The portable telescope control system project

Jeremy Bailey[a] and Richard Prestage[b]

[a]Anglo-Australian Observatory, PO Box 296
Epping, NSW 2121 Australia

[b]Joint Astronomy Centre, 660 N A'ohoku Place, Hilo
Hawaii 96720 USA

ABSTRACT

The Portable Telescope Control System (PTCS) project is a collaboration between the Anglo-Australian Observatory and the Joint Astronomy Centre, Hawaii. The project aims to develop telescope control software in a form which is portable between a wide range of computer systems, and which can easily be adapted to different telescopes. PTCS uses the DRAMA software environment which provides a high degree of operating system independence. The PTCS design is based on tested algorithms used in existing successful telescope control systems. The initial version of PTCS is now being interfaced to an EPICS based drive servo system for use with the James Clerk Maxwell sub-millimetre telescope on Mauna Kea.

Keywords: Telescope Control Systems, DRAMA, EPICS

1. INTRODUCTION

Telescope control software has up to now, generally been written specifically for each new telescope and typically each project requires several years of software effort. In many cases the telescope control software gets rewritten after only a few years as the system is found to be inadequate or the hardware it runs on becomes obsolete.

The PTCS project is motivated by the fact that the basic algorithms required for telescope pointing and tracking are now well understood, and it should therefore be possible to produce a system which is sufficiently general to meet the needs of a wide range of different telescopes. Such an approach avoids unnecessary duplication of effort. If such a system were widely adopted the standardization would bring other benefits. New telescopes would be able to adopt a system with a clearly defined and documented method of operation, and one which had been demonstrated to be accurate, efficient and thoroughly tested. Standard interfaces to the telescope would make it easier to move instruments between telescopes, and to provide generic software tools which interface smoothly with the telescope (such as catalogue access, solar system ephemerides, telescope pointing analysis etc.).

2. SYSTEM OVERVIEW

The basic aims of PTCS are to provide two different aspects of portability:

- Portability between different telescopes.
- Portability between different computer architectures and operating systems.

Further author information -
J.B.: Email: jab@aaoepp.aao.gov.au; Telephone: 61-2-9372-4823; Fax: 61-2-9372-4880
R.P..: Email: rmp@jach.hawaii.edu

SPIE Vol. 3112 • 0277-786X/97/$10.00

2.1. Portability between Telescopes

To achieve portability between different telescopes the system has been designed with an architecture which cleanly splits the generic aspects of the software such as the basic pointing and tracking algorithms, from the information that is specific to the implementation for one telescope.

The system is designed to provide a rich set of features which encompass the needs of telescopes operating at a variety of different wavelengths. The AAO and JAC, between them, operate telescopes working at optical, IR and sub-mm wavelengths and the collaboration therefore has wide experience of the different needs of telescopes operating at different wavelengths. Any one telescope may only need a subset of the features provided by PTCS for its standard operations, but the extra functionality in the system may prove valuable in meeting the need for occasional non-standard operating modes.

Of course, we cannot expect to anticipate all possible future requirements of a telescope control system and PTCS has therefore been designed with the ability to be build on and extend the basic set of features in a number of different ways to provide for future needs.

2.2. Portability between Computer Architectures and Operating Systems

The lifetime of a telescope is much longer than the typical lifetime of a computer system, and it is therefore important to design software that is not dependent on specific computer hardware or operating systems if it is not going to need frequent rewriting over the lifetime of the telescope.

This is achieved for PTCS by making use of the DRAMA software environment[1,2] developed at the AAO. DRAMA is designed to support the needs of real time distributed systems which may include a range of different operating systems. DRAMA provides an efficient inter-process and inter-processor message system (IMP), a hierarchical data structuring system (SDS)[3] which automatically handles data format conversion between different machine architectures, and a model of task objects which respond to messages requesting them to perform actions.

DRAMA has been implemented on a range of operating systems including various versions of Unix, the VxWorks real time operating system, and VMS for VAX and Alpha. DRAMA has been used to implement the instrument control and data reduction systems for the two degree field (2dF) project on the AAT. It it also being used by the Royal Greenwich Observatory for the ING telescopes on La Palma, and by the Joint Astronomy Centre in Hawaii. Components of DRAMA (IMP and SDS) are also being used in the Gemini and ESO VLT projects.

By implementing PTCS in the form of DRAMA tasks the system is made easily portable between all the operating systems supported by DRAMA. As well as DRAMA libraries and ANSI C Libraries, the only other package called from PTCS code is the SLALIB package written by Patrick Wallace.[4] SLALIB is written in standard C and compiles on a wide range of systems.

2.3. Algorithms

The basis of the algorithms used in PTCS is the design developed by Wallace for the Keck telescope.[5] Essentially the same algorithms have been used in successful telescope control systems for UKIRT[6] and the WHT[7] as well as the Keck itself. The algorithms for the Gemini telescope control system[8] currently under development are also very similar.

The Wallace design specifies in detail how a tracking position, typically specified as a mean RA and Dec, is transformed into a demand position for the telescope axes, taking into account relevant astrometric effects, telescope misalignments and other pointing corrections. Performing this transformation repeatedly at a rate of typically 20Hz, generates a stream of demand positions which can be used as input to the telescope drive servos.

To provide an implementation which could be run efficiently on the CPUs then available, Wallace proposed splitting the calculations into three sets which are performed at different frequencies (and in actual implementations by different processes). A SLOW process running at a repeat rate of about 5 minutes calculates slowly varying quantities such as the refraction parameters and parameters needed for the mean to apparent transformation. A MEDIUM process repeating every five seconds calculates two *osculating transformation matrices* (OTMs). The OTMs each represent a series of steps in the full pointing transformation, with sufficient accuracy to be valid over a small area of sky. The FAST process repeats at the full 20Hz rate and uses the OTMs to perform the pointing transformation more economically than would be possible if it was done in full. With this scheme Wallace was able to do the pointing calculations with a CPU loading of 7% on a Micro-Vax II.

PTCS does not use the OTM scheme and therefore requires only a SLOW and a FAST process and not a MEDIUM process. It does the full pointing transformation at the fast loop rate with the only precalculated information being that from the slow process. This leads to a much simpler implementation. It means that all intermediate steps in the pointing transformation are always available. It becomes much simpler to change to a new source or to a different tracking coordinate system as there is no need to calculate a new set of OTMs. It is also much easier to implement an upstream transformation from the encoder position.

Timing tests have shown that modern CPUs such as the PowerPC, Alpha and Sparc can perform the full pointing transformation from B1975.0 (an FK4 system with a non-standard equinox is the slowest of the supported coordinate systems) in times ranging from 20 to 50μs. Thus even if several such transformations have to be carried out per cycle (as is usually the case with PTCS), the CPU loading is likely to be less than 1% for a 20Hz repeat rate. There is therefore no need to adopt the more complex OTM scheme.

2.4. The Telescope Hardware Interface

The PTCS tasks are designed so that the same task executable can be used for different telescopes. The telescope specific part of the code is another DRAMA task called the Telescope Hardware Interface (THI) task. The THI task is the source off all telescope specific information needed by the PTCS. It makes available in its task parameters information such as the telescope location (latitude, longitude and height), telescope pointing parameters etc. which the PTCS needs. It also provides the interface by which the PTCS controls the telescope drives and any other mechanisms (such as the dome, instrument rotator, chopping secondary). The THI task can specify which mechanisms are to be controlled, and the PTCS will only generate demands for the required ones.

The THI task is also the basic source of timing in the system. The THI task sends messages (DRAMA Trigger messages) to the PTCS which contain the current time and encoder readings. In response to this the PTCS returns a message (a Kick message) to the THI task containing the demand positions for the various mechanisms for the next cycle. The rate at which the THI generates its trigger messages therefore determines the repeat rate of the basic pointing loop. Typically this could be a precise interval driven by a hardware interrupt, but there is no requirement that the cycle be precisely regular. The PTCS generates time stamped demands with velocity information, so that they can be extrapolated if necessary.

The THI task needs to be set up specifically for each telescope. A template task is provided which can be modified to include the code required. The code for the drive servos could be incorporated directly into the THI task, or this task could simply provide the link to some other software system which is directly controlling the hardware. In the case of the JCMT telescope control system, the first actual application of PTCS, the drive servos are implemented as an EPICS database, so all that the THI task has to do is to make the necessary channel access calls to send demand values on to the EPICS systems, and return time and encoder information for use by the PTCS.

2.5. Telescope Simulations

The structure of PTCS means that it is easy to test PTCS without access to any telescope hardware. It is simply necessary to use a test version of the THI task which provides a simulation of the telescope hardware. Such simulations can include the parameters of any specific telescope, so that a realistic simulation of that telescope is obtained.

2.6. The Virtual Telescope Interface

The external interface to the PTCS system is referred to as the virtual telescope interface. The concept of a virtual telescope was introduced in discussion of the AAT control system.[9] The virtual telescope is easier to use than the real telescope because it moves in the user's chosen coordinate system rather than in a coordinate system determined by the telescope mount. The software hides, as far as possible, the deficiencies of the real telescope by applying software corrections, thus presenting a virtual telescope which performs better than the real telescope.

The following coordinate systems are supported in the PTCS:

Galactic Latitude and Longitude
Mean Right Ascension and Declination
(either new (FK5) or old (FK4) systems for any equinox)
Apparent Right Ascension and Declination
Apparent Hour Angle and Declination
Topocentric Azimuth and Elevation
Observed Azimuth and Elevation
Telescope Mount System

All these options are available for use as the input coordinate system for a new target object. The source position may have associated proper motion, parallax and radial velocity (for a mean place), or differential tracking rates (for a solar system object).

The same set of coordinate system options are available for use as the tracking system for the virtual telescope. The tracking system is the starting point for the pointing transformation. Tracking can be defined as adjusting the telescope mount position to keep the position constant in the tracking system.

The PTCS virtual telescope interface takes the form of a set of actions and parameters of the PTCS DRAMA task.[10] The task actions provide the following functions:

- Specifying and slewing to target objects.
- Telescope position adjustments (offsets, scans etc.)
- Creating, deleting and selecting virtual telescopes.
- Defining and selecting instrument apertures, and controlling pointing axis position.
- Control of instrument rotator position.
- Chopping secondary control and beam selection.
- Pointing model adjustment.
- Dome control.

The parameters of the PTCS task return information including the following:

- Information on the current and next target.
- The telescope tracking coordinate system.
- Selected instrument aperture and pointing axis position.
- Instrument rotator position and status.
- Chopping secondary beam positions.
- Current focal station and image scale.
- Pointing model parameters.
- Dome position and status.
- Time in various forms (TAI, UTC, UT1, TDB, LST).
- Current telescope position.

Any of the parameters can be read by other DRAMA tasks in the system or monitored (a message is sent whenever the parameter value changes). However, a GET_STATUS action is provided to support applications such as maintaining a status display in a user interface with optimal efficiency. The GET_STATUS action is given a list of parameters that the caller is interested in and a specified update interval. A message containing all the required information is returned repeatedly at the specified interval.

3. PTCS FEATURES

3.1. Multiple Virtual Telescopes

The concept of a virtual guide telescope in addition to the main virtual telescope was introduced in the UKIRT telescope control system.[6] The PTCS supports a more general facility which enables the creation of multiple additional virtual telescopes in addition to the standard MAIN virtual telescope which is always active. These can be used for a number of purposes.

3.1.1. Virtual Guide Telescope

A virtual guide telescope is used in conjunction with an autoguider, or with the tip-tilt component of an adaptive optics system. The target position of the virtual guide telescope is that of the required guide star. The guide star position is taken through a pointing transformation analogous to that for the main telescope. However, instead of generating a demand position for the telescope mount, it generates a demand X,Y position in the focal plane, at which the guide probe or wavefront sensor has to be placed to pick up the guide star, when the mount position is that required for the main virtual telescope. This process involves a simple inversion of the code normally used to apply the collimation corrections during the telescope pointing transformation.

This scheme provides a very simple way of calculating the required guiding position which involves largely reuse of the standard telescope pointing code. However, the scheme does much more than just position the guide probe on the star. It allows naturally for **active** guiding where the guide position is continually adjusted during tracking. This adjustment is needed to allow for the following effects which will vary during tracking:

- Atmospheric dispersion — i.e. the difference in refraction between the guiding and observing wavelength. This can be substantial for AO systems which perform wavefront sensing in the optical for observations in the IR.
- Differential refraction — i.e. the difference in refraction between the observing position, and the position of an off-axis guide star.
- Any differential flexure between the guider and observing instrument.

The size of these effects are such that an active guiding scheme is essential to meet the tracking specifications of the current generation of large telescopes. For example the Gemini specification requires 0.01 arc sec rms over 1 hour.[11]

A further advantage of active guiding is that it allows the telescope to perform offsets and scans while guiding. This is done by sending exactly the same offset requests to the main virtual telescope, as would be done if guiding was not in use. If autoguiding is in operation, the offsets requested for the main telescope will change the required X,Y position of the guide probe, so the telescope will make its movements with the extra precision deriving from the autoguider control. Furthermore both the main target or guide star can have differential tracking rates applied to them, enabling such schemes as performing a map of Jupiter while guiding on one of its satellites.

3.1.2. Reference Virtual Telescope

The reference virtual telescope is simply an alternative to the main virtual telescope which can be switched into use by the SELECT_VT action. Its application is in complicated beam switching operations such as when a complex scan on a solar system object is being performed, but the background reference is a fixed sky position. Rather than requiring several steps to get to the reference position (turn off differential tracking, stop scanning, offset to required position), it is much simpler to set the reference virtual telescope to track the reference position, set the main virtual telescope up with the differential tracking rates and the required scan, and switch control between the two virtual telescopes when it is required to go to background measurement.

3.1.3. Other Uses of Virtual Telescopes

A number of other uses for additional virtual telescopes can be envisaged. For example a design has been proposed for Gemini[12] in which a chopping secondary is controlled using multiple virtual telescopes to represent different chop beams. This scheme can involve up to 15 virtual telescopes. It is not currently proposed to use such a scheme in PTCS which incorporates a much simpler implementation of chopping secondary control. However, the virtual telescope concept has been implemented in PTCS in such a way that it will be relatively straightforward to add additional types of virtual telescope in the future.

3.2. Offsets and Scans

The tracking position of the virtual telescope is composed of a base position plus an offset from base. The offset from base may be expressed in any of the available coordinate systems and does not have to be in the current tracking system. The offset may be expressed as a direct offset in the coordinate values, or in one of a number of tangent plane projections. The available projections[13] are the gnomonic (or TAN) projection, the orthographic (or SIN) and the ARC projection.

A number of actions are available to control the offset, but the most powerful is the SCAN action which is designed to be used as a building block to implement rasters and other more complex scans. The SCAN action specifies a coordinate system and projection type, a starting position in each coordinate, a scan velocity in each coordinate, and a scan duration. The scan action set the demand position to the start of the scan line. When the telescope reaches this start position it then moves along the scan line at the specified velocity. After scanning for the specified duration a number of completion options are available such as moving back to the start of the line, zeroing the offsets, or stopping at the end of the line.

3.3. Chopping Secondary Control

If the telescope has a chopping secondary it can interface to the PTCS in either an active or passive mode. In the active mode the PTCS generates the desired beam positions for the secondary and passes these to the chopping secondary hardware through the THI task. The actual chopping of the secondary between the two (or more) beam positions is not controlled by the PTCS (and in any case may occur at a higher rate than the pointing loop rate). As far as the PTCS is concerned the operation of the chopper leads to two different pointing axes, with different collimation offsets. The PTCS can be instructed to select either of these chopper beams as the current one (selecting a chopper beam involves changing the telescopes collimation offset and therefore causing the telescope mount to move, to realign the virtual telescope to the same position as before)

The passive mode is similar, but applies to a chopping secondary which is independently controlled and not interfaced to the PTCS. In this case the PTCS is simply informed of the two beam positions, or alternatively finds them itself by means of a calibration procedure which involves driving the telescope to each in turn. Selecting a chopper beam is the same as in the active case.

3.4. Use with TPOINT

PTCS is designed to be used in conjunction with the TPOINT telescope pointing analysis package written by Patrick Wallace.[14] Terms can be added to the PTCS pointing model using the same names as used in TPOINT. At present PTCS supports a subset of the full set of terms used in TPOINT. It is planned to add to this list and eventually support all TPOINT terms. A testing procedure is being developed which will allow PTCS to be used to generate simulated pointing data which can then be fed to TPOINT. This will verify that the pointing model is interpreted consistently in the two packages.

3.5. User Interfaces

DRAMA supports a number of different user interfaces but the most successful have been those based on Tcl/Tk.[15,16]

A Tcl interface has been provided to invoke any of the actions in the PTCS task. Using these it is easy to set up scripts to automate a sequence of telescope operations. The script in figure 1 provides an example of how this interface can be used.

The Tk package can easily be used to construct a graphical user interface for use with PTCS. It is easy to incorporate rapidly updated displays of telescope status by using the GET_STATUS action in conjunction with a Tk display window. A user interface of this type has been developed for testing PTCS and could be used as a template for developing interfaces for specific telescopes.

Figure 1. Example TCL script for PTCS

```
#
#  All commands have their argument in radians - as2r is a procedure
#  to convert arc seconds to radians
#

proc as2r {x} {
    return [expr {$x * 4.84813681e-6}]
}

#  Enter a source position (B1950 coordinates) and slew to it

target NGC1068 B1950 0.698640 -0.003927
slew

#  Wait until the telescope is tracking the source

wait TRACKING

#  Do an offset 5 arc seconds north (in the J2000 system)

set_tel_offset J2000 DIRECT 0.0 [as2r 5.0]
wait TRACKING

#  Set the current telescope position as the base position
#    i.e. the zero point for offsets and scans

set_base_here

#  Do a raster scan about this position over a 20 arc second square.
#  The scan consists of lines 2 arc seconds apart scanned at
#  2 arc seconds per second. The scan is specified in J2000
#  gnomonic (TAN) projection, tangent plane coordinates

set y_start [as2r -10.0]
set x_vel 0.0
set y_vel [as2r 2.0]

for {set x -10.0} {$x <= 10.0} {set x [expr {$x + 2.0}]} {
    set x_start [as2r $x]

#  do_scan does one line of the scan

    do_scan J2000 TAN $x_start $y_start $x_vel $y_vel 10.0 START
}
```

4. PROJECT STATUS

An initial implementation of PTCS with most of the important features was completed in December 1996. A number of tests were carried out on this system to verify the implementation of the algorithms. The results of the pointing code were compared with other systems. For example, it was used to reproduce the numerical example developed for the Gemini project.[17] Timing tests were carried out to verify that CPU loading was minimal. Currently in progress is the implementation of all the remaining features required to bring PTCS up to the specification.[10]

The initial application of PTCS will be to provide a replacement telescope control system for the James Clerk Maxwell sub-mm telescope (JCMT) on Mauna Kea. An EPICS servo system based on that currently used on UKIRT will provide the interface to the telescope drive hardware. The JCMT telescope control system has to fit into an existing observation control system based on the ADAM environment. This should be relatively straightforward as an interface between DRAMA and ADAM exists. It should therefore be possible to provide an interface task which emulates the old JCMT telescope control system while sending its commands to PTCS. The system is expected to be commissioned later in 1997.

REFERENCES

1. T. J. Farrell, K. Shortridge, and J. A. Bailey, "DRAMA: An environment for instrumentation software," *Bull. Amer. Astr. Soc.* **25**, pp. 954–957, 1993.
2. J. A. Bailey, T. Farrell, and K. Shortridge, "DRAMA: An environment for distributed instrumentation software," in *Telescope Control Systems*, P. T. Wallace, ed., *Proc. SPIE* **2479**, pp. 62–68, 1995.
3. J. A. Bailey, "A self-defining hierarchical data system," in *Astronomical Data Analysis Software and Systems II*, R. Hanisch, R. Brissendon, and J. Barnes, eds., *A.S.P. Conf Series* **52**, pp. 553–557, 1993.
4. P. Wallace, "SLALIB — Positional astronomy library," SUN 67.35, Rutherford Appleton Laboratory — Starlink Project, 1996.
5. P. Wallace, "Pointing and tracking algorithms for the Keck 10-meter telescope," in *Instrumentation for Ground-Based Optical Astronomy*, L. B. Robinson, ed., pp. 691–706, Springer-Verlag, New York, 1988.
6. J. Bailey, "UKIRT telescope control system — system overview," USN 5.2, Joint Astronomy Centre, 1989.
7. L. Jones, "William Herschel Telescope — control system overview," La Palma Technical Note 44, Royal Greenwich Observatory, 1987.
8. P. Wallace, "Gemini pointing algorithms," TCS/PTW 3.8, Rutherford Appleton Laboratory — Gemini Project, 1996.
9. J. O. Straede and P. T. Wallace, "The Anglo-Australian 3.9-meter telescope: Software controlled slewing, setting and tracking," *Publ. Astr. Soc Pacific* **88**, pp. 792–802, 1976.
10. J. Bailey, "Portable telescope control system — DRAMA task specification," PTCS 1.0, Anglo-Australian Observatory and Joint Astronomy Centre, 1997.
11. M. Mountain, "Gemini science requirements," Tech. Rep. 2.0, Gemini Project, 1994.
12. P. Wallace, "Autoguiding and M2 tip/tilt: Astrometric procedures," TCS/PTW 5.7, Rutherford Appleton Laboratory — Gemini Project, 1995.
13. E. Greisen, "Non-linear coordinate systems in AIPS," AIPS Memo 27, National Radio Astronomy Observatory, 1983.
14. P. Wallace, "TPOINT — Telescope pointing analysis system," SUN 100.13, Rutherford Appleton Laboratory — Starlink Project, 1997.
15. J. K. Ousterhout, *Tcl and the Tk Toolkit*, Addison Wesley, Reading, 1994.
16. T. J. Farrell, J. A. Bailey, and K. Shortridge, "Tcl/Tk with DRAMA — a natural for building user interfaces to instrumentation systems?," in *Astronomical Data Analysis Software and Systems IV*, R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds., *A.S.P. Conf Series* **77**, pp. 113–116, 1995.
17. P. Wallace, "Example pointing code," TCS/PTW 4.3, Rutherford Appleton Laboratory — Gemini Project, 1995.