

# Controlling the Discovery Channel Telescope Using LabVIEW

Philip Taylor – Observatory Sciences Ltd.

## THE CHALLENGE

Designing and implementing control software to be integrated into a networked distributed control system for a large optical telescope.

## Products:

LabVIEW, LabVIEW Datalogging and Supervisory Control Module, NI CompactRIO, NI Compact FieldPoint

The Discovery Channel Telescope (DCT) is a 4.2 m aperture telescope under development by the Lowell Observatory in partnership with Discovery Communications. It will be the fifth largest telescope in the continental United States. It is a powerful tool for many areas of modern astrophysics, from studies of the solar system to fundamental work in stellar, galactic, and extragalactic astronomy. Located 40 miles outside Flagstaff, Arizona, on top of a cinder cone at a site known as Happy Jack, it is among the most technically sophisticated ground-based telescopes of its size. Projects planned for the DCT include a survey of Kuiper Belt objects orbiting the sun beyond Neptune, comprehensive studies of comets that could contain clues to the origins of the solar system, and research into the formation and evolution of galaxies.

## Overall Control System

A set of networked distributed systems that are responsible for making observations will control the DCT. These systems include telescope dome control, environmental controls, instruments, active optics, wavefront sensing systems that maintain the telescope's image quality, and a guiding system that automatically monitors and adjusts the tracking of the telescope.

We currently use LabVIEW for large physics control applications and have adopted it as the primary software development tool for the DCT control systems, with the distributed subsystems communicating with each other using LabVIEW shared variables. The observatory control system acts as a coordinating node and performs monitoring and control using the LabVIEW Datalogging and Supervisory Control (DSC) Module. The LabVIEW DSC Module uses the Citadel historical database to log shared variable data from the shared variable engine (SVE).

We use NI CompactRIO systems for intelligent control of the primary and secondary mirror supports of the telescope as well as for dome control. Also, we use NI Compact FieldPoint controllers in the environmental control systems.

## THE SOLUTION

Using NI LabVIEW software to create a system for controlling the telescope mount control hardware and to develop commands and status communication with other systems based on LabVIEW using LabVIEW shared variables.



Figure 1. Facility Rendering

## Telescope Control System

The telescope control system (TCS) is one of the subsystems that comprise the overall DCT control system. It is specifically responsible for:

- Calculating the precise pointing and tracking coordinates for the telescope, which has azimuth, elevation, and Cassegrain Rotator axes
- Controlling the telescope mount via the mount control unit provided by the telescope's manufacturer (General Dynamics SATCOM) and communicating over Ethernet using a TCP/IP socket interface
- Implementing an interface to several other DCT subsystems including the guider, wavefront sensor, active optics, time, weather, and dome

Observatory Sciences Ltd. developed the TCS software under contract for the DCT project. Similar to most of the other DCT subsystems, we implemented the TCS using LabVIEW running on Windows XP.

Some of the major reasons for adopting LabVIEW for this application were:

- Rapid prototyping and productivity gains that were reported using LabVIEW on similar projects, such as the South African Large Telescope (SALT)
- An integrated development environment that provides a GUI as an integral part of the VI development

- Simple integration of existing software libraries in ANSI C (with a small amount of FORTRAN)
- Easy integration of Internet facilities such as FTP and Telnet access to remote data providers

### Common Software Architecture

Observatory Sciences Ltd. is currently producing control systems for several telescopes that have a common set of facilities as well as many features and subsystems that are specific to each telescope. We designed a flexible software framework based on LabVIEW to support the development of large-scale distributed systems. This provides a set of common facilities that significantly reduces the amount of bespoke software required for each telescope.

*“LabVIEW enabled rapid prototyping using a graphical development environment with easy integration of external application libraries as well as built-in debugging facilities.”*

The common software, as its name implies, is intended to be reusable across a diverse set of telescopes and portable between LabVIEW running on both Linux and Windows OSs. It supports both single- and multitelescope arrays with a local or distributed UI and can be easily integrated into high-level control systems that do not use LabVIEW.

The common software provides a set of software services and an infrastructure to support the component model used to construct a specific TCS application. A large amount of ANSI C code is incorporated from various scientific packages and integrated into LabVIEW via external libraries.

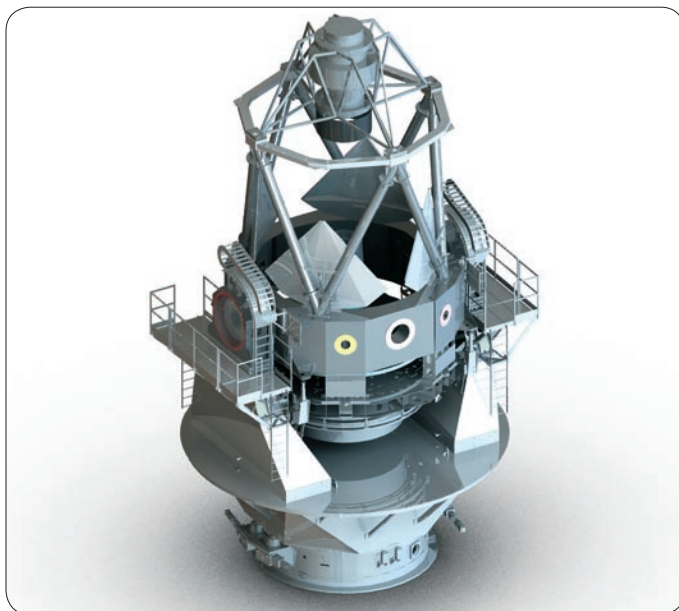


Figure 2. Mount Rendering

### Components and Commands

Software components are at the heart of the TCS software architecture. A component has a standard structure characterised by a set of commands that it responds to. It also maintains an overall health and activity status. The component VI library implements the common functionality that all components within the TCS must have.

A typical component is designed to handle interactions with a particular subsystem or piece of physical hardware. For example, in the DCT system, the components include those that control the mount, read the weather station data, and handle the data flowing to and from the automatic guiding system. A component can encapsulate any desired functionality so we also have components that update the International Earth Rotation Service (IERS)

parameters over the Internet and manage the pointing corrections for the telescope.

The command set that a component responds to is defined as an array of structures made up of the command name,

the parameters it requires, and a timeout. Each command has a command handler VI defined and the command is launched by invoking a dynamic VI.

### Pointing Kernel

The pointing kernel of the TCS calculates the required position, including azimuth and elevation, of the desired science target. For the TCS, all access to the functionality of these libraries is through the LabVIEW Call Library Function (CLF) nodes. We wrote a shareable library of wrapper functions for each function, which is encapsulated as a separate VI. The target positions are calculated and sent to the telescope's motion control system 20 times per second. The position demands are supplied with timestamps specifying an absolute time of day and the motion control system implements the demanded position at the specified time.

### Conclusion

We used LabVIEW to control a large, distributed control system and develop a general-purpose software infrastructure, which we deployed in several large software applications used to control professional optical telescopes. The DCT was the first system we completed with this infrastructure. Using the LabVIEW graphical development environment, we achieved rapid prototyping and easy integration of external application libraries as well as built-in debugging facilities.