SLR 2000 Remote Access Terminal

Randall L. Ricklefs, The University of Texas McDonald Observatory
Jan L. F. McGarry, NASA Goddard Space Flight Center

Unlike current satellite laser ranging stations, the SLR2000 systems are to run autonomously, requiring no operator intervention or control on a regular basis. However, during the initial system development and debugging as well as during regular maintenance and upgrades of these stations, some type of human interface beyond a shell prompt will be required. To fulfil this need the Remote Access Terminal (RAT) is being developed. It will allow the operator to examine current ranging system status and faults, display current and recent data of many types, and override system operating parameters and decisions. This is accomplished by running an X windows/Motif-based graphical interface program on a laptop computer connected via the Internet. The RAT accesses the SLR-2000 system via an information and command server on the SLR2000 Data Analysis system (DAN) across a TCP socket interface and via Network File System (NFS). The RAT is being built for a PC laptop running the freely available Unix-like operating system Linux (Red Hat distribution).

Introduction

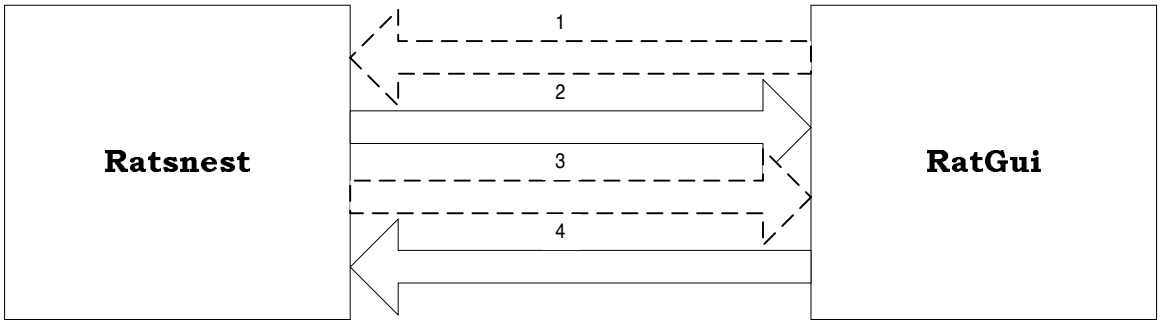
Current laser ranging systems are computer intensive, but still require one or more operators to actually take ranging data. The SLR2000 system is to break this mould and create a system that can range for months without human intervention. It will be necessary to check the station performance, retrieve data and replace predictions on a daily basis. This will be accomplished by standard automated ftp or login techniques. What is needed beyond this is a means to interact with the stations while on site or across the Internet during development or maintenance efforts and to look at what the system is seeing, what and why it is doing certain things, and to occasionally override the system. This will not be considered normal operations. To avoid the cost (monetary, space, and weight) of having a monitor on site, this work will generally be accomplished with a laptop computer carried to the site by the developer or technician.

Software Environment

To accomplish these objectives, several services are needed. First, one needs to know what the subsystems are doing. To learn this, a program (termed the “Ratsnest”) is to reside on the data analysis computer (DAN) and have access to status information in memory shared between DAN and the Pseudo-Operator (POP) computer systems. This status server communicates with the dedicated graphical user interface dubbed “Ratgui” running on the Remote Access Terminal (RAT) laptop computer as can be seen in Figure 1. When the Ratgui executes it asks for a socket connection with the Ratsnest. The Ratsnest responds by fulfilling the request. This becomes the command socket through which the Ratgui requests various services or actions, such as overriding a decision or requesting status information. The Ratsnest then requests a socket from the Ratgui, which the Ratgui accommodates. This is the status or 'gui' socket through which ratsnest periodically sends status and specially requested information.

In addition to the status, the RAT needs to display various types of files. Rather than passing these explicitly through the Ratsnest, the RAT also has DAN and POP file systems mounted via NFS accessible through the Ratgui. Thus the Ratsnest and NFS provide access to all vital information on the ranging system.

Whenever DAN is operating, the Ratsnest will be running and waiting for requests. Ratgui can then be run from anywhere across the net. There can be multiple copies talking to a single station. For safety reasons, only those logged in on the local network or from specially designated hosts can actually intervene in the operations of the station through this interface.



Host: Data Analysis Computer (DAN)

Multi-threaded Socket Server

Accesses POP/ DAN Shared Memory

Can support multiple instances of RatGui

0) DAN listens for socket request(s)

1) RAT requests command socket

2) Command socket established

3) DAN requests data/reply socket

4) Data/reply socket established

Host: Remote Acess Terminal (RAT)

Dual-threaded Socket Client

Accesses Shared memory via Sockets

Accesses Files via NFS

Figure 1 - DAN/RAT Communications

The Ratgui is primarily a display program built with c, X windows, and OSF Motif. Ratsnest is also written in c but has no interaction with the user. It is basically a multi-threaded data server or daemon.

The Ratsnest software can run under either Linux or LynxOS. Currently LynxOS is the operating system running on DAN, where the Ratsnest will normally execute. The Ratgui runs under Linux on the laptop, although it, too, can be run under Lynx. Lynx and Linux are both UNIX-like operating systems with similar extensive development software. While Lynx is a moderately priced POSIX-compliant real-time operating system, Linux is a free, general-purpose, mostly POSIX-compliant operating system that is quite robust and well documented. Both make extensive use of GNU compilers and utilities.

User Interface

As can be seen from figure 2, the ratgui program is a typical graphical interface program with a menu bar and work area. The work area consists of 4 distinct sub-areas. The top is a button area that has immediate control over the ranging system. This includes the 'abort move' and 'emergency stop' buttons. Next is an alarm area that signals abnormal conditions in the system. Some of these will actually flash by alternating between red and white foreground and background. Below this is a scrolled status display that shows the Ratsnest's response to the most recent Ratgui commands.

Finally, there is a status display that shows the current telescope and dome positions, weather, and system status information. The status indicators are color coded to indicate the level of functioning or malfunctioning. The indicator colors range from green, meaning all is well, to black, indicating system shutdown is imminent. Pressing the button for any computer system's status pops up a window showing the details of any faults.

Most of the additional functionality of the system is provided through the menu bar. There are 6 entries: Files, Tools, Control, View, Special, and Help. The Files menu items provide access to all of the relevant file types and an exit from the program. These entries allow selection of a particular file in the way of any Windows or Macintosh file menu. Many files are merely displayed in a text window without further elaboration, at least for now. Others invoke more involved displays.

The debug files, containing information on range machine basic data elements, mount positions and rates, and dome status have a series of pop-ups allowing the user to choose a certain time span and fields of interest from any one of the debug types (figure 3). These are then displayed in a spreadsheet pop-up (figure 4). The user can then create new columns that are simple mathematical combinations of other columns. This is meant to be a debugging and development tool, not a full-function spreadsheet. (There will be a separate commercial spreadsheet program available for more extensive work.)

Ranging files are displayed with 4 graphs in one paned window with sashes (figure 5). These include O-C residuals and range gate boundaries, azimuth and elevation offsets, mount model, and scan values, as well as a satellite ID and iris display. An additional graphic is a polar sky plot showing the satellite track. Post-fit ranging residuals are shown in figure 6.

Star calibration (mount model) data is shown in 8 plots, 3 being polar plots of data, model error, and the mount model corrections as a function of azimuth and elevation. Scatter plots show errors plotted against each other as well as the telescope azimuth and elevation. The user has the option to choose one or 2 mount model files to display at once for the sake of comparisons. A sample is shown in figure 7.

The environmental data selection, labeled 'Health and Safety', allows the display of any of the file's field, such as barometric pressure, temperature, critical voltages and the like as a function of time.

Image files, those produced by the star, sky, and security cameras, are displayed using XV, a popular UNIX-based image manipulation program. XV can display many file types, including the FITS format chosen for SLR2000 camera images.

The Tools menu allows re-executing previous commands or manually entering commands. It will also be possible to execute commands from a command file.

Next is the Control menu that allows the user to override the system's autonomous choices. Most of these produce pop-ups allowing the user to fill in values and select options. The star control pop-up allows the user to select a star from a polar plot and list of available stars. The scheduler pop-up allows the user to select an item from the current schedule. The camera control pop-up is shown in figure 8.

The View menu allows the user to control which of the many useful graphs will remain visible at once on a relatively small notebook screen. The Special menu allows killing Ratsnest and rebooting or shutting down the RAT. Finally, there is a help menu with pop-ups of textual help information on various parts of the system.

Other Points of Interest

It is clear that there will occasionally be need for a spreadsheet to look at certain elements of station files. The are a couple of commercial spreadsheets under examination that will hopefully meet most needs. It may be necessary to process some of the files into a format that can be imported into a spreadsheet. This will require expansion of the Ratgui and the launching of the spreadsheet from that program.

With many laptops being sold with sound cards and speakers, some type of aural feedback may be added to this system for such things as error conditions. This would be relatively simple to do, and prototype code has already been developed.

Conclusion

Monitoring and controlling an SLR2000 station will be accomplished through client and server software hosted on a pair of Unix computer systems. These will provide access to various data files, displaying them in appropriate fashions and allowing some level of control over the station hardware. The software uses standard technologies such as the POSIX multi-threading, TCP sockets, NFS, and X windows / Motif. While much of the required functionality exists in preliminary versions of the software, much remains to be expanded and improved as the rest of the SLR2000 system matures.

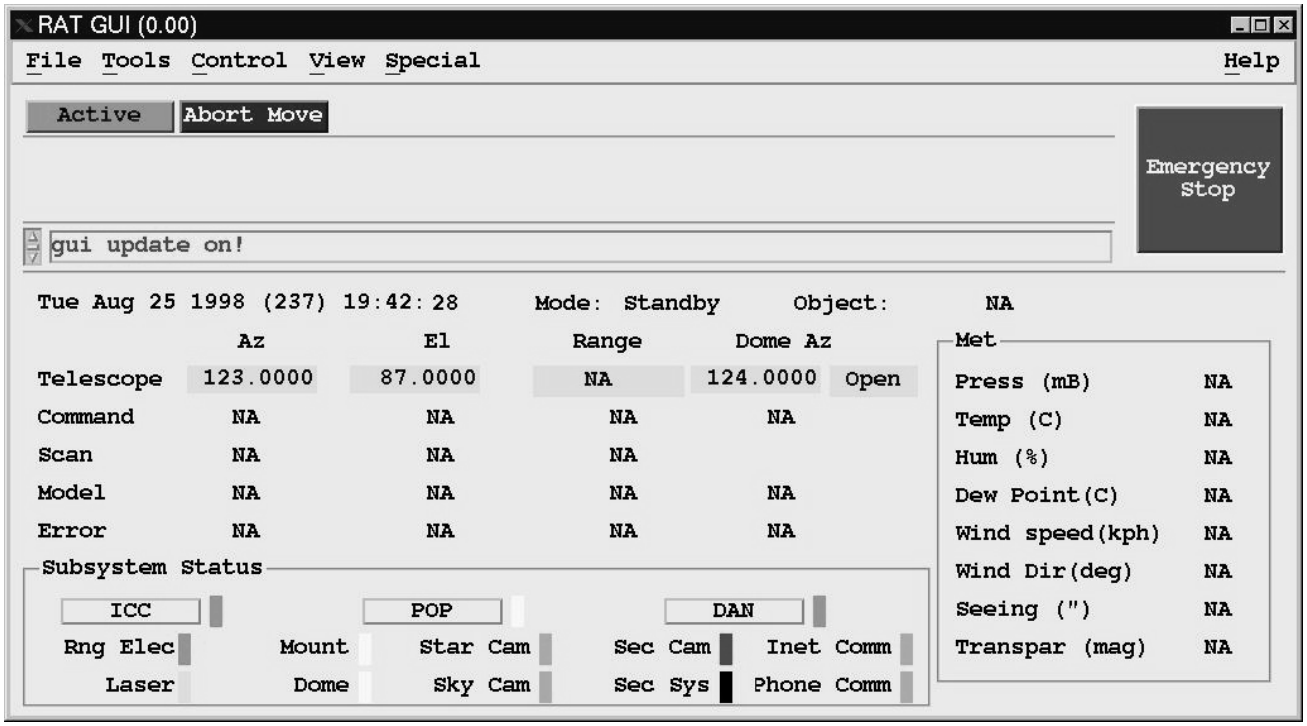


Figure 2 – RATGUI main display

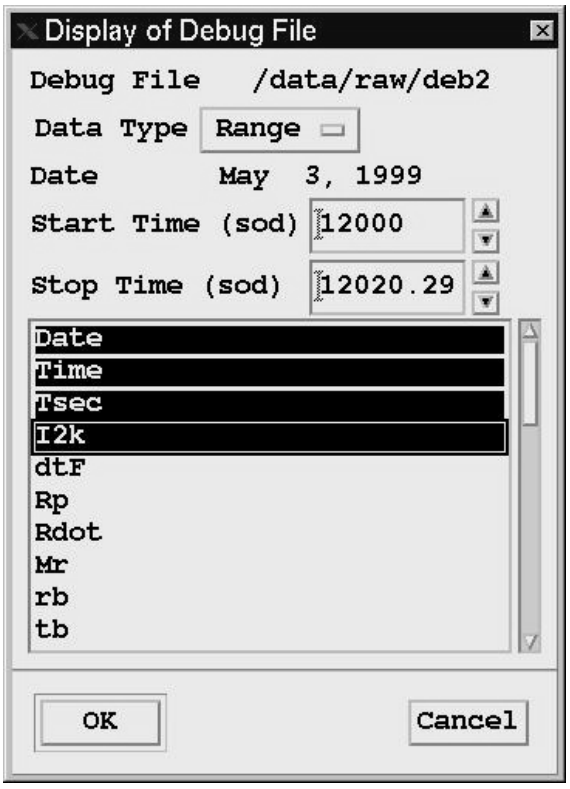


Figure 3 – RATGUI debug display time span and variable selection

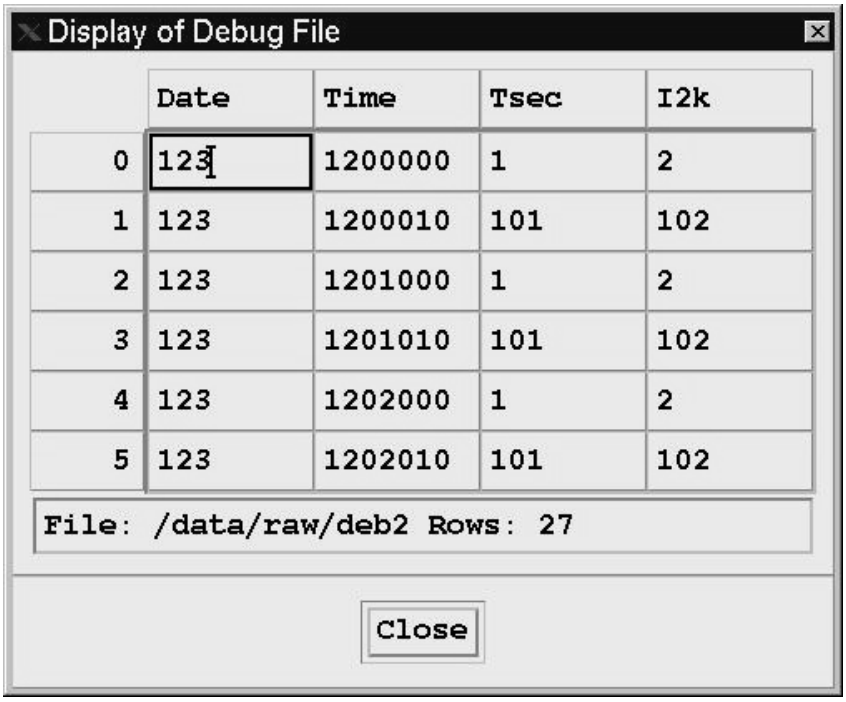


Figure 4 – RATGUI debug display spreadsheet

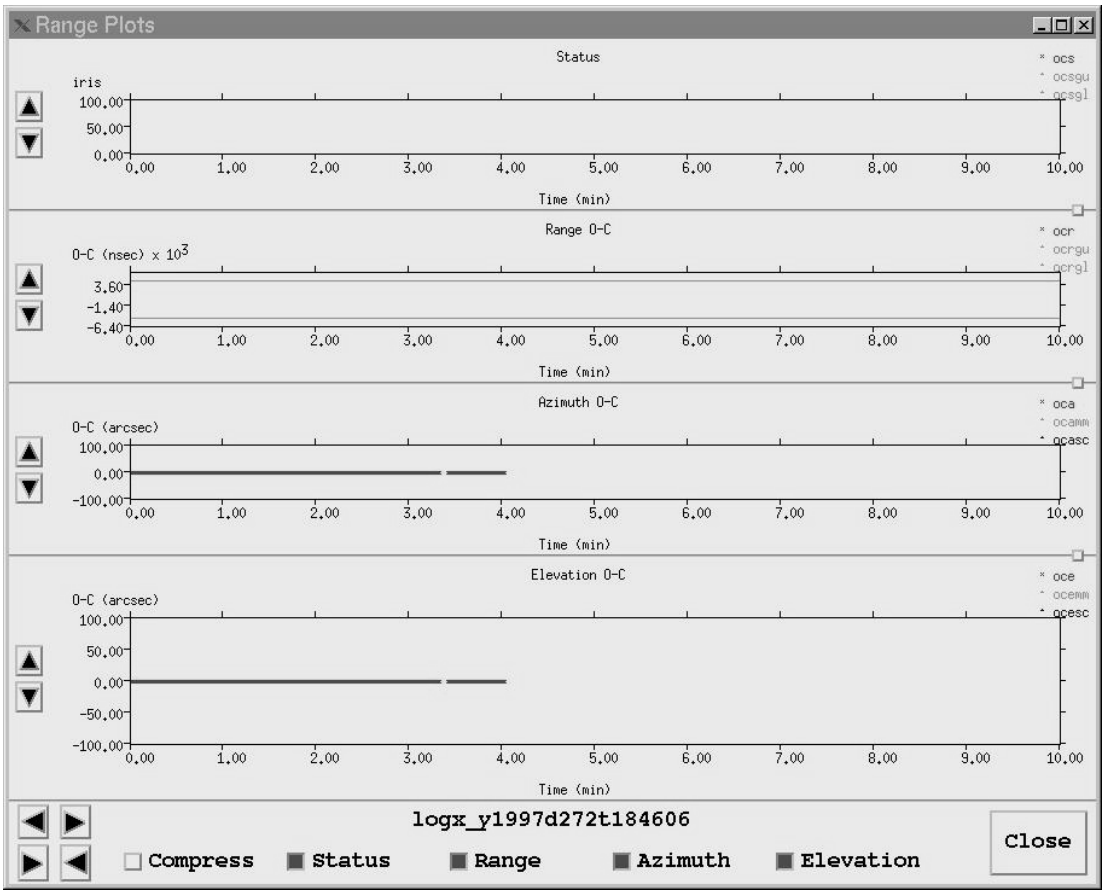


Figure 5 – RATGUI range data display

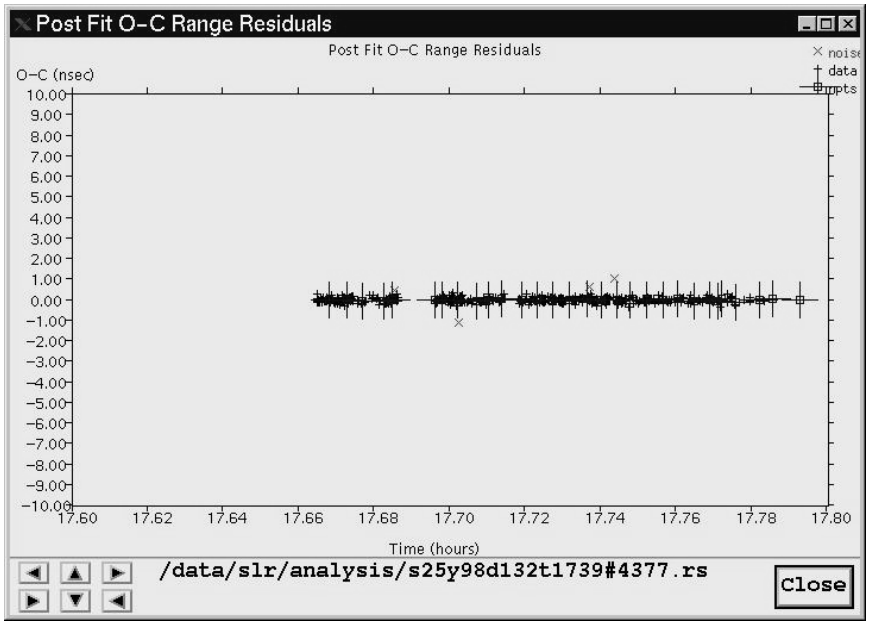


Figure 6 – RATGUI postfit residual plot

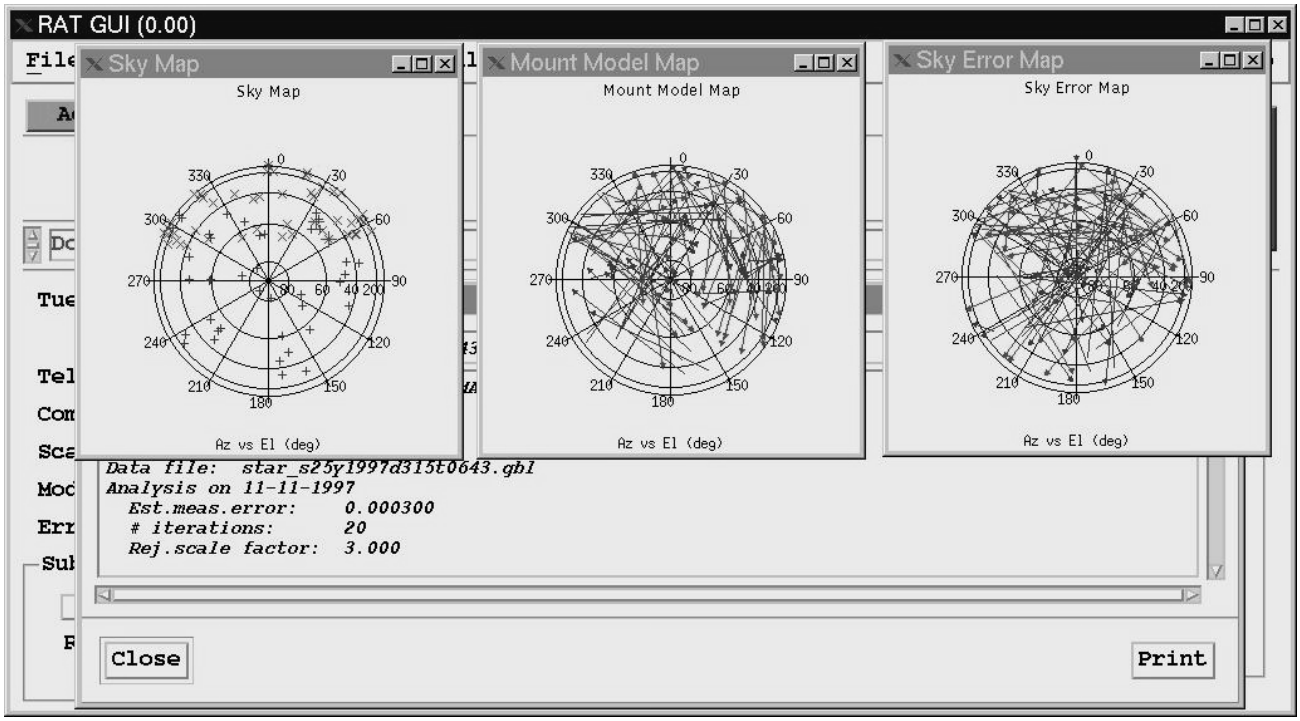


Figure 7 – RATGUI star calibration display (3 polar plots and solution summary)

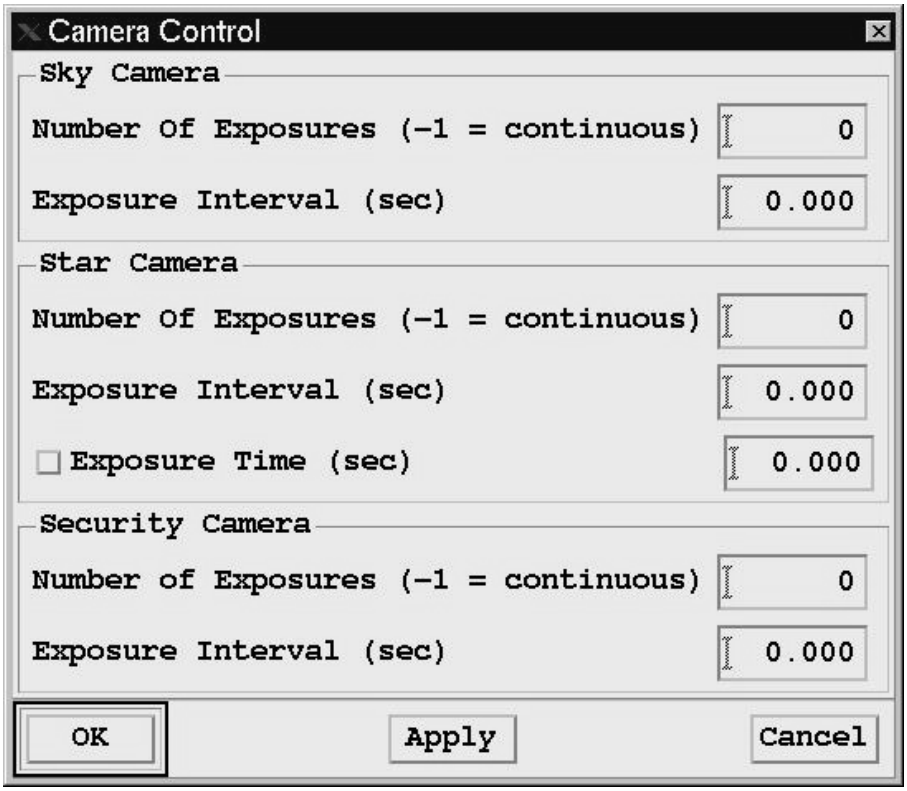


Figure 8 – Sample RATGUI control menu pop-up (Camera Control)