. The tracking data consists of a series of time-stamped azimuth and elevation coordinates. The coordinates are calculated using a conventional closed-form solution based on (a) the NORAD ephemeris for the satellite as given in terms of a two-line orbital element set, (b) the standard NORAD SDP4 orbital model and propagator, and (c) the lat/lon/altitude coordinates of the antenna of the propagation terminal. For more detail, see www.celestrak.com.

5.1 File Structure

Typically the tracking files will contain two weeks worth of data starting at midnight (GMT) on the first day and ending at midnight (GMT) of the last day. The program will accept longer duration tracking files, but the error in predicting the orbital position of the satellite would ordinarily become unacceptably large. At the end of two weeks the pointing error for the antenna located at NASA in Cleveland, Ohio is nominally 0.066° in azimuth and 0.022° in elevation. At the end of four weeks the pointing error grows to about 0.11° in azimuth and 0.04° in elevation. Thus, with an antenna having a nominal 0.5° beam width, it is important that new tracking data be generated on a bi-weekly basis.

CLE02326_14_ACTS.trk

```
52012800  211.72  33.71
52012805  211.69  33.67
52012810  211.67  33.62
52012815  211.64  33.58
52012820  211.62  33.54
52012825  211.60  33.49
52012830  211.57  33.46
52012835  211.55  33.42
52012840  211.53  33.38
52012845  211.51  33.35
52012850  211.49  33.31
52012855  211.47  33.28
```

Figure 3. Sample Tracking File (partial).

Figure 3 shows the start of a typical tracking file. The data is arranged in three tab-delimited columns:

Col 1 Time (in minutes, GMT, relative to Jan. 1, 1904)
Col 2 Azimuth (in degrees)
Col 3 Elevation (in degrees)

The time (in minutes) that appears in Col 1 is based on the time reference used by LabVIEW. (Note: 1/01/1904 = 0 minutes; 1/01/2003 = 52,070,400 minutes.) The time interval between records is 5 minutes. Typically a file covering a 2-week period will contain 4,032 records and will be 80 - 90KB in size. The *CLE02326_14_ACTS.trk* tracking file shown partially in Figure 3 is one that was generated to track the ACTS satellite from the Cleveland propagation terminal for the 14-day period from day 326 through day 339 of the year 2002.

There is essentially no limit to the number of days that may be included in any single tracking file. However, if it is determined that the tracking files will consistently be generated to cover in excess of a 1-month time span, then the initial size of the data array in the program should be increased. By default, the initial size of the tracking data array is set to 10,000 records (see *GTmain* Frame 1, subframe 0). At 288 records per day, that is sufficient to store 37.4 days-worth of tracking data. For longer periods of time, increase the number of records proportionally.

The tracking files are named so that the propagation terminal and satellite to which it pertains, as well the time span for which it applies, can all be discerned without having to open the file. The naming pattern is as follows:

STNyyddd_nn_ssss.trk

where

STN (2-4 char limit) is the propagation terminal designator.
yy (2 digits) is the year for the start of data in the file.
ddd (3 digits) is the day-of-the-year for the start of data.
_ is the underscore character.
nn (2 digits) is the number of days of data in the file.
_ is the underscore character.
ssss (2–4 char limit) is the satellite name.
.trk is the file extension, designating it as a tracking file.

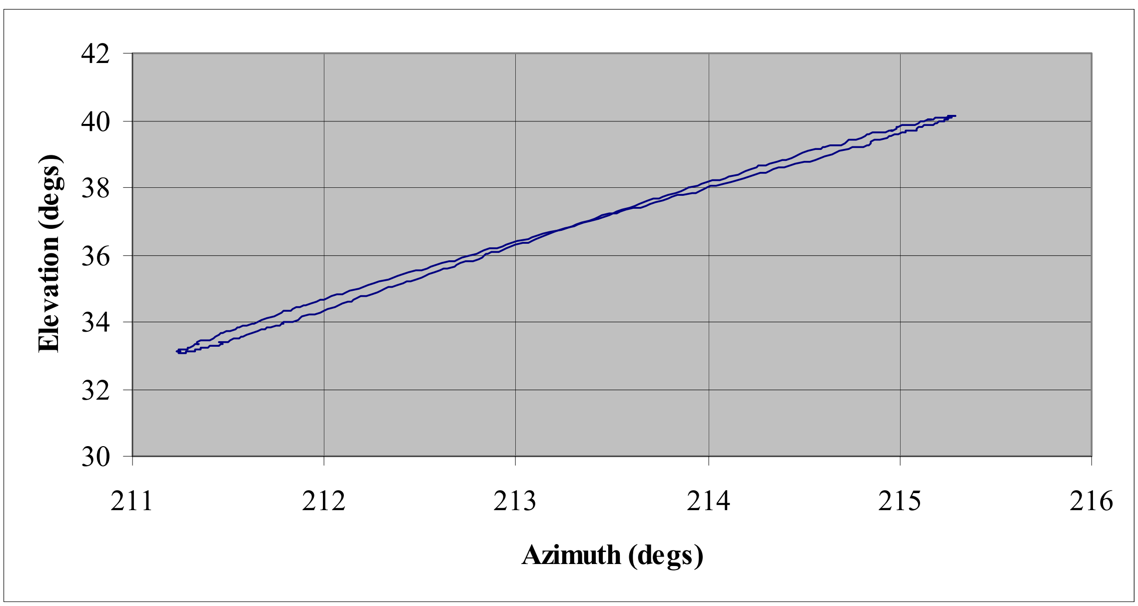

Figure 4. ACTS Measured Track for Day 322/2002.

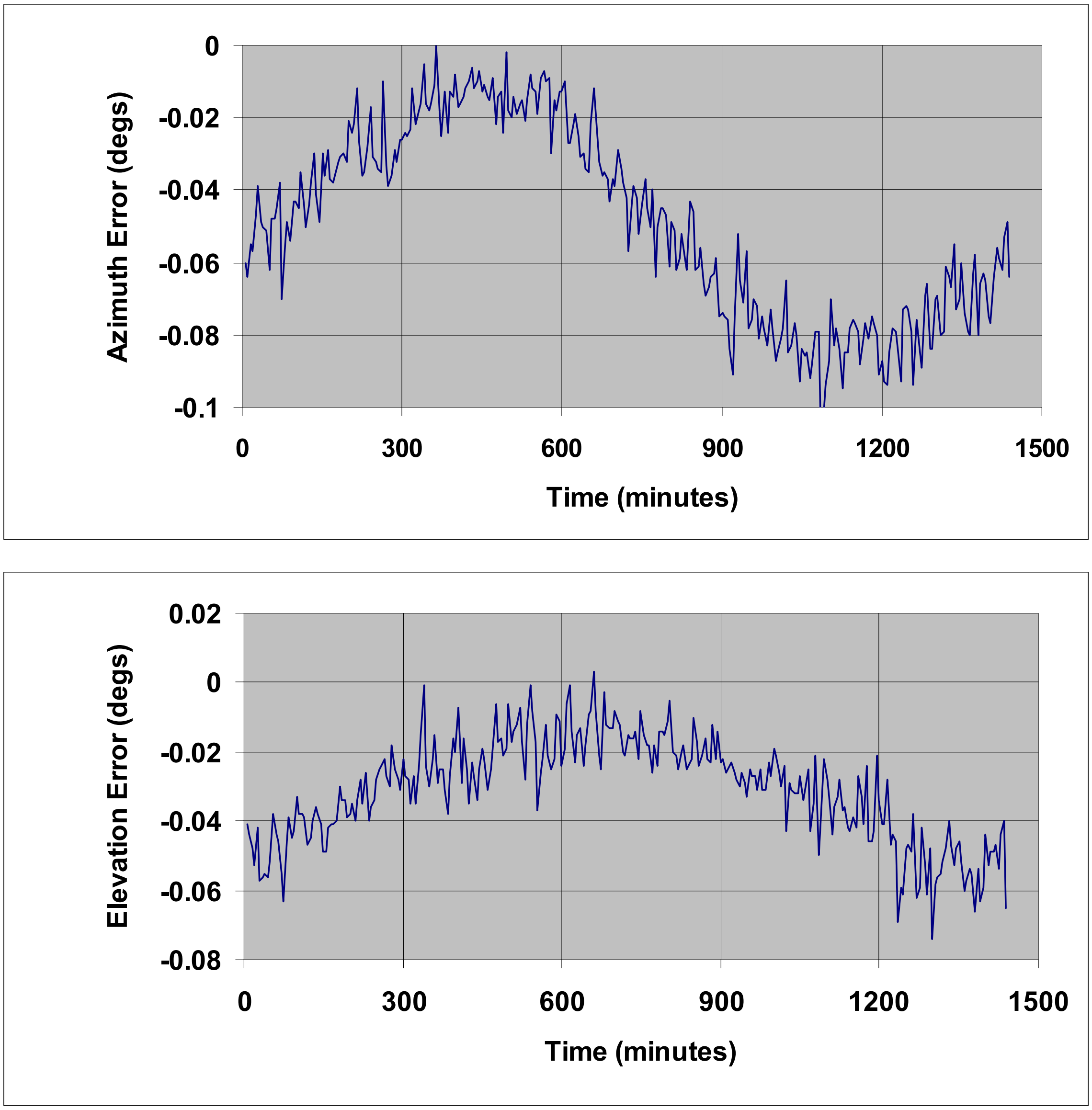

Figure 5. Difference Between the Predicted and Measured Coordinates for ACTS on Day 322/2002 .

5.2 Verification of the Tracking File Data

The validity of the method of calculation used to generate the tracking files was verified by a direct comparison of the predicted coordinates to the observed coordinates for the ACTS satellite as measured by the NASA Ground Station (NGS). Plotted in Figure 4 is the orbital track of the ACTS satellite as measured by the NGS over a 24-hour period on day 322 of the year 2002. Plotted in Figure 5 are the differences between the predicted and measured values for the azimuth and elevation coordinates as a function of time throughout that same 24-hour period. The difference values were calculated for each point in the measured data set as:

$$\Delta Az_i = (Az_{pred} - Az_{meas})_i \qquad \text{and} \qquad \Delta El_i = (El_{pred} - El_{meas})_i$$

where ΔAz_i and ΔEl_i are the point by point errors between the predicted and measured values for the azimuth and elevation coordinates, respectively, and where the predicted coordinates Az_{pred} and El_{pred} were themselves calculated from the tracking file data set by interpolation.

Clearly, the predominantly one-sided, sinusoidal differences seen in the Figure 5 suggest that there is a systematic error mechanism affecting both the azimuth and the elevation coordinates. The source of this error is currently under investigation. At this writing, it appears that the error is possibly due to timing and pointing inaccuracies in the measured data from the NGS, and possibly to the method of propagating the data tracking data. However, since the worst case pointing error is less than 0.1 degrees, which is small as compared to the beamwidth of the antenna, its effect on the propagation measurements will be negligible.

5.3 Data Transfer Process

The *GEOTracker* program reads the tracking data from the tracking file and stores it internally in a data array. When the program runs out of data from that data set it will automatically search for a new tracking file with current data. This process is one that normally takes place at midnight GMT.

The search begins by looking first in the *TracFile* subfolder on the C: drive for a filename that indicates it has data which spans the current date. If it finds such a file, the search is suspended and the data from that file is read into the data array. If it does not find such a file, the search continues by looking in the *Z:\GEOTracker\TracFiles* folder on the Zip drive, where Z is the drive letter for the Zip drive. If the program finds a current file there, it will copy that file to the *TracFile* subfolder on the C: drive and load the file data into the data array. If the program cannot find a current file on the Zip drive, it will cease searching and post a message to the screen prompting the operator to load a new data file. **For continuous operation, it is important that a disk containing a new tracking file be inserted into Zip drive <u>prior</u> the midnight GMT changeover.**

6.0 ACTUATOR CALIBRATION

The RC2000 Antenna Positioner is designed such that the azimuth and elevation of the antenna varies linearly with the number of turns of the corresponding motorized lead screw. Quantifying that relationship is a three-step process. First, the shaft encoders on the lead screws must be initialized. This first step, which is described in Section 6.1, is referred to as the *Resync* process. Second, the rate of change of the azimuth and elevation angles with every rotation of the actuator shafts must be determined. This second step is described in Section 6.2. Third, the intercept of the linear relationships must be established. This third step, which is referred to as the *Link* process, is described in Section 6.2.

6.1 *Resync*

The RC2000 Antenna Controller positions the antenna in terms of the rotational *counts* of the two lead screws. The maximum dynamic range of the shaft encoders is 0 to 65536 counts. Mechanical stops on each actuator, however, limit the range of travel of the actuator. Typically when tracking GEO satellites, the *counts* will range from about 30 to no more than 1000. The operational dynamic range of the shaft encoders is established by running the *Resync* process.

The *Resync* process is managed primarily by the RC2000 Antenna Controller. It is a canned routine that proceeds sequentially by first running the azimuth actuator over it full dynamic range. It then runs the elevation actuator over its full dynamic range. *GEOTracker* completes the process by setting the actuators to their mid-scale positions, which by default is *count* 333. The entire process, which takes several minutes to complete, proceeds automatically without operator intervention.

6.2 Actuator Rate of Change

The linear functional relationships between the azimuth and elevation of the antenna in *degrees* and the rotational position of its corresponding actuator in *counts* can be expressed as follows:

$$Cnt_{Az} = m_{Az} \bullet Deg_{Az} + b_{Az} \quad \text{and} \quad Cnt_{El} = m_{El} \bullet Deg_{El} + b_{El}$$

where m_{Az} and m_{El} are the rate of change parameters (in *counts/degree*) for the respective axes. Determining the value of these parameters is best accomplished by direct observation of a GEO satellite that is undergoing large “figure 8” orbital variations.

The method of determination involves “peaking up” on the satellite at its two orbital extremes. The azimuth and elevation in *counts* for the two extremes can be found by manually aiming the antenna so as to peak the signal received from the satellite as it “rounds the corner.” The azimuth and elevation in *degrees* of the two orbital extremes is obtained from the tracking file data. Thus, m_{Az} and m_{El} can be calculated as follows:

$$m_{Az} = \frac{(Cnt_{Az})_{\max} - (Cnt_{Az})_{\min}}{(Deg_{Az})_{\max} - (Deg_{Az})_{\min}}$$

and

$$m_{El} = \frac{(Cnt_{El})_{\max} - (Cnt_{El})_{\min}}{(Deg_{El})_{\max} - (Deg_{El})_{\min}}$$

The slope parameters supplied with the *GEOTracker* program were determined from data obtained by tracking ACTS from the Cleveland installation on day 318/2002. The data for the orbital extremes is tabulated in Table 4.

Orbital Extremes	Counts	Degrees
Azim. max.	382	215.22
Azim. min.	286	211.19
Elev. max.	445	40.13
Elev. min.	201	33.02

Table 4. ACTS Orbital Extremes from the Cleveland Terminal.

Using the data given in Table 4, the m_{Az} and m_{El} parameters were determined to be:

$$m_{Az} = \frac{382 - 286}{215.22 - 211.19} = 23.82 \; counts/\deg$$

and

$$m_{El} = \frac{445 - 201}{40.13 - 33.02} = 34.32 \; counts/\deg$$

These values are stored in the satellite data array of the *GTmain* vi for ACTS. Given the possibility that different satellites may have different m_{Az} and m_{El} parameters, the array stores the m_{Az} and m_{El} parameters on a satellite-by-satellite basis. The m_{Az} and m_{El} values read from this array are transferred to the AzCPD and ElCPD local variables, respectively, during the initialization process that occurs in Frame 0 and in Frame 1, subframe 0.

6.3 Link

Linking completes the actuator calibration process. It sets up one point on the line that defines the relationship between *counts* and *degrees* as a *reference* point. When converting *degrees* to *counts* (and visa versa), the program uses the *reference* point coordinates as an alternative to using the b_{Az} and b_{El} intercepts. The method of conversion is based on the following.

As indicated above in Section 6.2, the transfer function relating azimuth *counts* to azimuth *degrees* is in general:

$$Cnt_{Az} = m_{Az} \bullet Deg_{Az} + b_{Az}$$

Thus, at the *reference* point:

$$(Cnt_{Az})_{ref} = m_{Az} \bullet (Deg_{Az})_{ref} + b_{Az}$$

Subtracting these two equations, one from the other, the following relationship is obtained for the azimuth coordinate:

$$Cnt_{Az} = (Cnt_{Az})_{ref} + m_{Az}\left\lfloor Deg_{Az} - (Deg_{Az})_{ref} \right\rfloor$$

By similarity, the relationship between *counts* and *degrees* for the elevation coordinate is obtained:

$$Cnt_{El} = (Cnt_{El})_{ref} + m_{El}\left\lfloor Deg_{El} - (Deg_{El})_{ref} \right\rfloor$$

APPENDIX A

PROGRAM FILES

GEOTracker Program:

C:\GEOTracker\

DegsToCnts.vi	23 KB
GetAzEl.vi	30 KB
GetStatus.vi	45 KB
GetTime.vi	25 KB
GetTracFile.vi	45 KB
GT1.ICO	1 KB
GTgbls.vi	8 KB
GTmain.vi	577 KB
GTref.dat	1 KB
GTvars.ctl	34 KB
NMEATime.vi	73 KB
ParseResp.vi	47 KB
ReadRef.vi	111 KB
ReadTracFile.vi	48 KB
Resync.vi	37 KB
SaveRef.vi	99 KB
SelectSat.vi	37 KB
SendMove.vi	62 KB
Wait.vi	12 KB
YrDoy2Secs.vi	20 KB

20 files Total: 1,325 KB

NMEATime, version 5.2:

INSTALL.LOG	3 KB
NMEATime.chm	60 KB
NMEATime.exe	237 KB
OLEACC.dll	41 KB
Unwise.exe	125 KB

5 files Total: 463 KB

APPENDIX B

GTref.dat

Structure and contents: Fixed-length fields as follows:

Record 1: ss Satellite number, 2 digits: 0, 1, 2, . . . corresponding to the index number of the satellite as it appears in the satellite data array of the *GTmain* Diagram. (2 chars. + LF/CR)

Record n: aaa.aa [tab] ee.ee [tab] AAAA [tab] EEEE
where aaa.aa (6 chars.) is the azimuth ref in degrees,
ee.ee (5 chars.) is the elevation ref in degrees,
AAAA (4 chars.) is the azimuth ref in counts, and
EEEE (4 chars.) is the elevation ref in counts.
(22 chars. + LF/CR)

APPENDIX C

RC2000 Antenna Controller Command Summary

Command message format

Start char:		
	Start-of-text:	STX = 02H (2 dec.)
	Acknowledge:	ACK = 06H (6 dec.)
	No ack:	NAK = 15H (21 dec.)
Address:		Fixed at 32H (50 dec.) for RC2000 ver. F
Command Char.:		Various (see below)
Data bits:		
End-of-Message:		ETX = 03H (3 dec.)
Checksum char:		Bit-by-bit exclusive-OR of all characters in the message string from STX through (and including) ETX.

Response Messages

Unrecognized msg:	Byte 0:	NAK
	Byte 1:	address (32H)
	Byte 2:	cmd. code of the unrecognized message
	Byte 3:	ETX
	Byte 4:	Checksum
Status reply:	Byte 0:	ACK
	Byte 1:	address (32H)
	Byte 2:	31H
	Bytes 3 - 12:	Sat_name (may be blank)
	Bytes 14 - 18:	Azimuth (0 - 65535), or if limited then " CCW " or " CW "
	Bytes 19 - 23:	Elevation (0 - 65535), or if limited then " DOWN" or " UP"
	Bytes 24 - 25:	Pol. position
	Byte 26:	Pol. code and status
	Byte 27:	Azimuth movement/alarm status

Byte 27 bits:

7 6 5 4	3 2 1 0	
0 0 1 0	0 0 0 0	No alarm or movement
	0 0 1 0	East movement pending
	0 0 1 1	West movement pending
	0 1 0 0	East movement in progress
	0 1 0 1	West movement in progress
	0 1 1 1	Auto move is in progress
	1 0 0 0	Runaway alarm active
	1 0 0 1	Jammed alarm active
	1 1 0 0	Drive alarm active (overcurrent condition)

Byte 28: Elevation movement/alarm status

7 6 5 4	3 2 1 0	
0 0 1 0	0 0 0 0	No alarm or movement
	0 0 1 0	Down movement pending
	0 0 1 1	Up movement pending
	0 1 0 0	Down movement in progress
	0 1 0 1	Up movement in progress
	0 1 1 1	Auto move is in progress
	1 0 0 0	Runaway alarm active
	1 0 0 1	Jammed alarm active
	1 0 1 0	Limit alarm active
	1 1 0 0	Drive alarm active (overcurrent condition)

	Byte 29:	Pol. movement status
	Bytes 30 - 31:	Alarm codes
	Byte 32:	Pol. position info
	Byte 33:	Track mode
	Byte 34:	AGC
	Byte 35:	Pol. status flag
	Byte 36:	ETX
	Byte 37:	Checksum

<u>Command Messages</u>

Status query:	Byte 0:	STX
	Byte 1:	address (32H)
	Byte 2:	31H
	Byte 3:	ETX
	Byte 4:	Checksum
Auto Move:	Byte 0:	STX
(Form 2)	Byte 1:	address (32H)
	Byte 2:	32H
	Byte 3:	Pol. (must be blank)
	Bytes 4 - 13:	Az/El position aaaaaeeeee wherein count values are right-jusified and zero-padded to 5 characters each.
	Byte 14:	ETX
	Byte 15:	Checksum
Azim / Elev Jog:	Byte 0:	STX
	Byte 1:	address (32H)
	Byte 2:	33H
	Byte 3:	Direction: Azimuth values: "E" or "W" or "X" Elevation values: "D" or "U" or "X" where X = stop moving
	Byte 4:	Speed: F = Fast, or S = Slow. (Include F or S when specifying X)
	Bytes 5 - 8:	Duration: Range: 0000 - 9999 milliseconds
	Byte 9:	ETX
	Byte 10:	Checksum
Drive Reset:	Byte 0:	STX
(**Resync**)	Byte 1:	address (32H)
	Byte 2:	36H
	Byte 3:	"A" or "E" (or "P")
	Byte 4:	"R"
	Byte 5:	ETX
	Byte 6:	Checksum
		Note: To enable/disable AutoPol mode, set byte 3 = "P" and byte 4 = "N" or "F" for on/off.

NOTES