# OpenTSI

## Open Telescope Software Interface

*An open specification of a TPL2 based interface to position a telescope*

Version 1.0

**Document Id:**

4PI-DOC-03-008-03-0

Michael Ruder, Daniel Plasa

Revision 252 as of August 13, 2009 (michael)

**Revision History:**

| | |
|---|---|
| 08.05.2005 | Initial Version (alpha) (dp) |
| 27.09.2005 | Finalized alpha version (dp,mr) |
| 24.10.2005 | Added support for multiple pointing models (mr) |
| 27.10.2005 | Finalized beta version (mr,dp) |
| 01.07.2007 | Finalized version (mr) |

 $Id: OpenTSI.tex 252 2009-08-13 17:53:57Z michael$

# Contents

# 1 Preface

This document describes a high level interface for positioning and controlling of a telescope. For the communication the TPL2 protocol shall be used (refer to [2] for protocol specifications). In contrast to OpenTCI (see [1]) which mainly covers detailed control of the telescope hardware itself, OpenTSI is designed to provide a hardware independent interface to operate a telescope.

# 2 Basics

- The OpenTSI defines a list of modules which will give an ordered, hierarchical hardware independent access to all important high level telescope functions.
- A generic error handling interface is specified to determine the telescope ability to operate.
- Observable objects include sidereal objects, planets, moons, orbital elements and free definable time-position trajectories.
- A wide variety of coordinate systems can be used to both specify observable targets and transform between different systems.
- In addition to these basic modules there may be additional modules covering individual needs and functions. These modules shall be described in separate documents.

# 3 List of Basic Modules

The following modules are specified by the current version of the OpenTSI:

| Name | Function |
|---|---|
| `TELESCOPE` | Status and general setup of the telescope. |
| `OBJECT` | Target object specification in various coordinate systems. |
| `POINTING` | Setup and control of pointing/tracking. |
| `POSITION` | Positions of all axes of the telescope. |
| `AUXILIARY` | Control of auxiliary components of the telescope. |

Every module must have at least the `VERSION` variable. This variable shall give the client information whether the module is useable ($> 0$) on the current telescope. It will provide the client also with an interface version, an interface age and a revision number (packed into one number), giving the client information about the structure of the interface layout:

- The interface version will be increased whenever variables are added or removed or the functionality of variables is changed.
- The interface age allows clients to check if the expected interface version can be used even if the interface version is different from the expected version. The age is computed as maximum supported interface version minus minimum supported version.

  **Example**: interface version is 3 but all variables of version 2 are still in place to stay compatible with old clients. Therefore the age equals $3 - 2 = 1$.
- The interface revision indicates the code revision. In general: the higher, the better, fewer bugs etc.

The three numbers are packed into a single unsigned 32-bit integer with hexadecimal representation `0xIIIAARRR`, `III` is the interface version, `AA` is the age and `RRR` the revision number. A version equal zero means that this module is unavailable.

# 4 Operating the Telescope

- The most important task in operating is of course tracking a desired object. The object specification and setup of desired corrections (e.g. refraction, compensation of mount errors etc.) is done in the `OBJECT` and `POINTING` modules (see sections 6 and 7).
- Setting up the telescope and determining the telescope state is the next important task. During telescope operation, errors can arise from several different sources. One major source of errors is of course the telescope hardware itself (motor failures, broken fuses etc.), other errors may arise during normal operation, e.g. one user sets a target and waits for the telescope to reach that position while another user resets the target and user one's target will never be reached. The first class of errors is normally strictly hardware related, the latter class is hardware independent. This is covered by the `TELESCOPE` module (see section 5).
- Eventually information about telescope axis positions and movements may be needed. This information is provided in the `POSITION` module (see section 8).
- If the telescope has special components like temperature sensors, covers or paddles for manually controlling the telescope, these will be accessible in the `AUXILIARY` module (see section 9).

- In addition, implementations of an OpenTSI telescope control should provide TPL2 events to inform the client about important state changes, e.g. telescope on target, slewing etc. OpenTSI therefore defines a list of error / event numbers, see appendix C.

# 5 The TELESCOPE module

This module contains general telescope settings and the abstract error handling.

| Name | Type | Access | Description |
|---|---|---|---|
| `READY` | INT | RW | Prepare the telescope to become operational. Possible values are:<br>0: power down telescope,<br>1: power up telescope, become operational.<br>Depending on system configuration, mirror covers and a dome will be opened during power up and the telescope will slew to an initial position. Accordingly during power down, the telescope will slew to a park position and mirror covers and a dome will be closed. |
| `READY_STATE` | FLOAT | RO | Possible values are:<br>-3.0: local mode, no remote operation,<br>-2.0: emergency off,<br>-1.0: errors block operation,<br>0.0: shut down,<br>0.1 . . . 0.9: power up/down in progress,<br>1.0: fully operational. |
| `MOTION_STATE` | INT | RO | This is a bit-field, possible values are:<br>0: Telescope is not moving at all,<br>1: one or more axes are moving,<br>2: trajectories are being executed,<br>4: movement is blocked (e.g. by limits),<br>8: movement is in sync with current target, i.e. tracking is accurate,<br>16: movement is smoothed by given jerk/acceleration/velocity limits. |

| Name | Type | Access | Description |
|---|---|---|---|
| STOP | INT | WO | Immediate stop all telescope operations and axis movement. By writing 1 to this variable the telescope hardware will be asked to immediate stop all telescope axes. There is no guarantee that the axes will really be stopped successfully, especially if this command is issued while the telescope system reports errors. |
| CATALOG[] | STRING | RO | List of all supported catalogs. |
| INFO | MODULE | — | General information about the used telescope, see below. |
| CONFIG | MODULE | — | Information about the telescope configuration. |
| STATUS | MODULE | — | Detailed information for the telescope's operational state, see below. |

## 5.1 The TELESCOPE.INFO submodule

This module provides some general information about the telescope, mainly for informational purposes. The telescope name (TELESCOPE.INFO.NAME) can be used to distinguish single telescopes in a group of telescopes.

| Name | Type | Access | Description |
|---|---|---|---|
| NAME | STRING | RO | Telescope name (e.g. "DEEP_SKY NORTH"). |
| DIAMETER | FLOAT | RO | Telescope main mirror diameter in meters. |
| CABINET | STRING | RO | Manufacturer of the control cabinet. |
| MANUFACTURER | STRING | RO | Manufacturer of the telescope hardware. |

## 5.2 The TELESCOPE.CONFIG submodule

This module provides some general configuration information about the telescope.

| Name | Type | Access | Description |
|---|---|---|---|
| CAPABILITIES | INT | RO | A bit mask of features supported by the hardware:<br>1: reverse pointing (ZD $< 0$) is possible.<br>2: extended azimuth range ($> 360°$).<br>4: wrap-around azimuth.<br>8: extended dome range ($> 360°$).<br>16: wrap-around dome.<br>32: dome has azimut rotation (will be available as dome axis 0).<br>64: dome has open/closeable slit (will be available as dome axis 1).<br>128: dome has open/closeable flap (will be available as dome axis 2).<br>256: dome has open/closeable seals (will be available as dome axis 3).<br>512: dome has first moveable enclosure, e.g. simple fold enclosure (will be available as dome axis 1).<br>1024: dome has second moveable enclosure, e.g. two-part roll-roof dome (will be available as dome axis 2). |
| MOUNTOPTIONS | STRING | RO | Possible mount types of the telescope as comma-separated list. Possible values are:<br>"AZ-ZD" = azimuthal telescope mount,<br>"ZD-ZD" = zenith distance-zenith distance telescope mount,<br>"HA-DEC"= equatorial mount. |
| MOUNT | STRING | RW | Select the currently used mount type. Possible values, see above. |
| LOAD | INT | WO | Load last saved configuration from configuration file. The TSI may close the connection after this command, since it will be restarted with the new configuration information. |
| SAVE | INT | WO | Save the currently active configuration to the configuration files. |
| FOCUS_MODEL | MODULE | — | Thermal model for focus corrections. |
| PORT[] | MODULE | — | Information for each optical port of the telescope. |

### 5.2.1 The TELESCOPE.CONFIG.FOCUS_MODEL submodule

A temperature model for the focus can be defined here, which is based on the temperature sensors of the telescope. The formula used to calculate the focus offset can be found in appendix A

| Name | Type | Access | Description |
|---|---|---|---|
| `A0` | FLOAT | RW | General offset. |
| `A1[]` | FLOAT | RW | Linear coefficients. |
| `A2[]` | FLOAT | RW | Quadratic coefficients. |

### 5.2.2 The TELESCOPE.CONFIG.PORT[] submodule

Information on optical ports of the telescope is available here. The index is the OpenTCI index for the optical ports. The `MODEL[]` submodule allows access to the pointing model coefficients of all optical ports and for both normal and reverse orientation. However, for updating models, the `POINTING.MODEL` module should be used, as it also provides functions for adding measurements and doing calculations. A list of all coefficients can be found in section 7.2.

| Name | Type | Access | Description |
|---|---|---|---|
| `AVAILABLE` | INT | RO | Tells if the selected port is available:<br>-1: available, but disabled,<br>0: not available,<br>1: available. |
| `FOCUS_OFFSET` | FLOAT | RW | Relative focus correction for this port [mm]. |
| `DEROTATOR` | INT | RO | Information provided by the telescope hardware:<br>0: no derotator available,<br>1: derotator available,<br>2: derotator with extended range ($> 360°$).<br>4: wrap-around derotator. |
| `FILTER` | INT | RO | Information provided by the telescope hardware:<br>0: no filter available,<br>1 . . . : number of filter slots. |
| `FILTER_NAME[]` | STRING | RW | Symbolic filter names, as configured by the user. |
| `FILTER_OFFSET[]` | FLOAT | RW | Relative focus correction for this filter [mm]. |

| Name | Type | Access | Description |
|---|---|---|---|
| `MODEL[]` | MODULE | — | The mount model for this optical port:<br>0: for normal pointing,<br>1: for reverse pointing |

## 5.3 The TELESCOPE.STATUS submodule

This module covers error handling and telescope status reporting. Considering the two classes of errors mentioned above, the OpenTSI provides two ways of error notification:

1. Errors that occur during telescope operation by user interaction or by wrong user input (as described in the example) can be reported by TPL2 event messages.
2. The special module `TELESCOPE.STATUS` will provide error status for the telescope hardware which is summarized in different functional hardware groups (see below). TPL2 events can be utilized to notify the client that errors have occurred.

To provide an hardware independent abstract error model, the individual hardware errors of the telescope are summarized in several logical function groups. So far the following groups are defined:

| Group | Components |
|---|---|
| `DRIVES` | All axes of the telescope, e.g. AZ, ZD. |
| `SYSTEM` | All global errors, e.g. power failure, reference errors etc. |
| `AUXILIARY` | Other, e.g. sensors. |

The status module contains:

| Name | Type | Access | Description |
|---|---|---|---|
| `GLOBAL` | INT | RO | Global state of the telescope. Possible values are:<br>-2: no license data found,<br>-1: no telescope hardware found (no connection to an OpenTCI compliant telescope),<br>0: operational,<br>$\geq 0$: error bitmask:<br>1: PANIC, a severe condition, completely disabling the entire telescope,<br>2: ERROR, a serious condition, disabling important parts of the telescope system,<br>4: WARNING, a critical condition, which is not (yet) disabling the telescope,<br>8: INFO, a informal situation, which is not affecting the operation. |
| `LIST` | STRING | RO | A comma separated list of function groups that currently have problems. Each entry has the following format:<br><*Group*>\|<*Level*>:<*Component*>\|<*Level*>;...<br>:<*Error*>\|<*Detail*>\|<*Level*>\|<*Component*>;...<br><*Group*> is one of the listed groups above, <*Level*> has the same value as `GLOBAL` above and <*Component*> is for example the axis name (possibly including a index in `[]`). The optional <*Error*> / <*Detail*> information is probably hardware specific and should only be used for logging. At most one entry per group is generated. The <*Level*> fields are an inclusive-or of all underlying errors. If the delimiter should occur within the names or messages, they will be either escaped with a backslash or the entire entry is put in double quotes. |

| Name | Type | Access | Description |
|---|---|---|---|
| CLEAR | INT | WO | When set to same value as returned by GLOBAL, try to clear the telescope errors (otherwise, an error is generated). If the errors reported by the underlying soft- and/or hardware are not resolved, this will not be able to clear the errors. Only resolved errors will be removed from the error list. In some cases, the hardware might do another try when clearing the errors (e.g. retrying to initialize the motor etc.). Therefore, it is not advisable to continuously clear errors and might even lead to hardware damage. |
| ERROR_LIST | STRING | RO | **[Deprecated]** While this function should not be used anymore, it will return the telescope errors as reported by the telescope, e.g. no grouping etc. is done, as it is with the LIST variable. |

# 6 The OBJECT module

This is the module of the OpenTSI which provides functionality for object specification in various coordinate systems. Currently supported object coordinate system/-types are:

| Coordinates | Description |
|---|---|
| INSTRUMENTAL | Instrumental coordinates, typically AZ and ZD. |
| HORIZONTAL | True horizontal coordinates, AZ and ALT or ZD. |
| EQUATORIAL | Equatorial coordinates, RA, DEC with proper motions and in definable epoch/equinox. |
| ELEMENTS | Orbital elements (parabolic, near-parabolic and elliptic). |
| SOLARSYSTEM | Planets and Moons within the solar system. |
| CATALOG | Objects defined in a supported catalog. |

The specification of objects in these coordinate systems will be discussed below. Most coordinates can be given either as single position (e.g. RA, DEC) or as time-position trajectory which allow accurate and time synced arbitrary telescope movements.
In addition to submodules for all of these coordinate systems, the OBJECT module

contains another variable:

| Name | Type | Access | Description |
|---|---|---|---|
| TYPE | STRING | RW | Selected object type, will also be automatically updated, if coordinates are written to one of the submodules. |

## 6.1 The TRAJECTORY submodules

The TRAJECTORY submodule is available in INSTRUMENTAL, HORIZONTAL and EQUATORIAL coordinate systems. Instead of specifying the object only with one positions, an unlimited number of time-position points can be added to a trajectory. For convenience a block of these points can be written into a transfer buffer and be added into the internal trajectory with ADDPOINTS at any time. The provided time should be in the future (or the points will be ignored otherwise). It is also possible to overwrite part of the trajectory by sending points that are within the trajectory's runtime. The buffer structure will be described with the individual coordinate types.

| Name | Type | Access | Description |
|---|---|---|---|
| ADDPOINTS | INT | RW | On write: will add the number of buffer points (starting from BUFFER[0]) into the internal trajectory handling.<br>On read: returns the number of written points that have been written on the last write access. |
| CLEAR | FLOAT | WO | Clear all trajectory points with times that are later or equal the written value. |
| STARTTIME | FLOAT | RO | Time of the first trajectory point (NULL if no trajectory defined). |
| ENDTIME | FLOAT | RO | Time of the last trajectory point (NULL if no trajectory defined). |
| RUNTIME | FLOAT | RO | Remaining runtime of the trajectory (NULL if no trajectory defined). |
| BUFFER[] | MODULE | — | Module array which is used as buffer for new sample points (see below). |

## 6.2 The OBJECT.INSTRUMENTAL submodule

Setting coordinates or adding trajectory points will change OBJECT.TYPE to INSTRUMENTAL.

| Name | Type | Access | Description |
|---|---|---|---|
| AZ | FLOAT | RW | Azimuth. [°] |
| ZD | FLOAT | RW | Zenith distance. Will overwrite ALT. [°] |
| ALT | FLOAT | RW | Altitude/Elevation. Will overwrite ZD. [°] |
| NAME | STRING | RW | Optional name (for information only). |
| TRAJECTORY | MODULE | — | Trajectory module. |

Structure of Trajectory buffer is:

| Name | Type | Access | Description |
|---|---|---|---|
| UT1 | FLOAT | RW | UT1. [unix time, seconds since 01.01.1970 00:00:00] |
| AZ | FLOAT | RW | Azimuth. [°] |
| ZD | FLOAT | RW | Zenith distance. [°] |
| DEROTATOR | FLOAT | RW | Derotator (depending on POSITION.SETUP.DEROTATOR.SYNCMODE considered absolute or as offset). [°] |

## 6.3 The OBJECT.HORIZONTAL submodule

Setting coordinates or adding trajectory points will change OBJECT.TYPE to HORIZONTAL.

| Name | Type | Access | Description |
|---|---|---|---|
| AZ | FLOAT | RW | Azimuth. [°] |
| ALT | FLOAT | RW | Altitude/Elevation. Will overwrite ZD. [°] |
| ZD | FLOAT | RW | Zenith distance. Will overwrite ALT. [°] |
| NAME | STRING | RW | Optional name (for information only). |
| TRAJECTORY | MODULE | — | Trajectory module. |

Structure of Trajectory buffer is:

| Name | Type | Access | Description |
|---|---|---|---|
| UT1 | FLOAT | RW | UT1. [unix time, seconds since 01.01.1970 00:00:00] |
| AZ | FLOAT | RW | Azimuth. [°] |
| ALT | FLOAT | RW | Altitude. [°] |
| DEROTATOR | FLOAT | RW | Derotator (depending on POSITION.SETUP.DEROTATOR.SYNCMODE considered absolute or as offset). [°] |

## 6.4 The OBJECT.EQUATORIAL submodule

Setting coordinates or adding trajectory points will change `OBJECT.TYPE` to `EQUATORIAL`.

| Name | Type | Access | Description |
|---|---|---|---|
| `EPOCH` | FLOAT | RW | Epoch of the coordinates. [julian year] |
| `EQUINOX` | FLOAT | RW | Equinox of the coordinates. [julian year] |
| `RA` | FLOAT | RW | Right ascension. [h] |
| `DEC` | FLOAT | RW | Declination. [°] |
| `RA_PM` | FLOAT | RW | Proper motion in right ascension. [h/year] |
| `DEC_PM` | FLOAT | RW | Proper motion in declination. [°/year] |
| `RA_RATE` | FLOAT | RW | Additional rate in right ascension (for non-stellar objects). This rate is applied to `RA` when the tracking is started. Therefore, `RA` has to be the position of the object at start of tracking. [h/sec] |
| `DEC_RATE` | FLOAT | RW | Additional rate in declination (for non-stellar objects). This rate is applied to `DEC` when the tracking is started. Therefore, `DEC` has to be the position of the object at start of tracking. [°/sec] |
| `NAME` | STRING | RW | Optional name (for information only). |
| `TRAJECTORY` | MODULE | — | Trajectory module. |

Structure of Trajectory buffer is:

| Name | Type | Access | Description |
|---|---|---|---|
| `UT1` | FLOAT | RW | UT1. [unix time, seconds since 01.01.1970 00:00:00] |
| `RA` | FLOAT | RW | Right ascension. [h] |
| `DEC` | FLOAT | RW | Declination. [°] |
| `DEROTATOR` | FLOAT | RW | Derotator (depending on `POSITION.SETUP.DEROTATOR.SYNCMODE` considered absolute or as offset). [°] |

The epoch/equinox (`EPOCH`/`EQUINOX`), proper motion (`RA_PM`/`DEC_PM`) and rate (`RA_-RATE`/`DEC_RATE`) settings will be considered, but cannot be specified for each trajectory point.

## 6.5 The OBJECT.ELEMENTS submodule

Setting orbital elements parameters will change `OBJECT.TYPE` to `ELEMENTS`.

| Name | Type | Access | Description |
|---|---|---|---|
| `TYPE` | FLOAT | RW | Type of orbital elements:<br>0: elliptic,<br>1: parabolic,<br>2: near-parabolic. |
| `EQUINOX` | FLOAT | RW | Equinox of the coefficients. [julian year] |
| `T` | FLOAT | RW | All types: Time of passage through perihelion. [julian year] |
| `I` | FLOAT | RW | All types: Inclination. [°] |
| `OMEGA_PERIHELION` | FLOAT | RW | All types: Argument of perihelion. [°] |
| `OMEGA_NODE` | FLOAT | RW | All types: Longitude of ascending node. [°] |
| `A` | FLOAT | RW | Elliptic: semimajor axis. [AU] |
| `E` | FLOAT | RW | Elliptic/near-parabolic: Eccentricity. |
| `Q` | FLOAT | RW | Parabolic/near-parabolic: Perihelion distance. [AU] |
| `NAME` | STRING | RW | Optional name (for information only). |
| `DISTANCE` | FLOAT | RO | Object distance (for information only). [AU] |

## 6.6 The OBJECT.SOLARSYSTEM submodule

Selecting solar system objects will change `OBJECT.TYPE` to `SOLARSYSTEM`.

| Name | Type | Access | Description |
|---|---|---|---|
| `OBJECT` | INT | RW | Number of object. Possible Values:<br>0: Sun, 1: Mercury, 2: Venus, 3: Earth, 4: Mars, 5: Jupiter, 6: Saturn, 7: Neptune, 8: Uranus, 9: Pluto. |

| Name | Type | Access | Description |
|---|---|---|---|
| `MOON` | INT | RW | Moon of the object (only Earth, Jupiter and Saturn):<br>0: Object itself<br>Earth: 1: Moon<br>Jupiter: 1: Io, 2: Europa, 3: Ganymed, 4: Callisto<br>Saturn: 1: Mimas, 2: Enceladus, 3: Tethys, 4: Dione, 5: Rhea, 6: Titan, 7: Hyperion, 8: Iapetus |
| `NAME` | STRING | RO | Name of that object (for information only). |
| `DISTANCE` | FLOAT | RO | Object distance (for information only). [AU] |

Tracking or pointing to the sun may be disabled for security reasons.

## 6.7 The OBJECT.CATALOG submodule

Selecting a catalog object will change `OBJECT.TYPE` depending on the catalog entry to most likely `EQUATORIAL` or `ELEMENTS`.

| Name | Type | Access | Description |
|---|---|---|---|
| `NAME` | STRING | RW | Name of catalog (see above). |
| `OBJECT` | STRING | RW | Name of object (e.g. “M51”). |

# 7 The POINTING module

This module allows acquisition of an astronomical object (which has been selected using the `OBJECT` module) and also provides options to configure the way the telescope will acquire the object. The following variables are available:

| Name | Type | Access | Description |
|---|---|---|---|
| TRACK | INT | RW | Start or stop tracking on a target. Possible values are:<br>0: stop tracking.<br>1: (re)start tracking (this is also needed, if a new object has been selected).<br>2: goto object coordinates. The telescope will move to the current object coordinates and stay there. This will end a running a tracking.<br>3: update filter wheel position only. This will not affect a running tracking.<br>4: update focus position only. This will not affect a running tracking.<br>5: update derotator position only. This is only possible, if POINTING.SETUP.DEROTATOR.SYNCMODE was 0 at begin of tracking or no tracking is active.<br>6: update dome position only. This is only possible, if POINTING.SETUP.DOME.SYNCMODE was 0 at begin of tracking or no tracking is active. |
| ORIENTATION | INT | RO | The used tracking orientation:<br>0: normal pointing (ZD $> 0$)<br>1: reverse pointing (ZD $< 0$)<br>If reverse tracking is available depends on the hardware (see above). Also, the POINTING.SETUP module allows to activate or deactivate this feature. |
| DEROTATOR_OFFSET | FLOAT | RO | The offset for the derotator (in relative mode) or the derotator position (in absolute mode). This will include both user settings and automatic settings (if this optimization is selected (see below)). [°] |
| FOCUS_OFFSET | FLOAT | RO | The automatically selected offset for the focus. [mm] |
| SLEWTIME | FLOAT | RO | Time needed to reach the target once tracking is started. [s] |

| Name | Type | Access | Description |
|---|---|---|---|
| TRACKTIME | FLOAT | RO | Time available for tracking the target before mount limits will be reached. [s] |
| TRACKLIMITS | STRING | RO | Comma separated list of limits that will be reached eventually:<br>AXIS_PosMin: Axis will hit minimum position limit,<br>AXIS_PosMax: Axis will hit maximum position limit,<br>AXIS_SpeedMax: Axis will go over maximum speed limit,<br>TRAJECTORY_EndOfData: User-defined trajectory data will end.<br>OBJECT_BelowHorizon: Object will go below horizon.<br>OBJECT_Invisible: Object is invisible at current geographic location.<br>For AXIS, the same names as in POSITION.INSTRUMENTAL will be used. |
| TARGETDISTANCE | FLOAT | RO | The RMS distance of all axes from the target. [°] |
| SETUP | MODULE | — | Configuration of the tracking/slewing mode. |
| MODEL | MODULE | — | Configuration of the mount model (correction for mechanical errors and errors due to the optical alignment). |

## 7.1 The POINTING.SETUP submodule

In the SETUP module, you can configure how the target should be acquired. Most of the settings will however be ignored if OBJECT.TYPE is INSTRUMENTAL.

| Name | Type | Access | Description |
|---|---|---|---|
| REFRACTION | INT | RW | Consider atmospheric refraction. Possible values are:<br>0: no refraction correction,<br>1: use refraction correction. The temperature and pressure will be taken from the ENVIRONMENT module. |

| Name | Type | Access | Description |
|---|---|---|---|
| ORIENTATION | INT | RW | Desired orientation for tracking:<br>0: normal pointing (ZD > 0)<br>1: reverse pointing (ZD < 0)<br>2: allow automatic selection |
| OPTIMIZATION | INT | RW | Optimization mode for slewing to/tracking of objects:<br>0: no optimization. An extended azimuth/derotator range will not be used.<br>1: maximize track time. Extended ranges of azimuth/derotator and, if selected, automatic orientation selection will be used to allow the maximum possible track time of the selected object.<br>2: minimize slew time. As above, all features will be used, to allow fastest acquisition of the target from the current telescope position.<br>The optimization will also consider MIN_TRACKTIME (see below). |
| MIN_TRACKTIME | FLOAT | RW | The minimal required track time. This will be considered, if OPTIMIZATION is used. If the selected target cannot be tracked for the given time, the call to POINTING.TRACK will fail. [s] |
| USE_PORT | INT | RW | Optical port, specifies used derotator and filter and will also rotate M3 (if supported by the telescope hardware). |
| LOCAL | MODULE | — | Location and time settings (see below). |
| ENVIRONMENT | MODULE | — | Environmental conditions setup (see below). |
| FOCUS | MODULE | — | Focus specific options (see below). |
| DEROTATOR | MODULE | — | Derotator specific options (see below). |
| FILTER | MODULE | — | Filter wheel specific options (see below). |
| DOME | MODULE | — | Dome specific options (see below). |

### 7.1.1 The POINTING.SETUP.LOCAL submodule

Location and time settings can be set here.

| Name | Type | Access | Description |
|---|---|---|---|
| SYNCMODE | INT | RW | Sync telescope location with information provided by the telescope hardware:<br>0: don't sync, use custom values (see below),<br>1: sync, use the values from the telescope<br>Writing to any of the variables below will set this variable to 0. |
| LATITUDE | FLOAT | RW | Telescope's latitude. Positive for northern hemisphere. [°] |
| LONGITUDE | FLOAT | RW | Telescope's longitude. Positive east of Greenwich. [°] |
| HEIGHT | FLOAT | RW | Telescope's height. [m] |
| UT1-UTC | FLOAT | RW | Difference between UT1 and UTC. [s] |
| TAI-UTC | FLOAT | RW | Difference between TAI and UTC. [s] |

### 7.1.2 The POINTING.SETUP.ENVIRONMENT submodule

Weather status information or custom values.

| Name | Type | Access | Description |
|---|---|---|---|
| SYNCMODE | INT | RW | Synchronize with environment monitor:<br>0: don't sync, use custom values,<br>1: sync. |
| TEMPERATURE | FLOAT | RW | Temperature. [°C] |
| PRESSURE | FLOAT | RW | Air pressure. [mBar] |

### 7.1.3 The POINTING.SETUP.FOCUS submodule

Allows the user to setup the focus settings.

| Name | Type | Access | Description |
|---|---|---|---|
| SYNCMODE | INT | RW | Sync focus movement to various conditions. This is a bitmap, the following bits are defined:<br>0: don't sync,<br>1: sync with focus position.<br>2: sync with thermal model,<br>4: sync with port offsets,<br>8: sync with filter offsets,<br>16: turn off focus motor during tracking. |

| Name | Type | Access | Description |
|---|---|---|---|
| `POSITION` | FLOAT | RW | Focus position. [mm] |

### 7.1.4 The POINTING.SETUP.DEROTATOR submodule

Allows the user to setup the derotator (if available on the used port).

| Name | Type | Access | Description |
|---|---|---|---|
| `SYNCMODE` | INT | RW | Sync derotator movement. Possible values:<br>0: don't sync,<br>1: absolute position, given as `OFFSET` (or `DEROTATOR` for trajectories),<br>2: true orientation,<br>3: as 2, but use `OFFSET` (or `DEROTATOR` for trajectories) as an additional offset,<br>4: true orientation plus a 0, 90, 180 or 270 degree offset (to allow longer tracktimes or short slewtimes),<br>5: as 4, but use `OFFSET` (or `DEROTATOR` for trajectories) as an additional offset,<br>6: true orientation plus arbitrary offset (for even better optimization). |
| `OFFSET` | FLOAT | RW | Derotator position (`SYNCMODE` 1) or position offset (`SYNCMODE` 3 and 5). [°] |

### 7.1.5 The POINTING.SETUP.FILTER submodule

Allows the user to select the used filter (if available on the used port).

| Name | Type | Access | Description |
|---|---|---|---|
| `INDEX` | INT | RW | Allow selection of filter by index. |
| `NAME` | STRING | RW | Allow selection of filter by name. |

### 7.1.6 The POINTING.SETUP.DOME submodule

Dome related setup.

| Name | Type | Access | Description |
|---|---|---|---|
| SYNCMODE | INT | RW | Synchronize dome movements with telescope. Possible values:<br>0: don't sync,<br>1: fixed position, given as OFFSET,<br>2: sync discrete (allow a position deviation from the azimuth axis according to MAX_DEVIATION),<br>3: as 2, but use OFFSET as an additional offset,<br>4: sync continuous (may not be suited for all domes),<br>5: as 4, but use OFFSET as an additional offset. |
| MAX_DEVIATION | FLOAT | RW | For discrete movement (SYNCMODE 2 and 3), the maximum position deviation that the dome can have from the telescope azimuth. If set to NULL, an automatically calculated value is used, which takes dome/-mount parameters (diameter, slit width, ...) into account. |
| OFFSET | FLOAT | RW | Dome position (SYNCMODE 1) or position offset (SYNCMODE 3 and 5). [°] |

## 7.2 The POINTING.MODEL submodule

This module holds pointing model coefficients for the currently used optical port and also provides functionality to build a new model from taken measurements. There are several model types available, which will be discussed later on.
The coefficients of the models can be calculated internally or by using additional programs (e.g. TPoint). The formulas of the models can be found in appendix B.
The following variables are available:

| Name | Type | Access | Description |
|---|---|---|---|
| TYPE | INT | RW | Select pointing model type:<br>0: none,<br>1: classic,<br>2: extended. |

| Name | Type | Access | Description |
|---|---|---|---|
| ADD | STRING | WO | Write here to take a measurement, which is named "value". If an empty string is specified and the currently pointed object has a name, it will be used instead. |
| CALCULATE | FLOAT | RW | Calculate a model and fill all parameters with the new model on write. Possible values are:<br>1: calculate,<br>2: calculate and reset any axis OFFSET to 0.<br>Returns fit quality (mean error) on read, if calculation was successful. [°] |
| CALCULATE_DETAIL | STRING | RO | After a successful calculation, returns a list of errors for each individual measurement in the format:<br>number,az_error,zd_error,derotator_error,dome_error,total_error;...;<br>-1;total_az_error,total_zd_error,total_derotator_error,total_dome_error [°] |
| LIST | STRING | RO | List of all measurements in the format:<br>number,name,az,offset_az,<br>zd,offset_zd,<br>derotator,offset_derotator,<br>dome,offset_dome;number,... |
| COUNT | INT | RO | Number of measurements taken so far. |
| CLEAR | INT | WO | Will remove all measurements from list. |
| REMOVE | INT | WO | Remove $n$-th measurement from list. For $n < 0$, use one of the last measurements (-1: last, -2: second to last, ...). |
| FILE | STRING | RW | Filename of the measurement file. |
| LOAD | INT | WO | Load measurements from file:<br>1: load and overwrite current list,<br>2: load and append to current list. |
| SAVE | INT | WO | Save measurements to file:<br>1: save and overwrite file,<br>2: save and append to file. |

| Name | Type | Access | Description |
|---|---|---|---|
| AUTO_SAVE | INT | RW | Automatically save measurements to file:<br>0: disabled,<br>1: enabled. |
| CLASSIC | MODULE | — | Coefficients for the classic model. |
| EXTENDED | MODULE | — | Coefficients for the extended model. |

### 7.2.1 The POINTING.MODEL.CLASSIC submodule

This is the classic model which is known from the literature. It consists of offsets for all telescope axes (AOFF/ZOFF/DOFF) and additionally of geometric corrections, which take into account, that the telescope might not be well leveled (AN/AE), the zenith distance axis is not perpendicular to the azimuth axis (NPAE), that the optical axis is not perpendicular to the zenith distance axis (BNP) and that the tube might be bent, depending on the zenith distance (TF).

| Name | Type | Access | Description |
|---|---|---|---|
| AOFF | FLOAT | RW | Azimuth offset. [°] |
| ZOFF | FLOAT | RW | Zenith distance offset. [°] |
| DOFF | FLOAT | RW | Derotator offset. [°] |
| AN | FLOAT | RW | Tilt of azimuth axis toward north. [°] |
| AE | FLOAT | RW | Tilt of azimuth axis toward east. [°] |
| NPAE | FLOAT | RW | Error in perpendicularity of azimuth and zenith distance axis. [°] |
| BNP | FLOAT | RW | Error in perpendicularity of optical and zenith distance axis. [°] |
| TF | FLOAT | RW | Sagging of tube. [°] |

### 7.2.2 The POINTING.MODEL.EXTENDED submodule

In comparison to the classic model, this pointing model has been extended by four coefficients which take into account that the encoders are not mounted precisely in the center of the axis rotation (AES/AEC for azimuth, ZES/ZEC for zenith distance) as well as by some phenomenological coefficients for higher order effects (AS2A, AC2A, AS3A, AC3A, ZS2A, ZC2A, ZS3A,ZC3A, ZS4A, ZC4A, C5). Additionally, the coefficients for azimuth tilt correction are now separate for azimuth and altitude (AAN, ZAN, AAE, ZAE).

| Name | Type | Access | Description |
|---|---|---|---|
| AOFF | FLOAT | RW | Azimuth offset. [°] |
| ZOFF | FLOAT | RW | Zenith distance offset. [°] |

| Name | Type | Access | Description |
|---|---|---|---|
| `DOFF` | FLOAT | RW | Derotator offset. [°] |
| `COFF` | FLOAT | RW | Astrodome offset. [°] |
| `AAN` | FLOAT | RW | Tilt of azimuth axis toward north (correction for azimuth axis). [°] |
| `ZAN` | FLOAT | RW | Tilt of azimuth axis toward north (correction for zenith distance axis). [°] |
| `AAE` | FLOAT | RW | Tilt of azimuth axis toward east (correction for azimuth axis). [°] |
| `ZAE` | FLOAT | RW | Tilt of azimuth axis toward east (correction for zenith distance axis). [°] |
| `NPAE` | FLOAT | RW | Error in perpendicularity of azimuth and zenith distance axis. [°] |
| `BNP` | FLOAT | RW | Error in perpendicularity of optical and zenith distance axis. [°] |
| `AES` | FLOAT | RW | Eccentricity of azimuth encoder, sine part. [°] |
| `AEC` | FLOAT | RW | Eccentricity of azimuth encoder, cosine part. [°] |
| `ZES` | FLOAT | RW | Eccentricity of zenith distance encoder, sine part. [°] |
| `ZEC` | FLOAT | RW | Eccentricity of zenith distance encoder, cosine part (this coefficient also includes the sagging of the tube). [°] |
| `AS2A` | FLOAT | RW | Phenomenological correction of azimuth due to double azimuth argument, sine part. [°] |
| `AC2A` | FLOAT | RW | Phenomenological correction of azimuth due to double azimuth argument, cosine part. [°] |
| `AS3A` | FLOAT | RW | Phenomenological correction of azimuth due to triple azimuth argument, sine part. [°] |
| `AC3A` | FLOAT | RW | Phenomenological correction of azimuth due to triple azimuth argument, cosine part. [°] |
| `ZS2A` | FLOAT | RW | Phenomenological correction of zenith distance due to double azimuth argument, sine part. [°] |

| Name | Type | Access | Description |
|---|---|---|---|
| ZC2A | FLOAT | RW | Phenomenological correction of zenith distance due to double azimuth argument, cosine part. [°] |
| ZS3A | FLOAT | RW | Phenomenological correction of zenith distance due to triple azimuth argument, sine part. [°] |
| ZC3A | FLOAT | RW | Phenomenological correction of zenith distance due to triple azimuth argument, cosine part. [°] |
| ZS4A | FLOAT | RW | Phenomenological correction of zenith distance due to quadruple azimuth argument, sine part. [°] |
| ZC4A | FLOAT | RW | Phenomenological correction of zenith distance due to quadruple azimuth argument, cosine part. [°] |
| C5 | FLOAT | RW | Phenomenological correction of zenith distance due to inverse sine of zenith distance. [°] |

# 8 The POSITION module

This module allows access to real-time positions and state of all axes and other telescope components.

| Name | Type | Access | Description |
|---|---|---|---|
| LOCAL | MODULE | — | Time and position information. |
| INSTRUMENTAL | MODULE | — | Instrumental coordinates. |
| HORIZONTAL | MODULE | — | Horizontal coordinates. |
| EQUATORIAL | MODULE | — | Equatorial coordinates. |

## 8.1 The POSITION.LOCAL submodule

This is the position and time information as provided by the telescope hardware.

| Name | Type | Access | Description |
|---|---|---|---|
| SIDEREAL_TIME | FLOAT | RO | Current local sidereal time. [h] |
| UTC | FLOAT | RO | Current telescope time in UTC. [seconds since 01.01.1970 00:00:00] |

| Name | Type | Access | Description |
|---|---|---|---|
| UT1 | FLOAT | RO | Current telescope time in UT1. [seconds since 01.01.1970 00:00:00] |
| TAI | FLOAT | RO | Current telescope time in TAI. [seconds since 01.01.1970 00:00:00] |
| UT1-UTC | FLOAT | RO | Difference between UT1 and UTC. [s] |
| TAI-UTC | FLOAT | RO | Difference between TAI and UTC. [s] |
| LATITUDE | FLOAT | RO | Telescope's latitude. Positive for northern hemisphere. [°] |
| LONGITUDE | FLOAT | RO | Telescope's longitude. Positive east of Greenwich. [°] |
| HEIGHT | FLOAT | RO | Telescope's height. [m] |

## 8.2 The POSITION.INSTRUMENTAL submodule

The following telescope axes will be available. Non native axes (which are not provided by the telescope) may be simulated by the software which is however not mandatory:

| Axis | Description |
|---|---|
| AZ | Azimuth. |
| ZD | Zenith distance. |
| ALT | Elevation/Altitude. |
| RA | Right ascension. |
| DEC | Declination. |
| FOCUS | Focus. |
| DEROTATOR[] | Derotators. |
| FILTER[] | Filter wheels. |
| DOME[] | Astrodome axes. |

For all these axes, the following information is provided:

| Name | Type | Access | Description |
|---|---|---|---|
| POWER_STATE | FLOAT | RO | Power state of the component. Possible values are:<br>-1.0: emergency off,<br>0.0: off,<br>0.5: almost on,<br>1.0: on. |

| Name | Type | Access | Description |
|---|---|---|---|
| REFERENCED | FLOAT | RO | Referenced state of this component. Possible values are:<br>0.0: not referenced,<br>1.0: referenced,<br>0.0 . . . 1.0 values will indicate the progress during referencing. |
| ERROR_STATE | INT | RO | If this variable is $\neq 0$, then the axis has currently some error. |
| LIMIT_STATE | INT | RO | If this variable is $\neq 0$, then the axis is currently at some limit. |
| MOTION_STATE | INT | RO | If this variable is $\neq 0$, then the axis is currently moving (see TELESCOPE.MOTION_-STATE). |
| REALPOS | FLOAT | RO | True current position in component specific units. See [1] for an explanation of coordinate systems. |
| CURRPOS | FLOAT | RO | The current position in degrees, possibly corrected by the given OFFSET (in the same unit as REALPOS). |
| CURRSPEED | FLOAT | RO | The current speed of the axis in units/s. |
| TARGETPOS | FLOAT | RW | The position the axis is currently moving to (in the same unit as REALPOS), possibly corrected by the current OFFSET. |
| OFFSET | FLOAT | RW | Additional offset (in the same unit as REALPOS) that will be added to *all* positioning requests also during tracking. This offset will be used for pointing model measurements. |
| TARGETDISTANCE | FLOAT | RO | The distance between CURRPOS and TARGETPOS. |

DEROTATOR and FILTER are arrays. The index is used to select the derotator / filter wheel of a specific optical port. For an enumeration of the optical ports, please refer to [?].

DOME is also an array, as some domes offer more than one axis (azimuth rotation and slit open/close etc.). TELESCOPE.CONFIG.CAPABILITIES can be used to find out which axes the dome offers and at what indices they reside. Further details can be found in [?].

## 8.3 The POSITION.HORIZONTAL submodule

Provides the telescope position in true horizontal coordinates (not corrected for mount errors or refraction).

| Name | Type | Access | Description |
|---|---|---|---|
| AZ | FLOAT | RO | Azimuth in true horizontal coordinates. [°] |
| ALT | FLOAT | RO | Elevation/Altitude in true horizontal coordinates. [°] |
| ZD | FLOAT | RO | Zenith distance in true horizontal coordinates. [°] |
| DEROTATOR | FLOAT | RO | Derotator position. [°] |
| DOME | FLOAT | RO | Dome position. [°] |

## 8.4 The POSITION.EQUATORIAL submodule

Provides the telescope position in equatorial coordinates.

| Name | Type | Access | Description |
|---|---|---|---|
| RA_J2000 | FLOAT | RO | Current right ascension of telescope in J2000.0 reference system. [h] |
| DEC_J2000 | FLOAT | RO | Current declination of telescope in J2000.0 reference system. [°] |
| RA_CURRENT | FLOAT | RO | Current right ascension of telescope in current equinox. [h] |
| DEC_CURRENT | FLOAT | RO | Current declination of telescope in current equinox. [h] |
| PARALLACTIC_ANGLE | FLOAT | RO | Current parallactic angle. [°] |
| POSITION_ANGLE | FLOAT | RO | Current position angle. [°] |

# 9 The AUXILIARY module

This module holds information of the auxiliary telescope systems (e.g. COVERS, PADDLE, SENSORS).

## 9.1 The AUXILIARY.COVER submodule

This module (or, optionally, module array) provides functions to move covers, e.g. mirror covers.

| Name | Type | Access | Description |
|---|---|---|---|
| REALPOS | FLOAT | RO | Position of the cover:<br>0: closed,<br>1: open. |
| TARGETPOS | FLOAT | RW | Target position of the cover as above. |

## 9.2 The AUXILIARY.DOME submodule

This module provides functions to open/close the dome (rotating it is done with POSITION.INSTRUMENTAL.DOME or POINTING.SETUP.DOME). Currently, only completely opening or closing is supported. In case of a dome with slit and flap or two-part roll-roofs, finer control may be desired. Please use POSITION.INSTRUMENTAL.DOME[x].

| Name | Type | Access | Description |
|---|---|---|---|
| REALPOS | FLOAT | RO | Position of the dome enclosure(s):<br>0: closed,<br>1: open. |
| TARGETPOS | FLOAT | RW | Target position of the dome enclosure(s) as above. |

## 9.3 The AUXILARY.PADDLE submodule

| Name | Type | Access | Description |
|---|---|---|---|
| ACTIVE | INT | RW | Manual control activation state. Possible values:<br>0: disabled,<br>1: enabled. |
| BRIGHTNESS | FLOAT | RW | Brightness of buttons. Possible values range from:<br>0.0 . . . 1.0 (lowest . . . brightest setting). |
| SPEED | FLOAT | RO | Speed setting on the manual control device. |
| MODE | INT | RO | Mode of the manual control (e.g. key switch on the paddle). Possible values are:<br>0: disabled,<br>1: enabled.<br>Note that the actual state of the manual control is an "AND" conjunction of MODE and ACTIVE. |

| Name | Type | Access | Description |
|---|---|---|---|
| `SELECTION` | INT | RO | A number representing a selection (e.g. selected device) that was made with the manual control. |
| `SELECTION_TEXT` | STRING | RO | A textual description of the selected device/speed. |

## 9.4 The AUXILIARY.SENSOR[] submodule

This module array provides functions to read out sensors of the telescope.

| Name | Type | Access | Description |
|---|---|---|---|
| `DESCRIPTION` | STRING | RO | A textual description of the sensor position. |
| `UNIT` | STRING | RO | The unit of the sensor device like Pa, °C etc. |
| `VALUE` | FLOAT | RO | Sensor value in the specified unit. |

## 9.5 The AUXILIARY.SWITCH[] submodule

This module array provides functions to read and write to switches of the telescope.

| Name | Type | Access | Description |
|---|---|---|---|
| `DESCRIPTION` | STRING | RO | A textual description of the switch position. |
| `UNIT` | STRING | RO | The unit of the switch device like Pa, °C etc. |
| `VALUE` | FLOAT | RW | Switch value in the specified unit. |

# A Formula for the simple thermal focus correction

The TSI includes a simple thermal focus correction which is based on the temperature sensors of the telescope. All $N$ temperature sensors will be assigned a consecutive index number, starting at 0. The order of the sensors is based on the TCI ordering of temperature sensors (see [1]).
Then the temperature dependent focus offset will be calculated using the following formula:

$$\varDelta_{\mathrm{Foc}} = \sum_{i=0}^{N-1} \left( a_{2_i} T_i^2 + a_{1_i} T_i \right) + a_0$$

# B Formulas of the different Pointing Models

The formulas in the following sections refer to the true instrumental axes of the telescope hardware. Depending on the quantity of measurements, the calculation of the model may only determine some of the coefficients.

## B.1 Azimuthal mounted telescopes

### B.1.1 The Classic Model

The classic model takes several geometric errors of the telescopes into account. An error in the leveling of the telescope toward north (AN) and toward east (AE), a non-perpendicularity of the azimuth and zenith distance axis (NPAE), a non-perpendicularity of the optical and the zenith distance axis (BNP) and a sagging of the tube (TF).

$$
\begin{aligned}
\Delta_{\mathrm{Az}} &= -c_{\mathrm{AN}} \cdot \sin A \cdot \cot Z \\
&+ c_{\mathrm{AE}} \cdot \cos A \cdot \cot Z \\
&+ c_{\mathrm{NPAE}} \cdot \cot Z \\
&- c_{\mathrm{BNP}} \cdot \frac{1}{\sin Z} \\
&+ c_{\mathrm{AOFF}} \\
\Delta_{\mathrm{ZD}} &= c_{\mathrm{AN}} \cdot \cos A \\
&+ c_{\mathrm{AE}} \cdot \sin A \\
&+ c_{\mathrm{TF}} \cdot \sin Z \\
&+ c_{\mathrm{ZOFF}} \\
\Delta_{\mathrm{Der}} &= c_{\mathrm{DOFF}}
\end{aligned}
$$

### B.1.2 The Extended Model

The extended model includes, in addition to the classic model, four coefficients that take the eccentricity of the encoders into account (AES, AEC, ZES, ZEC). The constants TF of the classic model is included in EEC, so there are only three new coefficients. Furthermore it adds several phenomenological coefficients (AS2A, AC2A, AS3A, AC3A, ZS2A, ZC2A, ZS3A, ZC3A, ZS4A, ZC4A, C5) for higher order effects of unknown origin. Additionally, the coefficients for azimuth tilt correction are now separate for azimuth and altitude (AAN, AAE, ZAN, ZAE).

$$
\begin{aligned}
\Delta_{\mathrm{Az}} &= c_{\mathrm{AAN}} \cdot \sin A \cdot \cot Z \\
&- c_{\mathrm{AAE}} \cdot \cos A \cdot \cot Z \\
&+ c_{\mathrm{NPAE}} \cdot \cot Z \\
&- c_{\mathrm{BNP}} \cdot \frac{1}{\sin Z} \\
&+ c_{\mathrm{AES}} \cdot \sin A \\
&+ c_{\mathrm{AEC}} \cdot \cos A \\
&+ c_{\mathrm{AS2A}} \cdot \sin(2A) \cdot \cot Z \\
&+ c_{\mathrm{AC2A}} \cdot \cos(2A) \cdot \cot Z \\
&+ c_{\mathrm{AS3A}} \cdot \sin(3A) \cdot \cot Z \\
&+ c_{\mathrm{AC3A}} \cdot \cos(3A) \cdot \cot Z \\
&+ c_{\mathrm{AOFF}} \\
\Delta_{\mathrm{ZD}} &= c_{\mathrm{ZAN}} \cdot \cos A \\
&+ c_{\mathrm{ZAE}} \cdot \sin A \\
&+ c_{\mathrm{ZES}} \cdot \sin Z \\
&+ c_{\mathrm{ZEC}} \cdot \cos Z \\
&+ c_{\mathrm{ZS2A}} \cdot \sin(2A) \\
&+ c_{\mathrm{ZC2A}} \cdot \cos(2A) \\
&+ c_{\mathrm{ZS3A}} \cdot \sin(3A) \\
&+ c_{\mathrm{ZC3A}} \cdot \cos(3A) \\
&+ c_{\mathrm{ZS4A}} \cdot \sin(4A) \\
&+ c_{\mathrm{ZC4A}} \cdot \cos(4A) \\
&+ c_{\mathrm{C5}} \cdot \frac{1}{\sin Z} \\
&+ c_{\mathrm{ZOFF}} \\
\Delta_{\mathrm{Der}} &= c_{\mathrm{DOFF}} \\
\Delta_{\mathrm{Dome}} &= c_{\mathrm{COFF}}
\end{aligned}
$$

# C Events

This paragraph will be completed in near future.

# References

[1] M. Ruder and D. Plasa. *OpenTCI, Open Telescope Control Interface — An open specification of a TPL2 based interface to control a telescope*. 4pi systeme GmbH. `4PI-DOC-03-008-02`.

[2] M. Ruder and D. Plasa. *TPL2, Transfer Protocol Language, V2 — A protocol for client-server based exchange of data and commands over a TCP/IP network connection.* 4pi systeme GmbH. `4PI-DOC-03-008-01`.