OpenTSI
Open Telescope Software Interface

An open specification of a OpenTPL based interface to position a telescope

Interface specification

Version 2.0

Document ID:

`tclm-tsi:spec-en`

Revision 53 as of July 19, 2019 15:26:28

tau-tec

Revision history:

Date	Version	Changes
08.05.2005	0.0	Initial Version (alpha) (dp)
27.09.2005	0.1	Finalized alpha version (dp,mr)
24.10.2005	0.2	Added support for multiple pointing models (mr)
27.10.2005	0.5	Finalized beta version (mr,dp)
01.07.2007	1.0	Finalized version (mr)
10.08.2012	1.1	Addition of port covers, instrument selectors and basic hexapod/active optics support, updated layout (mr,ms)
29.08.2012	1.5	Extensions for non-horizontal telescope mounts, pointing model handling, air mass, refraction (mr,ms)
20.06.2013		Added additional pointing model definitions for non-horizontal telescopes (mr)
in progress	2.0	Additional models (dome, AO, hexapod), siderostat mode, extended pointing model measurement creation, access to currently tracked object, connection-local object/pointing setup, access to object trajectories (pointing and currently tracked), individual configuration saving (mr,ms)

OpenTSI — Open Telescope Software Interface (An open specification of a OpenTPL based interface to position a telescope)
Version 2.0
Document ID: `tclm-tsi:spec-en`
Revision 53 as of July 19, 2019 15:26:28 by Michael Ruder

`tclm-tsi_spec-en.lyx 53 2019-07-19 15:26:28Z ruder`

Contents

1. Preface

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

2. Basics

2.1. Overview

- The OpenTSI defines a list of modules which will give an ordered, hierarchical and 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 convert between different systems.
- In addition to these basic modules there may be additional modules covering individual needs and functions. These modules will be described in separate documents.

2.2. Conventions in this document

This manual uses the `typewriter font` whenever OpenTPL protocol text is shown. Text that is enclosed in <*...*> stands as a placeholder. So `VARIABLE=`<*value*>[`,`<*value*>[`,...`]] means that some real values and not the string <value> is expected by the server. Sometimes parameters are optional. This is shown as in the previous example by putting the text in [...]. If these brackets are used in the OpenTPL protocol itself, they will be typeset as `<...>` and `[...]`.

In the sections that describe the OpenTSI modules and variables, tables will be used with the following columns:

Name the OpenTPL module/variable name. For the full OpenTPL path, the prefix from the section caption needs to be added. For instance, if a variable `VAR` is discussed in the section describing the `HARDWARE.MONITOR` submodule, the OpenTPL path of this variable is `HARDWARE.MONITOR.VAR`.

T The data type of the OpenTPL variable. Valid types can be found in [2].

R The standard read level of the OpenTPL variable. While it is not recommended to do so, it is of course possible to adapt the individual levels of the different variables to specific needs. Therefore, clients should not be designed in a way that they expect the here documented levels, but either check the real levels or correctly cope with a `DENIED` error. Refer to [2] for an explanation of the read (and write) levels.

W The standard write level of the OpenTPL variable. The same comments as for the read level hold here as well.

Description The detailed explanation of the functionality behind the variable.

3. List of basic modules and general comments

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

Name	Function
`TELESCOPE`	Status and general setup of the telescope (see section 4)
`OBJECT`	Target object specification in various coordinate systems (see section 5)
`POINTING`	Setup and control of pointing/tracking (see section 6)
`CURRENT`	Information on the currently tracked object and its setup (see section 7)
`POSITION`	Positions of all axes of the telescope (see section 8)
`AUXILIARY`	Control of auxiliary components of the telescope (see section 9)

3.1. Testing the availability of a module

While most OpenTSI modules are mandatory, some can be unavailable depending on the underlying telescope. These modules will be marked with "(optional)" in the detailed description in the following sections. Availability of such modules can be tested by reading their `VERSION` variable. It will return a non-zero value if the module exists.

If a module is not available, it may also be missing at all, resulting in an OpenTPL error message, reporting this module (or the <*module*>`.VERSION` variable) as `UNKNOWN`.

Some of the modules are defined as an array (shown as <*module*>`[]`) to allow multiple instances. If only one instance is available, the module may not be defined as an array (unless the array index is used to define the position of the device on the telescope).

3.2. Common module variables

Every module must have at least the following variable, which will not be listed in the detailed module description sections:

<table>
<tr><th>Name</th><th>T</th><th>R</th><th>W</th><th>Description</th></tr>
<tr><td><code>VERSION</code></td><td><code>i</code></td><td></td><td>-1</td><td>Interface version (<code>I</code>), interface age (<code>A</code>) and revision number (<code>R</code>), coded as <code>0xIIIIAARR</code> ($\neq 0$ if the module exists, $= 0$ if it does not):

The interface version will be increased whenever variables are added or removed. The latest OpenTSI version is <code>0x0020</code> which corresponds to version 2.0. This number must be the same for all available OpenTSI modules on a single server.
The interface age allows clients to check if the discovered 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 revision number indicates the code revision. In general: the higher, the better, fewer bugs etc.
</td></tr>
</table>

3.3. Command execution times

When reading from variables, values will be returned immediately (e.g. the typical callback run time is in the order of 1 ms). However, writing to variables may in some cases, where the completion of the triggered action is waited for, take longer times to finish. Therefore, for every writable variable where `SET` will not return immediately, this is explicitly stated in the following chapters.

3.4. Permissions

The OpenTSI allows read access to all variables for any user (regardless of its `RLEVEL`). Writing to variables has however been restricted by functional groups. This of course will not allow fine grained access control, but makes it possible to declare users that for instance can only track new objects, while others can power up/down the telescope and the one with the lowest `WLEVEL` can also modify the configuration of the telescope software. The following table gives an overview of the functional groups accessible for a certain `WLEVEL`.

`WLEVEL`	Access rights
50	Access to `OBJECT` and `POINTING` (besides `POINTING.TRACK`) to allow calculation but no influence on the telescope state
40	Access to `POINTING.TRACK` and `POSITION.INSTRUMENTAL.`<*>`.OFFSET` to allow acquisiton of new objects and centering those. Only errors of type `INFO` can be cleared.
30	Access to `TELESCOPE.READY` and `AUXILIARY` to allow power up/down of the telescope and opening/closing of dome and mirror covers etc. Only errors up to `WARNING` can be cleared.

WLEVEL	Access rights
20	Access to POSITION.INSTRUMENTAL and TELESCOPE.CONFIG besides SAVE to allow model creation and configuration changes. Only errors up to ERROR can be cleared.
10	Access to all remaining areas, especially TELESCOPE.CONFIG.<*>.SAVE to save configuration changes permanently. Clearing of any errors allowed.
0	Admin user, allowed to all areas of the OpenTSI and the management interface of the OpenTPL server (SERVER module).

3.5. Private Data

To guarantee consistent data, the OBJECT, POINTING and CURRENT.TRAJECTORY modules are using OpenTPL SYSVARs. This means, that data in this modules is connection specific. Each connection uses its own set of variables and can modify settings without other connections seeing this change. To allow mutual modifications of these areas by multiple connections, an additional set of variables exists which is global. It can be selected by setting POINTING.GLOBAL_DATA to 1.

4. The TELESCOPE module

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

Name	T	R	W	Description
POWER	i		30	Power up/down the telescope (without moving to startup or park position, however telescope will perform referencing of axes if necessary). Possible values are: **0** Turn off the telescope **1** Turn on the telescope SET will return after the telescope has reached the desired operation state or on an error.
PARK	i		30	Park/unpark the telescope. The telescope needs to be turned on to successfully complete this command. Possible values are: **0** Unpark, i.e. move accordingly configured telescope axes to their startup positions **1** Park, i.e. move accordingly configured telescope axes to their parking positions SET will return after the telescope axes have reached their positions or on an error.

Name	T	R	W	Description
`READY`	i		30	Prepare the telescope to become operational. Possible values are: **0** Power down telescope (same as setting `PARK`=1 and, after completion, setting `POWER`=0) **1** Power up telescope (same as setting `POWER`=1 and, after completion, setting `PARK`=0) `SET` will return after the telescope has reached the desired operation state or on an error.
`READY_STATE`	f		-1	The current operation status of the telescope. Possible values are: **-3.0** Local mode, no remote operation possible **-2.0** Emergency stop **-1.0** Errors block operation **0.0** Shut down **>0.0 … <1.0** Power up/down in progress **1.0** Fully operational
`MOTION_STATE`	i		-1	Returns the status of the telescope regarding ongoing movements. If the telescope is currently not moving, a value of zero is returned. The status is bit coded (bits not listed below are reserved and need to be zero): **Bit 0** One or more axes are moving **Bit 1** Trajectories are being executed **Bit 2** Movement is blocked (e.g. by limits) **Bit 3** Telescope is on target position (i.e. stopped or, if tracking, following the target) **Bit 4** Movement is restricted by given jerk/acceleration/velocity limits **Bit 5** Telescope is “unparked”, i.e. moved on startup position by `READY`=1 or `PARK`=0 **Bit 6** Telescope is “parked”, i.e. moved on park position by `READY`=0 or `PARK`=1
`REFERENCING_AXES`	s		-1	A comma separated list of axes that are not yet referenced. The axis names are the same as for the `POSITION.INSTRUMENTAL` axes.

Name	T	R	W	Description
SLEWING_AXES	s		-1	A comma separated list of axes that are currently slewing, i.e. not having reached their target position and are still moving towards it. If they cannot reach the position anymore due to an error or because the position lies beyond the limit, they won't be listed here but in BLOCKED_AXES instead. The axis names are the same as for the POSITION.INSTRUMENTAL axes.
BLOCKED_AXES	s		-1	A comma separated list of axes that are currently blocked, either due to an error or by sitting at a software or hardware limit. The axis names are the same as for the POSITION.INSTRUMENTAL axes.
STOP	i		40	On write, all telescope operations and axis movements are stopped as soon as possible. There is no guarantee that the axes will really be stopped successfully, especially if this command is issued while the telescope system reports errors. Therefore, this function may not be used as a replacement for emergency stop!
INFO	M			General information about the used telescope (see section 4.1)
CONFIG	M			Information about the telescope configuration (see section 4.2)
MEASUREMENT	M			Handling of measurements for telescope models (pointing, active M1/M2, ...) (see section 4.3)
STATUS	M			Detailed information on the telescope's operational state (see section 4.4)

4.1. 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	T	R	W	Description
NAME	s		-1	Telescope name
DIAMETER	f		-1	Telescope main mirror diameter in meters
CABINET	s		-1	Manufacturer of the control cabinet
MANUFACTURER	s		-1	Manufacturer of the telescope software
DOME	M			General information about the astrodome (see section 4.1.1)

4.1.1. The TELESCOPE.INFO.DOME submodule

This module provides some general information about the astrodome (if available), mainly for informational purposes.

Name	T	R	W	Description
NAME	s		-1	Astrodome name
DIAMETER	f		-1	Astrodome diameter

Name	T	R	W	Description
`SLITWIDTH`	f		-1	Width of astrodome slit, if available
`CABINET`	s		-1	Manufacturer of the astrodome control cabinet

4.2. The `TELESCOPE.CONFIG` submodule

This module provides some general configuration information about the telescope.

Name	T	R	W	Description
`CAPABILITIES`	i		-1	The bit coded features supported by the hardware: **Bit 0** Telescope can be operated as altazimuth mount with azimuth and zenith distance axes **Bit 1** Telescope can be operated as equatorial mount with hour angle and declination axes **Bit 2** Telescope can be operated as alt-alt mount with two zenith distance axes
`MOUNTTYPE`	i		10	Selection of telescope mount type: **1** Altazimuth mount with azimuth and zenith distance axes **2** Equatorial mount with hour angle and declination axes **3** Alt-alt mount with two zenith distance axes On read: The currently active mount type.. On write: Select the mount type to use from the available ones (see above). The OpenTSI may close all connections after this command, since it will be reconfigured for the new mount setup.
`ORIENTATION`	f		-1	Only for mount type 3 (Alt-alt): Azimuth of `ZD[0]` rotation axis. [°]
`LOAD`	i	-1	20	Load last saved configuration from the configuration files, when writing 1.The OpenTSI may close all connections after this command, since it will be restarted with the new configuration information. `SET` returns after the configuration has been loaded.

Name	T	R	W	Description
SAVE	i	-1	10	Save the currently active configuration to the configuration files, when writing 1. SET returns after the configuration has been saved. (Beside saving all configuration data with this command, it is also possible to save certain areas of the configuration selectively, see sections 4.2.1 through 4.2.11) *Note:* Settings from configuration files that did not exist during the loading of the configuration will not be saved, even if the settings have been changed. In this case, those settings need to be saved explicitly from the respective module, which will create the corresponding configuration file.
LOCAL	M			Configuration of time and location information (see section 4.2.1)
ENVIRONMENT	M			Configuration of environment information (see section 4.2.2)
POINTING	M			Configuration of telescope pointing (see section 4.2.3)
SIDEROSTAT	M			Specific information and configuration for the siderostat mode (see section 4.2.4)
ACTIVE_M1	M			Specific information and configuration of the active M1 support (e.g. mirror bending units of an active optics system) (see section 4.2.5)
ACTIVE_M2	M			Specific information and configuration of the active M2 support (e.g. hexapod) (see section 4.2.6)
COVER	M			Specific information and configuration of the main mirror covers (see section 4.2.7)
PORT_SELECT	M			Specific information and configuration of the port selector (e.g. moveable M3 mirror) (see section 4.2.8)
PORT[]	M[]			Information and configuration for each optical port of the telescope (see section 4.2.9)
DOME	M			Astrodome specific information and configuration (see section 4.2.10)
				Only for mount type 1
AZ	M			Specific information and configuration of the telescope azimuth (see section 4.2.11)
ZD	M			Specific information and configuration of the telescope zenith distance (see section 4.2.11)
				Only for mount type 2
HA	M			Specific information and configuration of the telescope hour angle (see section 4.2.11)
DEC	M			Specific information and configuration of the telescope declination (see section 4.2.11)
				Only for mount type 3
ZD[]	M[]			Specific information and configuration of the telescope zenith distances (see section 4.2.11)

4.2.1. The `TELESCOPE.CONFIG.LOCAL` submodule

In this module, the OpenTSI global location and time settings can be configured. For instance this data could be regularily updated by background daemons. The values can be overridden for individual pointings using `POINTING.SETUP.LOCAL` (see section 6.1.1).

Name	T	R	W	Description
`SYNCMODE`	i		20	Synchronize telescope location with information provided by the telescope hardware. The value is bit coded (if the bit is set, the value will be used from the telescope hardware, if the bit is cleared, the value will used from the fields below): **Bit 0** `LATITUDE` **Bit 1** `LONGITUDE` **Bit 2** `HEIGHT` **Bit 3** `UT1-UTC` **Bit 4** `TAI-UTC` Writing to any of the variables below (besides `SAVE` and `LOAD`) will set the corresponding bit to 0. Reading from the variables below will always return the active value (even if the bit for synchronisation is set)
`LATITUDE`	f		20	Latitude of telescope site, positive for northern hemisphere [°]
`LONGITUDE`	f		20	Longitude of telescope site, positive east of Greenwich [°]
`HEIGHT`	f		20	Height of telescope site above sea level [m]
`UT1-UTC`	f		20	Difference between UT1 and UTC [s]
`TAI-UTC`	f		20	Difference between TAI and UTC [s]
`SAVE`	i	-1	10	Save the currently active location and time configuration to the configuration file, when writing 1. `SET` returns after the configuration has been saved.
`LOAD`	i	-1	20	Reload the location and time configuration from the configuration file, when writing 1.

4.2.2. The `TELESCOPE.CONFIG.ENVIRONMENT` submodule

This module allows the configuration of the OpenTSI global environment settings (used for refraction calculation). This data can for instance be updated by a custom background daemon that writes data from a weather station in these variables. The values can be overridden for individual pointings using `POINTING.SETUP.ENVIRONMENT` (see section 6.1.2).

Name	T	R	W	Description
SYNCMODE	i		20	Synchronize with environment monitor connected to the telescope. The value is bit coded (if the bit is set, the value will be used from the telescope hardware, if the bit is cleared, the value will used from the fields below): **Bit 0** TEMPERATURE **Bit 1** PRESSURE Writing to any of the variables below (besides SAVE and LOAD) will set the corresponding bit to 0. Reading from the variables below will always return the active value (even if the bit for synchronisation is set)
TEMPERATURE	f		20	Ambient Temperature [°C]
PRESSURE	f		20	Air pressure on site [mBar]
SAVE	i	-1	10	Save the currently active environment data to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the environment configuration from the configuration file, when writing 1.

4.2.3. The TELESCOPE.CONFIG.POINTING submodule

This module allows the configuration of some pointing specific settings

Name	T	R	W	Description
HORIZON_LIMIT	s		20	Horizon limit of the telescope. The semicolon separated list consists of azimuth positions and minimum allowed elevations (separated by commas) in the following format: $<$*start_azimuth*$>$,$<$*min_elevation*$>$[;$<$*start_azimuth*$>$, $<$*min_elevation*$>$[;...]] The allowed ranges are $0° \leq start\_azimuth < 360°$ and $-2.5° \leq min\_elevation \leq 90°$. The ranges will be sorted ascending in azimuth. For a certain azimuth, the elevation value for the largest azimuth from this list that is smaller is used. The largest given azimuth will be used until the end of the azimuth range and until the smallest given azimuth (wrap-around). If the list is empty, a $<$*min_elevation*$>$ of -2.5° is used for all azimuth values.
OVERRIDABLE_HORIZON	i		-1	The horizon limit can be overridden using POINTING.SETUP.IGNORE_HORIZON, when reading 1. Writing is not allowed for security reasons, value needs to be changed directly in the configuration file.
ALLOW_SUN	i		-1	Pointing to sun (via OBJECT.SOLARSYSTEM) allowed, when reading 1 or forbidden, when reading 0. Writing is not allowed for security reasons, value needs to be changed directly in the configuration file.

Name	T	R	W	Description
ALLOW_MOON	i		-1	Pointing to moon (via OBJECT.SOLARSYSTEM) allowed, when reading 1 or forbidden, when reading 0. Writing is not allowed for security reasons, value needs to be changed directly in the configuration file.
SAVE	i	-1	10	Save the currently active pointing settings to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the pointing settings from the configuration file, when writing 1.

4.2.4. The TELESCOPE.CONFIG.SIDEROSTAT submodule

This modules provides configuration information about the siderostat mode.

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0** Mount operates as siderostat/heliostat/uranostat (beam will be reflected by a mirror on the mount to a fixed target)
INSTRUMENT_AZ	f		20	Azimuth coordinate of fixed target (instrument or secondary mirror) in true horizontal coordinates [°]
INSTRUMENT_ALT	f		20	Elevation coordinate of fixed target (instrument or secondary mirror) in true horizontal coordinates [°]
PORT_AZ_ORIENTATION	f		20	Orientation of mount azimuth in respect to true horizontal azimuth at first observation port [°]
PORT_AZ_SPACING	f		20	Azimuth spacing between observation ports [°]
SAVE	i	-1	10	Save the currently active configuration of the siderostat mode to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the siderostat configuration from the configuration file, when writing 1.

4.2.5. The TELESCOPE.CONFIG.ACTIVE_M1 submodule

This modules provides configuration information about an active M1 support, e.g. active optics system or electric collimation support.

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0** System consists of several actuators that can be moved between -1...+1, with 0 being the forceless middle position. The number of actuators is available from the dimension of the STARTUP_POS/PARK_POS arrays.
STARTUP_POS[]	f[]		20	Startup position for each actuator of the active M1 system. A NULL value will disable moving this actuator to a startup position.

Name	T	R	W	Description
PARK_POS[]	f[]		20	Park position for each actuator of the active M1 system. A NULL value will disable moving this actuator to a park position.
SAVE	i	-1	10	Save the currently active configuration of the active M1 system to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the configuration of the active M1 system from the configuration file, when writing 1.
MODEL	M			Model for the active M1 system (see section 4.2.5.1)

4.2.5.1. The TELESCOPE.CONFIG.ACTIVE_M1.MODEL submodule

In this module, the model for the active M1 correction is defined (see appendix B.4.1).

Name	T	R	W	Description
TYPE	i		20	Select active M1 model type: **-1** None **0** SAMPLE, model is based on individual measurements (actuator positions of closest measurement will be used)
CALCULATE	f		20	On write: The model coefficients are calculated from the measurements in TELESCOPE.MEASUREMENT.ACTIVE_M1 when writing 1. SET will return after the model has been calculated, which can take a while especially if many measurements were taken. On read: Return total error of currently active model in respect to currently loaded measurements
LIST	s		20	List of all measurements used for the active model. The format is: <*name*>,<*zenith distance*>,<*focus offset*>,<*act*$_1$>,[<*act*$_2$>[,...]][;<*name*>...[;...]]
DATA[]	M[]			Coefficients for the various model types (see section 4.2.5.1.1)

4.2.5.1.1. The TELESCOPE.CONFIG.ACTIVE_M1.MODEL.DATA[] submodule

This module array contains the available active M1 models. Which and how many are available depends on the implementation and possibly also on the mount type. Refer to appendix B.4.1 for the most common models.

Name	T	R	W	Description
NAME	s		-1	Name of the active M1 model. The following names are predefined, all others are reserved: **SAMPLE** See appendices B.4.1
COEFFICIENT[]	M[]			Coefficients for this model (see section 4.2.5.1.2)

4.2.5.1.2. The TELESCOPE.CONFIG.ACTIVE_M1.MODEL.DATA[].COEFFICIENT[] submodule

This module array contains the coefficient for the active M1 model. For the models described in appendix B.4.1, the same coefficient names will be used here.

Name	T	R	W	Description
NAME	s		-1	Name of the coefficient (as described in appendix B.4.1)
VALUE	f		20	Value of the coefficient

4.2.6. The TELESCOPE.CONFIG.ACTIVE_M2 submodule

This module provides configuration information about an active M2 support, e.g. linear focus or hexapod system for collimation and compensating gravity effects. The model for the active M2 system is defined on a per-port basis, please refer to section 4.2.9.6 for more information.

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0** System allows vertical alignment (Z, arbitrary sign for movement towards M1) **Bit 1** System allows horizontal alignment (X/Y, arbitrary orientation but perpendicular to Z-axis) **Bit 2** System allows tip-tilt alignment (U (rotation around X-axis)/V (rotation around Y-axis)) **Bit 3** System allows rotation around vertical alignment axis (W)
STARTUP_POS[]	f[]		20	Startup position for each actuator of the active M2 system. A NULL value will disable moving this actuator to a startup position. The orientation of the coordinate system is described in [1]. **0** X axis [mm] **1** Y axis [mm] **2** Z-axis [mm] **3** U-axis [°] **4** V-axis [°] **5** W-axis [°]

Name	T	R	W	Description
PARK_POS[]	f[]		20	Park position for each actuator of the active M2 system. A NULL value will disable moving this actuator to a park position. The orientation of the coordinate system is described in [1]. **0** X axis [mm] **1** Y axis [mm] **2** Z-axis [mm] **3** U-axis [°] **4** V-axis [°] **5** W-axis [°]
FIT_WEIGHT[]	f[]		20	Fit weight for each axis of the active M2 system (with exception of the W-axis). The orientation of the coordinate system is described in [1]. **0** X axis [1/mm] **1** Y axis [1/mm] **2** Z-axis [1/mm] **3** U-axis [1/°] **4** V-axis [1/°]
SAVE	i	-1	10	Save the currently active configuration of the active M2 system to the configuration file: **1** Save all active M2 configuration data SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the active M2 configuration from the configuration file: **1** Load all active M2 configuration data

4.2.7. The TELESCOPE.CONFIG.COVER submodule

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0** Open/close main mirror cover(s)
STARTUP_POS	f		20	Startup position for mirror covers. A NULL value will disable moving the mirror covers to a startup position.
PARK_POS	f		20	Park position for mirror covers. A NULL value will disable moving the mirror covers to a park position.

Name	T	R	W	Description
SAVE	i	-1	10	Save the currently active configuration of the mirror covers to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the configuration of the mirror covers from the configuration file, when writing 1.

4.2.8. The TELESCOPE.CONFIG.PORT_SELECT submodule

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0…13** TCI Port is available
STARTUP_POS	f		20	Startup position for the port selector. A NULL value will disable moving the port selector to a startup position.
PARK_POS	f		20	Park position for the port selector. A NULL value will disable moving the port selector to a park position.
SAVE	i	-1	10	Save the currently active configuration of the port selector to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the configuration of the port selector from the configuration file, when writing 1.

4.2.9. 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 (refer to [1]). The MODEL[] and ACTIVE_M2[] submodule allows access to the pointing model resp. active M2 model coefficients of all optical ports, for both normal and reverse orientation, including their calculation from the global measurements (in TELESCOPE.MEASUREMENT).

Name	T	R	W	Description
FOCUS_OFFSET	f		20	Relative focus correction for this port (will be applied if selected in POINTING.SETUP.FOCUS.SYNCMODE, see section 6.1.4) [mm]
SAVE	i	-1	10	Save the currently active configuration of the port to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the configuration of the port from the configuration file, when writing 1.
PORT_COVER	M			Specific information and configuration of a port cover on this optical port (see section 4.2.9.1)
DEROTATOR	M			Specific information and configuration of a derotator on this optical port (see section 4.2.9.2)
FILTER	M			Specific information and configuration of a filter wheel on this optical port (see section 4.2.9.3)
INSTRUMENT	M			Specific information and configuration of an instrument selector on this optical port (see section 4.2.9.4)

Name	T	R	W	Description
MODEL[]	M[]			The mount models (correction for mechanical errors and errors due to the optical alignment) for this optical port (see section 4.2.9.5): **0** for normal pointing **1** for reverse pointing
ACTIVE_M2[]	M			The active M2 models for this optical port (see section 4.2.9.6): **0** for normal pointing **1** for reverse pointing

4.2.9.1. The TELESCOPE.CONFIG.PORT[].PORT_COVER submodule

This module provides configuration information about a port cover on the selected optical port.

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0** Open/close port cover(s)
STARTUP_POS	f		20	Startup position for the port cover. A NULL value will disable moving the port cover to a startup position.
PARK_POS	f		20	Park position for the port cover. A NULL value will disable moving the port cover to a park position.
SAVE	i	-1	10	Save the currently active configuration of the port cover to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the port cover configuration from the configuration file, when writing 1.

4.2.9.2. The TELESCOPE.CONFIG.PORT[].DEROTATOR submodule

This module provides configuration information about a derotator on the selected optical port.

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0** Derotator with extended range ($> 360\,°$) **Bit 1** Derotator can move infinitely
STARTUP_POS	f		20	Startup position for the derotator. A NULL value will disable moving the derotator to a startup position.
PARK_POS	f		20	Park position for the derotator. A NULL value will disable moving the derotator to a park position.

Name	T	R	W	Description
`SAVE`	i	-1	10	Save the currently active configuration of the derotator to the configuration file, when writing 1. `SET` returns after the configuration has been saved.
`LOAD`	i	-1	20	Reload the derotator configuration from the configuration file, when writing 1.
`MODEL`	M			Model for derotator pointing correction (see section 4.2.9.2.1).

4.2.9.2.1. The `TELESCOPE.CONFIG.PORT[].DEROTATOR.MODEL` submodule This module allows the configuration of a pointing correction for the derotator (refer to appendix B.4.3 for details).

Name	T	R	W	Description
`DOFF`	f		20	Derotator offset [°]
`DES`	f		20	Eccentricity of derotator encoder, sine part [°]
`DEC`	f		20	Eccentricity of derotator encoder, cosine part [°]

4.2.9.3. The `TELESCOPE.CONFIG.PORT[].FILTER` submodule This module provides configuration information about a filter wheel on the selected optical port.

Name	T	R	W	Description
`CAPABILITIES`	i		-1	The bit coded features supported by the hardware: **Bit 0** Filter wheel with discrete positions. The number of filters is available from the dimension of the `NAME`/`FOCUS_OFFSET` arrays.
`STARTUP_POS`	f		20	Startup position for the filter wheel. A `NULL` value will disable moving the filter wheel to a startup position.
`PARK_POS`	f		20	Park position for the filter wheel. A `NULL` value will disable moving the filter wheel to a park position.
`NAME[]`	s[]		20	Symbolic filter names for name based filter selection
`FOCUS_OFFSET[]`	f[]		20	Relative focus correction for this filter (will be applied if selected in `POINTING.SETUP.FOCUS.SYNCMODE`, see section 6.1.4) [mm]
`SAVE`	i	-1	10	Save the currently active configuration of the filter wheel to the configuration file, when writing 1. `SET` returns after the configuration has been saved.
`LOAD`	i	-1	20	Reload the filter wheel configuration from the configuration file, when writing 1.

4.2.9.4. The `TELESCOPE.CONFIG.PORT[].INSTRUMENT` submodule This module provides configuration information about an instrument selector on the selected optical port.

Name	T	R	W	Description
CAPABILITIES	i		-1	The bit coded features supported by the hardware: **Bit 0** Instrument selector with discrete positions. The number of instruments is available from the dimension of the NAME/FOCUS_OFFSET arrays.
STARTUP_POS	f		20	Startup position for the instrument selector. A NULL value will disable moving the instrument selector to a startup position.
PARK_POS	f		20	Park position for the instrument selector. A NULL value will disable moving the instrument selector to a park position.
NAME[]	s[]		10	Symbolic instrument names for named based instrument selection
FOCUS_OFFSET[]	f[]		10	Relative focus correction for this instrument (will be applied if selected in POINTING.SETUP.FOCUS.SYNCMODE, see section 6.1.4) [mm]
SAVE	i	-1	10	Save the currently active configuration of the instrument selector to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the instrument selector configuration from the configuration file, when writing 1.

4.2.9.5. The TELESCOPE.CONFIG.PORT[].MODEL[] submodule This module holds pointing model coefficients for the selected optical port and also provides functionality to calculate a new model from taken measurements. The coefficients of the models can also be calculated by the user and then set here. The formulas of common models can be found in appendix B.

The following variables are available:

Name	T	R	W	Description
TYPE	i		20	Select pointing model type: **-1** None **0 ...** Index in the DATA[] array for a certain model type with a specific set of coefficients
CALCULATE	f		20	On write: The model coefficients are calculated from the measurements in TELESCOPE.MEASUREMENT.MODEL when writing 1. The new model is then used for the port and orientation this variable is accessed in, e.g. when using TELESCOPE.CONFIG.PORT[2].MODEL[0].CALCULATE, the normal orientation model for port 2 will be calculated. SET will return after the model has been calculated, which can take a while especially if many measurements were taken. On read: Return total error of currently active model in respect to currently loaded measurements [°]

Name	T	R	W	Description
RESIDUALS	s		-1	List of residual errors for each measurement in the `TELESCOPE.MEASUREMENT.MODEL` module in respect to the currently active pointing model for the port and orientation this variable is accessed in. $<ID>,<\chi_{A_1}>,<\chi_{A_2}>,<\chi_{\text{total}}>[;<ID>\ldots[;\ldots]];$ $-1,<\chi_{A_{1,\text{total}}}>,<\chi_{A_{2,\text{total}}}>,<\chi_{\text{grand\_total}}>$ The following fields are returned: **<*ID*>** A consecutive number referring to the measurement in the `TELESCOPE.MEASUREMENT.MODEL.LIST` the following errors apply to. The special last line with a number -1 contains only the total errors and does therefore not correspond to a real measurement. **$<\chi_{A_1}>/<\chi_{A_2}>$** The (signed) error of this measurement in relation to the current pointing model, corrected for its projection to the sky **$<\chi_{\text{total}}>$** The total error of the measurement as euclidean distance **$<\chi_{A_{1,\text{total}}}>/<\chi_{A_{2,\text{total}}}>/<\chi_{\text{grand\_total}}>$** RMS of individual errors of all measurements
SAVE	i	-1	10	Save the currently active pointing model coefficients to the configuration file, when writing 1. `SET` returns after the configuration has been saved.
LOAD	i	-1	20	Reload the pointing model coefficients from the configuration file, when writing 1.
DATA[]	M[]			Coefficients for the various model types (see section 4.2.9.5.1)

4.2.9.5.1. The `TELESCOPE.CONFIG.PORT[].MODEL[].DATA[]` submodule This module array contains the available pointing model. Which and how many are available depends on the implementation and possibly also on the mount type. Refer to appendix B for the most common models.

Name	T	R	W	Description
NAME	s		-1	Name of the pointing model. The following names are predefined, all others are reserved: **CLASSIC** See appendices B.1.1, B.2.1, B.3.1 **EXTENDED** See appendices B.1.2, B.2.2, B.3.2 **CUSTOM_<*name*>** User defined pointing model <*name*>
COEFFICIENT[]	M[]			Coefficients for this model (see section 4.2.9.5.2)

4.2.9.5.2. The TELESCOPE.CONFIG.PORT[].MODEL[].DATA[].COEFFICIENT[] submodule This module array contains the coefficient for the pointing model. For the models described in appendix B, the same coefficient names will be used here.

Name	T	R	W	Description
NAME	s		-1	Name of the coefficient (as described in appendix B)
VALUE	f		20	Value of the coefficient

4.2.9.6. The TELESCOPE.CONFIG.PORT[].ACTIVE_M2[] submodule In this module, the model for the active M2 correction is defined (see appendix B.4.2).

Name	T	R	W	Description
TYPE	i		20	Select active M2 model type: **-1** None **0 ...** Index in the DATA[] array for a certain model type with a specific set of coefficients
CALCULATE	f		20	On write: The model coefficients are calculated from the measurements in TELESCOPE.MEASUREMENT.ACTIVE_M2 when writing 1. The new model is then used for the port and orientation this variable is accessed in, e.g. when using TELESCOPE.CONFIG.PORT[2].ACTIVE_M2[0].CALCULATE, the normal orientation model for port 2 will be calculated. SET will return after the model has been calculated, which can take a while especially if many measurements were taken. On read: Return total error of currently active model in respect to currently loaded measurements

Name	T	R	W	Description
RESIDUALS	s		-1	List of residual errors for each measurement in the TELESCOPE.MEASUREMENT.ACTIVE_M2 module in respect to the currently active model for the port and orientation this variable is accessed in. $<ID>,<\chi_X>,<\chi_Y>,<\chi_Z>,<\chi_U>,<\chi_V>,<\chi_{\text{total}}>$ $[;<ID>...[;...]];-1,<\chi_{X,total}>,<\chi_{Y,total}>,$ $<\chi_{Z,total}>,<\chi_{U,total}>,<\chi_{V,total}>,<\chi_{\text{grand\_total}}>$ The following fields are returned: **<*ID*>** A consecutive number referring to the measurement in the TELESCOPE.MEASUREMENT.ACTIVE_M2.LIST the following errors apply to. The special last line with a number -1 contains only the total errors and does therefore not correspond to a real measurement. **$<\chi_X>$...$<\chi_V>$** The (signed) error of this measurement in relation to the current model **$<\chi_{\text{total}}>$** The total error of the measurement as euclidean distance **$<\chi_{X,total}>$...$<\chi_{V,total}>$/$<\chi_{\text{grand\_total}}>$** RMS of individual errors of all measurements
SAVE	i	-1	10	Save the currently active M2 coefficients to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the active M2 model coefficients from the configuration file, when writing 1.
DATA[]	M[]			Coefficients for the various model types (see section 4.2.9.6.1)

4.2.9.6.1. The TELESCOPE.CONFIG.PORT[].ACTIVE_M2[].DATA[] submodule This module array holds the coefficients for the different model types. Refer to appendix B.4.2 for a description of the individual models.

Name	T	R	W	Description
NAME	s		-1	Name of the active M2 model. The following names are predefined, all others are reserved: **Z_T** See appendix B.4.2.1 **XYZUV_GRAV_T** See appendix B.4.2.2
COEFFICIENT[]	M[]			Coefficients for this model (see section 4.2.9.6.2)

4.2.9.6.2. The `TELESCOPE.CONFIG.PORT[].ACTIVE_M2[].DATA[].COEFFICIENT[]` submodule This module array holds the individual coefficients for a certain model. A description of the model and its coefficients can be found in appendix B.4.2.

Name	T	R	W	Description
`NAME`	s		-1	Name of the coefficient (as described in appendix B.4.2)
`VALUE`	f		20	Value of the coefficient

4.2.10. The `TELESCOPE.CONFIG.DOME` submodule

This module provides configuration information for the astrodome.

Name	T	R	W	Description
`CAPABILITIES`	i		-1	The bit coded features supported by the hardware: **Bit 0** Astrodome has azimuth rotation **Bit 1** Astrodome has open/closeable slit **Bit 2** Astrodome has open/closeable flap **Bit 3** Astrodome has open/closeable seals **Bit 4** Astrodome has open/closeable vents **Bit 5** Astrodome has moveable windshield/blind/zenith distance **Bit 6** Astrodome has first moveable enclosure, e.g. simple fold enclosure **Bit 7** Astrodome has second moveable enclosure, e.g. two-part roll-roof dome **Bit 8** Astrodome azimuth with extended range ($> 360\,°$) **Bit 9** Astrodome azimuth can turn infinitely
`STARTUP_POS[]`	f[]		20	Startup position for all dome axes. A `NULL` value will disable moving this astrodome axis to a startup position. The dome axes are assigned as follows [1]: **0** Azimuth rotation [°] **1** Slit/First movable enclosure [0=Close, 1=Open] **2** Flap/Second movable enclosure [0=Close, 1=Open] **3** Seals [0=Close/inflate, 1=Open/deflate] **4** Vents [0=Close, 1=Open] **5** Windshield/blind/zenith distance [°]

Name	T	R	W	Description
PARK_POS[]	f[]		20	Park position for all dome axes. A NULL value will disable moving this astrodome axis to a park position. The dome axes are assigned as follows [1]: **0** Azimuth rotation [°] **1** Slit/First movable enclosure [0=Close, 1=Open] **2** Flap/Second movable enclosure [0=Close, 1=Open] **3** Seals [0=Close/inflate, 1=Open/deflate] **4** Vents [0=Close, 1=Open] **5** Windshield/blind/zenith distance [°]
MAX_DEVIATION	f		20	Maximum position deviation that the dome can have from the telescope azimuth. This is used for discrete dome tracking and can be overridden for individual pointings (see section 6.1.8). [°]
SAVE	i	-1	10	Save the currently active configuration of the astrodome to the configuration file, when writing 1. SET returns after the configuration has been saved.
LOAD	i	-1	20	Reload the configuration of the astrodome from the configuration file, when writing 1.
MODEL	M			Model for dome pointing correction due to parallax of non-centered or asymmetric mounts (e.g. German equatorial) (see section 4.2.10.1).

4.2.10.1. The TELESCOPE.CONFIG.DOME.MODEL submodule This modules allows the definition of a dome pointing correction model, taking asymmetric mounts (e.g. German equatorial) as well as non-centered mount setup into account (refer also to appendix B.4.4).

Name	T	R	W	Description
CAOFF	f		20	Astrodome azimuth offset [°]
CZOFF	f		20	Astrodome zenith distance offset [°]
CDMCN	f		20	Distance of mount center (intersection of mount axes) from astrodome center towards north [m]
CDMCE	f		20	Distance of mount center (intersection of mount axes) from astrodome center towards east [m]
CDMCZ	f		20	Distance of mount center (intersection of mount axes) from astrodome center towards zenith [m]
CDOMCA	f		20	Distance of optical axis from mount center (intersection of mount axes) in direction of second mount axis [m]
CDOMCP	f		20	Distance of optical axis from mount center (intersection of mount axes) in direction perpendicular to second mount axis [m]

4.2.11. The `TELESCOPE.CONFIG.`<*axis*> submodules

Depending on mount configuration, either the modules `AZ/ZD` (`MOUNTTYPE` 1), `HA/DEC` (`MOUNTTYPE` 2) or `ZD[]` (`MOUNTTYPE` 3) will exist. All of them have the same layout:

Name	T	R	W	Description
`CAPABILITIES`	i		-1	The bit coded features supported by the hardware. For the first telescope axis (`AZ`, `HA`, `ZD[0]`) this can be: **Bit 0** Axis has extended range ($> 360\,°$) **Bit 1** Axis can turn infinitely For the second telescope axis (`ZD`, `DEC`, `ZD[1]`) this can be: **Bit 0** Axis allows reverse pointing (e.g. `ZD` $< 0\,°$, `DEC`, `ZD[1]` $> 90\,°$)
`STARTUP_POS`	f		20	Startup position for the telescope axis. A `NULL` value will disable moving the telescope axis to a startup position.
`PARK_POS`	f		20	Park position for the telescope axis. A `NULL` value will disable moving the telescope axis to a park position.
`SAVE`	i	-1	10	Save the currently active configuration of the telescope axis to the configuration file, when writing 1. `SET` returns after the configuration has been saved.
`LOAD`	i	-1	20	Reload the configuration of the telescope axis from the configuration file, when writing 1.

4.3. The `TELESCOPE.MEASUREMENT` submodule

This module gives access to the global measurement handling for the different telescope models.

Name	T	R	W	Description
`MODEL`	M			Handling of measurements for pointing models (see section 4.3.1)
`ACTIVE_M1`	M			Handling of measurements for active M1 models (see section 4.3.2)
`ACTIVE_M2`	M			Handling of measurements for active M2 models (see section 4.3.3)

4.3.1. The `TELESCOPE.MEASUREMENT.MODEL` submodule

This module allows the administration of measurements for pointing model creation.

Name	T	R	W	Description
COUNT	i		-1	Number of measurements taken so far
LIST	s		-1	List of all measurements currently in memory in the format: <*ID*>,<*name*>,<A_1>,<Δ_{A_1}>,<A_2>,<Δ_{A_2}> [;<*ID*>,...[;...]] The following fields are returned: **<*ID*>** A consecutive number identifying the measurement, which can be used to delete that measurement via `REMOVE` and locate the residual error in the `RESIDUALS` variable of the `TELESCOPE.CONFIG.PORT[].MODEL[]` module. **<*name*>** The name of the tracked object or a user defined name when the measurement was added **<A_1>/<A_2>** The coordinates of the measurement in ideal mount coordinates. The meaning therefore depends on the used mount type **<Δ_{A_1}>/<Δ_{A_2}>** The total correction offset (e.g. including a possible correction from the pointing model that was active when the measurement was taken)
REMOVE	i	-1	20	Remove measurements from list: $n > 0$ remove n^{th} measurement $n < 0$ remove $\lvert n\rvert^{\text{th}}$ measurement from the end (-1: last, -2: second to last, ...)
CLEAR	i	-1	20	Clear measurement list, when writing 1.
NEW	M			Creation of a new measurement (see section 4.3.1.0.1)
FILE	M			Saving and loading of measurements (see section 4.3.1.0.2)

4.3.1.0.1. The `TELESCOPE.MEASUREMENT.MODEL.NEW` submodule With this module, new pointing model measurements can be created.

Name	T	R	W	Description
`ADD`	i	-1	20	On write: Create a new measurement and add it to the list of measurements: **1** Use current telescope position and `OFFSET`s of the telescope main axes **2** Use the currently tracked object and `RA`, `DEC` and `EQUINOX` in this module as the exact on sky position that has been calculated at `UTC` for instance by a WCS match. **3** Same as 2, but the exact position needs to be specified as true horizontal coordinates `AZ` and `ALT` **4** Same as 2, but instead of the tracked object, the object currently setup in the `OBJECT` and `POINTING` modules is used. In this case, it must be assured that all data including the pointing model matches with that at the time the exact on sky position has been calculated. For location, time and environmental data, the possibility to override those via `POINTING.SETUP.LOCAL` and `POINTING.SETUP.ENVIRONMENT` can be used (see sections 6.1.1 and 6.1.2). **5** Same as 4, but the exact position is assumed to be specified as true horizontal coordinates `AZ` and `ALT` *Note*: For adding modes 2...5, the telescope `OFFSET`s will not be taken into account. On read: Return <*ID*> of last added measurement.
`NAME`	s		20	Name of the measurement. For `ADD` mode 1–3, a `NULL` value will be replaced by the name of the currently tracked object, for the other modes by the name defined in the `OBJECT` module.
`UTC`	f		20	Time at which the exact position has been calculated (for modes 2–5). [UTC seconds since 01.01.1970 00:00:00]
`RA`	f		20	Exact right ascension position of the telescope that has been calculated or measured (for modes 2 and 4) [h]
`DEC`	f		20	Exact declination position of the telescope that has been calculated or measured (for modes 2 and 4) [°]
`EQUINOX`	f		20	Equinox of the coordinates (for modes 2 and 4). A value of `NULL` will be replaced by the equinox corresponding to `UTC`. [Julian year]
`AZ`	f		20	Exact azimuth position of the telescope that has been calculated or measured (for modes 3 and 5) [°]
`ALT`	f		20	Exact (unrefracted) altitude/elevation of the telescope that has been calculated or measured (for modes 3 and 5) [°]

4.3.1.0.2. The `TELESCOPE.MEASUREMENT.MODEL.FILE` submodule This module handles saving and loading measurements on disk and the administration of measurement files.

Name	T	R	W	Description
NAME	s		20	File name of measurement list. It must not contain "," and possibly other, operation system specific, special characters. It must also not contain path elements (invalid characters might be replaced by "_" or the setting of the file name could fail). The operations LOAD, SAVE and DELETE will all refer to the here selected file. The names of the available files can be acquired by reading LIST.
LOAD	i	-1	20	Load measurements from file NAME: **1** Load and overwrite current list **2** Load and append to current list
SAVE	i	-1	20	Save measurements to file NAME: **1** Save and overwrite file **2** Save and append to file
DELETE	i	-1	20	Delete file NAME, when writing 1.
AUTO_SAVE	i		20	Automatically append each new measurement to the file NAME: **0** Disabled **1** Enabled
LIST	s		-1	Return the names of all currently available measurement files as a comma separated list.

4.3.2. The TELESCOPE.MEASUREMENT.ACTIVE_M1 submodule

This module allows the administration of measurements for the active M1 model creation.

Name	T	R	W	Description
`COUNT`	i		-1	Number of measurements taken so far
`LIST`	s		-1	List of all measurements currently in memory in the format: <*ID*>,<*name*>,<*zenith distance*>,<*focus offset*>,<act_1>,[<act_2>[,...]][;<*ID*>,...[;...]] ***<ID>*** A consecutive number identifying the measurement, which can be used to delete that measurement via `REMOVE`. ***<name>*** The name of the tracked object or a user defined name when the measurement was added ***<zenith distance>*** The instrumental zenith distance position at which the measurement was taken ***<focus offset>*** The offset for the focus that compensates the defocus caused by the deformation of the M1with the actuators <act_i> Positions of the actuators of the active M1 system
`REMOVE`	i	-1	20	Remove measurements from list: $n > 0$ remove n^{th} measurement $n < 0$ remove $\lvert n\rvert^{\mathrm{th}}$ measurement from the end (-1: last, -2: second to last, ...)
`CLEAR`	i	-1	20	Clear measurement list, when writing 1.
`NEW`	M			Creation of a new measurement (see section 4.3.2.1)
`FILE`	M			Saving and loading of measurements (see section 4.3.2.2)

4.3.2.1. The `TELESCOPE.MEASUREMENT.ACTIVE_M1.NEW` submodule With this module, new measurements for the active M1 model can be created.

Name	T	R	W	Description
ADD	i	-1	20	On write: Create a new measurement and add it to the list of measurements: **1** Use current telescope and active M1 actuator positions and OFFSET of focus or hexapod Z axis **2** Use zenith distance position, focus offset and active M1 actuator position from this module (see below) On read: Return <*ID*> of last added measurement.
NAME	s		20	Name of the measurement. For ADD mode 1, a NULL value will be replaced by the name of the currently tracked object.
ZD	f		20	Instrumental zenith distance of the telescope for ADD mode 2 [°]
FOCUS_OFFSET	f		20	Offset in focus for this measurement for ADD mode 2. As the active M1 correction may lead to a varying defocus, this focus offset can be used as a correction when the correction is applied (if selected in POINTING.SETUP.FOCUS.SYNCMODE, see section 6.1.4) [mm]
AO_BENDER	f[]		20	Position array with the active M1 actuator positions for ADD mode 2 [-1..+1]

4.3.2.2. The TELESCOPE.MEASUREMENT.ACTIVE_M1.FILE submodule

This module handles saving and loading measurements on disk and the administration of measurement files.

Name	T	R	W	Description
NAME	s		20	File name of measurement list. It must not contain "," and possibly other, operation system specific, special characters. It must also not contain path elements. The operations LOAD, SAVE and DELETE will all refer to the here selected file. The names of the available files can be acquired by reading LIST.
LOAD	i	-1	20	Load measurements from file NAME: **1** Load and overwrite current list **2** Load and append to current list
SAVE	i	-1	20	Save measurements to file NAME: **1** Save and overwrite file **2** Save and append to file
DELETE	i	-1	20	Delete file NAME, when writing 1.
AUTO_SAVE	i		20	Automatically append each new measurement to the file NAME: **0** Disabled **1** Enabled
LIST	s		-1	Return the names of all currently available measurement files as a comma separated list.

4.3.3. The `TELESCOPE.MEASUREMENT.ACTIVE_M2` submodule

This module allows the administration of measurements for the active M2 model creation.

Name	T	R	W	Description
`COUNT`	i		-1	Number of measurements taken so far
`LIST`	s		-1	List of all measurements currently in memory in the format: *<ID>*,*<name>*,<A_1>,<A_2>,*<in-focus>*,<pos_X>,<pos_Y>,<pos_Z>,<pos_U>,<pos_V>,<$sensor_0$>,...,<$sensor_i$>[;*<ID>*,...[;...]] ***<ID>*** A consecutive number identifying the measurement, which can be used to delete that measurement via `REMOVE` and locate the residual error in the `RESIDUALS` variable of the `TELESCOPE.CONFIG.PORT[].ACTIVE_M2[]` module ***<name>*** The name of the tracked object or a user defined name when the measurement was added <A_1>/<A_2> The coordinates of the measurement in instrumental mount coordinates. The meaning therefore depends on the used mount type. ***<in-focus>*** Flag to mark measurement being focused or unfocused: **0** measurement was taken out of focus **1** measurement was take in focus <$pos_{<axis>}$> Position of the individual axes of the active M2 system <$sensor_i$> Reading of sensor with the index i. *Note*: Only sensors are handled for which values are available.
`REMOVE`	i	-1	20	Remove measurements from list: $n > 0$ remove n^{th} measurement $n < 0$ remove $\lvert n\rvert^{\text{th}}$ measurement from the end (-1: last, -2: second to last, ...)
`CLEAR`	i	-1	20	Clear measurement list, when writing 1.
`NEW`	M			Creation of a new measurement (see section 4.3.3.1)
`FILE`	M			Saving and loading of measurements (see section 4.3.3.1.1)

4.3.3.1. The `TELESCOPE.MEASUREMENT.ACTIVE_M2.NEW` submodule With this module, new measurements for the active M2 model can be created.

Name	T	R	W	Description
ADD	i	-1	20	Create a new measurement and add it to the list of measurements: **1** Use current telescope position and OFFSET of focus or hexapod Z axis **2** Use telescope position, focus offset, sensor values and hexapod position from this module (see below)
NAME	s		20	Name of the measurement. For ADD mode 1, a NULL value will be replaced by the name of the currently tracked object.
IN_FOCUS	i		20	Flag to mark measurement as taken in focus for ADD mode 1 and 2: **0** out of focus **1** in focus
X	f		20	X-axis position for ADD mode 2 [mm]
Y	f		20	Y-axis position for ADD mode 2 [mm]
Z	f		20	Z-axis position for ADD mode 2 [mm]
U	f		20	U-axis position for ADD mode 2 [°]
V	f		20	V-axis position for ADD mode 2 [°]
SENSOR[]	f[]		20	Sensor readings for ADD mode 2 [°C]
Only for mount type 1				
AZ	f		20	Instrumental azimuth position for this measurement for ADD mode 2
ZD	f		20	Instrumental zenith distance position for this measurement for ADD mode 2
Only for mount type 2				
HA	f		20	Instrumental hour angle position for this measurement for ADD mode 2
DEC	f		20	Instrumental declination position for this measurement for ADD mode 2
Only for mount type 3				
ZD[]	f[]		20	Instrumental zenith distance 0 and 1 position for this measurement for ADD mode 2

4.3.3.1.1. The TELESCOPE.MEASUREMENT.ACTIVE_M2.FILE submodule This module handles saving and loading measurements on disk and the administration of measurement files.

Name	T	R	W	Description
NAME	s		20	File name of measurement list. It must not contain "," and possibly other, operation system specific, special characters. It must also not contain path elements. The operations LOAD, SAVE and DELETE will all refer to the here selected file. The names of the available files can be acquired by reading LIST.
LOAD	i	-1	20	Load measurements from file NAME: **1** Load and overwrite current list **2** Load and append to current list
SAVE	i	-1	20	Save measurements to file NAME: **1** Save and overwrite file **2** Save and append to file
DELETE	i	-1	20	Delete file NAME, when writing 1.
AUTO_SAVE	i		20	Automatically append each new measurement to the file NAME: **0** Disabled **1** Enabled
LIST	s		-1	Return the names of all currently available measurement files as a comma separated list.

4.4. The TELESCOPE.STATUS submodule

This module covers error handling and telescope status reporting. The OpenTSI distinguishes two sources of errors:

1. Errors that occur during telescope operation by user interaction or by wrong user input (object is below horizon, coordinates are outside the allowed range etc.) are reported by OpenTPL event messages. Additionally, the command that caused the error will fail.
2. The special module TELESCOPE.STATUS will provide error status for the telescope hardware which is summarized in different functional hardware groups (see below). OpenTPL events are 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, refer to [1] for a list
SYSTEM	All global errors, e.g. power failure, emergency stop etc.
AUXILIARY	Other, e.g. SENSOR and SWITCH
UNKNOWN	Errors that cannot be linked to one of the groups above (this should normally not be the case)

The status module contains:

Name	T	R	W	Description
GLOBAL	i		-1	Global state of the telescope. Possible values are: **-2** no valid license data found, **-1** no telescope hardware found (no connection to an OpenTCI compliant telescope), **0** operational, **>0** bit coded errors, see below **Bit 0** PANIC, a severe condition, completely disabling the entire telescope, **Bit 1** ERROR, a serious condition, disabling important parts of the telescope system, **Bit 2** WARNING, a critical condition, which is not (yet) disabling the telescope, **Bit 3** INFO, a informal situation, which is not affecting the operation.
LIST	s		-1	A comma separated list of function groups that currently have problems in the following format: *<group>*`|`*<level>*[`:`*<component>*`|`*<level>*[`;`*<component>*...]] [`:`*<error>*`|`*<detail>*`|`*<level>*`|`*<component>*[`;`*<error>*...]] [`,`*<group>*...] ***<group>*** One of the above listed function groups ***<level>*** Bitwise "OR" of all errors in the group resp. component or for the individual error. The bits have the same meaning as for `GLOBAL`. ***<component>*** The OpenTCI module name (possibly including a index in `[]`). ***<error>*** The hardware specific error code ***<detail>*** The hardware specific detail information for the error code. The information from *<error>*/*<detail>* should only be used for logging, as it is hardware specific and may change at any time. At most one entry per group is generated. If the delimiters 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	T	R	W	Description
CLEAR_PANIC	i	-1	10	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 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.
CLEAR_ERROR	i	-1	20	Same as CLEAR_PANIC, but will fail if errors more severe than ERROR exist on the telescope (no errors are cleared in this case)
CLEAR_WARNING	i	-1	30	Same as CLEAR_PANIC, but will fail if errors more severe than WARNING exist on the telescope (no errors are cleared in this case)
CLEAR_INFO	i	-1	40	Same as CLEAR_PANIC, but will fail if errors more severe than INFO exist on the telescope (no errors are cleared in this case)

5. The OBJECT module

This is the module of the OpenTSI which provides functionality for object specification in various coordinate systems. It contains a submodule for each supported object coordinate system and a variable for the current object type. All variables in this module are usually private to the current connection (refer to section 3.5 for an explanation).

Name	T	R	W	Description
TYPE	i!		50	Selected object type, will also be automatically updated, if coordinates are written to one of the submodules. Possible values are: **1** Instrumental, object specification in INSTRUMENTAL submodule (see section 5.3) **2** Horizontal, object specification in HORIZONTAL submodule (see section 5.4) **3** Equatorial, object specification in EQUATORIAL submodule (see section 5.5) **4** Solar system object, object specification in SOLARSYSTEM submodule (see section 5.6) **5** Orbital elements of different type, object specification in ELEMENTS submodule (see section 5.7)
DIFFERENTIAL	M			Differential pointing, applied additionally to the selected object (see section 5.1)
INSTRUMENTAL	M			Instrumental coordinates, typically AZ/ZD, HA/DEC or ZD[] (see section 5.3)
HORIZONTAL	M			True horizontal coordinates, AZ and ALT or ZD (see section 5.4)

Name	T	R	W	Description
EQUATORIAL	M			Equatorial coordinates, RA, DEC with proper motions and in definable epoch/equinox (see section 5.5)
SOLARSYSTEM	M			Planets plus Pluto and some moons within the solar system (see section 5.6)
ELEMENTS	M			Orbital elements (parabolic, near-parabolic and elliptic) (see section 5.7)
TLE	M			Objects in earth orbit, specified as two-line elements (see section 5.8)

The specification of objects in these coordinate systems will be discussed in the next sections. Most coordinates can be given either as single position (e.g. RA, DEC) or as time-position trajectory which allows accurate and time synchronized arbitrary telescope movements.

After specifying an object in one of the above modules (and thereby changing the `TYPE` to the respective object type), the `INSTRUMENTAL`, `HORIZONTAL` and `EQUATORIAL` submodules can be used to retrieve the coordinates of this object in these respective coordinate systems. The `EQUATORIAL` module thereby takes the last set value for `EQUINOX` into account and returning its coordinates for this equinox. All other parameters (like proper motion and rates) are not taken into account.

If the object type is changed, all trajectory points of the former object type are discarded and some other values may be reset as well. If the type is set to 1, 2 or 3, the former object's coordinates are converted for this point in time to this new type (even, if the object was of this type before). This can for instance be used to explicitly track a non-equatorial object in equatorial mode by first setting it up and then changing the type to 3.

5.1. The OBJECT.DIFFERENTIAL submodule

This module allows to perform a differential pointing relative to the selected object. It is therefore not the type of an object (like the following sections) but a modification of that object.

Name	T	R	W	Description
TYPE	i!		50	Differential pointing type: **0** None (no differential pointing, values below are ignored) **1** Equatorial using cartesian coordinates on a tangent plane (differential pointing is applied to the apparent equatorial position). This will not be applied to objects of type 1 and 2. **2** Horizontal using cartesian coordinates on a tangent plane (differential pointing is applied to the apparent horizontal position). This will not be applied to objects of type 1
X	f!		50	First axis for differential pointing, oriented along right ascension (`TYPE` 1) resp. azimuth (`TYPE` 2) for `ANGLE` = 0° [°]
Y	f!		50	Second axis for differential pointing, oriented along declination (`TYPE` 1) resp. altitude (`TYPE` 2) for `ANGLE` = 0° [°]

Name	T	R	W	Description
ANGLE	f!		50	Rotation of the X/Y coordinate system in orientation of right ascension resp. azimuth (clockwise) [°]
SCALING	i!		50	Scaling of the fraction towards right ascension/azimuth: **0** no scaling **1** scaling with $^1/_{\cos}$ of declination/altitude

5.2. The `OBJECT.`<*type*>`.TRAJECTORY` submodule

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. These points are transferred block by block through the `BUFFER[]` and are added from there into the internal trajectory with `ADDPOINTS`. The provided time should be in the future (otherwise the points will be ignored). It is also possible to overwrite part of the trajectory by sending points that are within the trajectory's run time. The buffer structure will be described with the individual coordinate types. The following variables are available in all trajectory submodules:

Name	T	R	W	Description
ADDPOINTS	i!		50	On write: Add the number of buffer points (starting from `BUFFER[0]`) into the internal trajectory handling On read: Return the number of points that have been written on the last write access `SET` will return after the points have been added internally, usually rather quickly.
CLEAR	f!	-1	50	Clear all trajectory points with times that are later or equal the written value [UTC seconds since 01.01.1970 00:00:00] `SET` will return after the points have been deleted internally, usually rather quickly.
STARTTIME	f!		-1	Time of the first trajectory point (`NULL` if no trajectory is defined) [UTC seconds since 01.01.1970 00:00:00]
ENDTIME	f!		-1	Time of the last trajectory point (`NULL` if no trajectory is defined) [UTC seconds since 01.01.1970 00:00:00]
RUNTIME	f!		-1	Remaining run time of the trajectory (`NULL` if no trajectory is defined) [s]
BUFFER[]	M[]			Module array which is used as buffer for new sample points (layout is described at the individual coordinate sections) (see sections 5.3, 5.4 and 5.5)

5.3. The `OBJECT.INSTRUMENTAL` submodule

Setting coordinates or adding trajectory points will change `OBJECT.TYPE` to 1. The coordinate system and therefore the available variables in this module and its `TRAJECTORY` submodule will depend on the telescope mount type (`TELESCOPE.CONFIG.MOUNTTYPE`, see section 4.2). The following table lists the correlation between mount type and axes.

Name	T	R	W	Description
NAME	s!		50	Object name (for information only)
TRAJECTORY	M			Trajectory module (see section 5.2)
Only for mount type 1				
AZ	f!		50	Azimuth [°]
ZD	f!		50	Zenith distance, will overwrite ALT [°]
ALT	f!		50	Altitude/Elevation, will overwrite ZD [°]
Only for mount type 2				
HA	f!		50	Hour angle [°]
DEC	f!		50	Declination [°]
Only for mount type 3				
ZD[]	f![]		50	Zenith distance, array of two axes [°]

Structure of the trajectory buffer is:

Name	T	R	W	Description
UTC	f!		50	UTC [seconds since 01.01.1970 00:00:00]
DEROTATOR	f!		50	Derotator (depending on POSITION.SETUP.DEROTATOR.SYNCMODE considered absolute or as offset to its calculated orientation) [°]
Only for mount type 1				
AZ	f!		50	Azimuth [°]
ZD	f!		50	Zenith distance [°]
Only for mount type 2				
HA	f!		50	Hour angle [°]
DEC	f!		50	Declination [°]
Only for mount type 3				
ZD[]	f![]		50	Zenith distance, array of two axes [°]

5.4. The OBJECT.HORIZONTAL submodule

Setting coordinates or adding trajectory points will change OBJECT.TYPE to 2.

Name	T	R	W	Description
AZ	f!		50	Azimuth [°]
ALT	f!		50	Altitude/elevation, will overwrite ZD [°]
ZD	f!		50	Zenith distance, will overwrite ALT [°]
NAME	s!		50	Object name (for information only)
AIR_MASS	f!		-1	Air mass (calculated according to Rozenberg (1966) for sea level elevation, taking refraction into account)
REFRACTION	f!		-1	Altitude offset due to atmospheric refraction (calculated as currently selected in POINTING.SETUP.REFRACTION), adding it to ALT will yield the refracted altitude [°]
TRAJECTORY	M			Trajectory module (see section 5.2)

Structure of the trajectory buffer is:

Name	T	R	W	Description
UTC	f!		50	UTC [seconds since 01.01.1970 00:00:00]

Name	T	R	W	Description
AZ	f!		50	Azimuth [°]
ALT	f!		50	Altitude [°]
DEROTATOR	f!		50	Derotator (depending on POSITION.SETUP.DEROTATOR.SYNCMODE considered absolute or as offset to its calculated orientation) [°]

5.5. The OBJECT.EQUATORIAL submodule

Setting coordinates or adding trajectory points will change OBJECT.TYPE to 3.

Name	T	R	W	Description
EPOCH	f!		50	Epoch of the coordinates. A value of NULL will be replaced by the current epoch. [Julian year]
EQUINOX	f!		50	Equinox of the coordinates. A value of NULL will be replaced by the current equinox. [Julian year]
RA	f!		50	Right ascension [h]
DEC	f!		50	Declination [°]
RA_PM	f!		50	Proper motion in right ascension [h/year]
DEC_PM	f!		50	Proper motion in declination [°/year]
RATE_START	f!		50	Start time for the additional rates RA_RATE and DEC_RATE (see below). At this time, the object coordinates are exactly RA and DEC. A value of NULL will be replaced by the current time. [UTC seconds since 01.01.1970 00:00:00]
RA_RATE	f!		50	Additional rate in right ascension (for non-stellar objects). This rate is applied to RA in relation to the start time RATE_START. [h/s]
DEC_RATE	f!		50	Additional rate in declination (for non-stellar objects). This rate is applied to DEC in relation to the start time RATE_START. [°/s]
NAME	s!		50	Object name (for information only)
TRAJECTORY	M			Trajectory module (see section 5.2)

Structure of Trajectory buffer is:

Name	T	R	W	Description
UTC	f!		50	UTC [seconds since 01.01.1970 00:00:00]
RA	f!		50	Right ascension [h]
DEC	f!		50	Declination [°]
DEROTATOR	f!		50	Derotator (depending on POSITION.SETUP.DEROTATOR.SYNCMODE considered absolute or as offset to its calculated orientation) [°]

The EQUINOX setting will be considered for trajectories as well, but cannot be specified for each point.

5.6. The OBJECT.SOLARSYSTEM submodule

Selecting solar system objects will change OBJECT.TYPE to 4.

Name	T	R	W	Description
`OBJECT`	i!		50	Number of object. Possible Values: **0** Sun **1** Mercury **2** Venus **3** Earth **4** Mars **5** Jupiter **6** Saturn **7** Uranus **8** Neptune **9** Pluto If `MOON` has a value that does not match to the set planet, `MOON` will be reset to 0 or, in case of Earth, to 1.
`MOON`	i!		50	Moon of the object (only Earth, Jupiter and Saturn): **0** Object itself **Earth** **1** Moon **Jupiter** **1** Io **2** Europa **3** Ganymede **4** Callisto **Saturn** **1** Mimas **2** Enceladus **3** Tethys **4** Dione **5** Rhea **6** Titan **7** Hyperion **8** Iapetus If the moon is not valid for the selected `OBJECT`, it will be reset to 0 or, in case of Earth, to 1.
`NAME`	s!		-1	Object name (for information only)
`DISTANCE`	f!		-1	Object distance (for information only) [AU]

Tracking or pointing to the sun and/or moon may be disabled in the configuration files for security reasons. In this case, trying to track those objects will fail. However, tracking them or pointing towards them by other means (e.g. using an equatorial specification) will still be possible!

5.7. The OBJECT.ELEMENTS submodule

Setting orbital elements parameters will change `OBJECT.TYPE` to 5.

Name	T	R	W	Description
`TYPE`	`f!`		50	Type of orbital elements: **0** Elliptic **1** Parabolic **2** Near-parabolic Some coefficients are only valid for certain types, see below for a detailed description.
`EQUINOX`	`f!`		50	All types: Equinox of the coefficients [Julian year]
`T`	`f!`		50	All types: Time of passage through perihelion [seconds since 01.01.1970 00:00:00]
`I`	`f!`		50	All types: Inclination [°]
`OMEGA_PERIHELION`	`f!`		50	All types: Argument of perihelion [°]
`OMEGA_NODE`	`f!`		50	All types: Longitude of ascending node [°]
`A`	`f!`		50	Elliptic: Semi-major axis [AU]
`E`	`f!`		50	Elliptic/near-parabolic: Eccentricity
`Q`	`f!`		50	Parabolic/near-parabolic: Perihelion distance [AU]
`NAME`	`s!`		50	Object name (for information only)
`DISTANCE`	`f!`		-1	Object distance (for information only) [AU]

5.8. The OBJECT.TLE submodule

Setting two-line elements parameters will change `OBJECT.TYPE` to 6.

Name	T	R	W	Description
`LINE1`	`s!`		50	First line of the two-line element set.
`LINE2`	`s!`		50	Second line of the two-line element set.
`NAME`	`s!`		-1	The name of the object, composed of launch year, number and piece (for information only)
`SUN_ALTITUDE`	`f!`		-1	Unrefracted altitude of the sun from the object (for information only) [°]
`DISTANCE`	`f!`		-1	Object distance (for information only) [AU]

6. 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. All variables in this module are usually private to the current connection (refer to section 3.5 for an explanation). The following variables are available:

Name	T	R	W	Description
TRACK	i!		40	On write: Start or stop tracking on a target. Possible values are: **0** Stop tracking **1** (Re)start tracking (this is also needed, if a new object has been selected) **2** Go to object coordinates. The telescope will move to the current object coordinates and stay there. This will end a running a tracking. **3** Update filter wheel position to the currently configured one. This will not affect a running tracking. **4** Update active M2 position to the currently configured one (including the correction from the M2 model). This will not affect a running tracking. **5** Update derotator position to the currently configured one. This is only possible, if `POINTING.SETUP.DEROTATOR.SYNCMODE` was 0 at begin of tracking or no tracking is active. **6** Update dome position to the currently configured one. This is only possible, if `POINTING.SETUP.DOME.SYNCMODE` was 0 at begin of tracking or no tracking is active. **7** Update instrument selector position to the currently configured one. This will not affect a running tracking. **8** Update correction for active M1 model. This will not affect a running tracking. If a tracking is running and the setup optical port (`POINTING.SETUP.USE_PORT`) does not match the one from the running tracking, the actions 3, 4, 5 and 7 will fail as they would affect not the currently active optical port. On read: Return current tracking status of telescope. Possible values are: **0** Telescope is not tracking **1** Telescope is tracking (or slewing to target)

Name	T	R	W	Description
`ORIENTATION`	i!		-1	Used tracking orientation: **0** Normal pointing (e.g.(e.g. `ZD` $\geq 0\,°$, `DEC`, `ZD[1]` $\leq 90\,°$) **1** Reverse pointing (e.g.(e.g. `ZD` $< 0\,°$, `DEC`, `ZD[1]` $> 90\,°$) Whether reverse tracking is available depends on the hardware (see section 4.2.11). Also, the `POINTING.SETUP` module allows to activate or deactivate this feature (see section 6.1).
`DEROTATOR_OFFSET`	f!		-1	Offset for the derotator (in relative mode) or derotator position (in absolute mode). This will include both user settings and automatic settings (if this optimization is selected (see section 6.1.5)). [°]
`FOCUS_OFFSET`	f!		-1	Automatically selected offset for the focus (composed from active M2 model, port, filter, instrument and active M1 model offset, see section 6.1.4) [mm]
`SLEWTIME`	f!		-1	Time needed to reach the target once tracking is started [s]
`TRACKTIME`	f!		-1	Time available for tracking the target before mount limits will be reached [s]
`TRACKLIMITS`	s!		-1	Comma separated list of limits that will be reached eventually: **<*axis*>`_PosMin`** Axis will hit minimum position limit **<*axis*>`_PosMax`** Axis will hit maximum position limit **<*axis*>`_SpeedMax`** Axis will go over maximum speed limit **`TRAJECTORY_EndOfData`** User-defined trajectory data will end **`OBJECT_BelowHorizon`** Object will go below horizon **`OBJECT_Invisible`** Object is invisible at current geographic location For <*axis*>, the same names as in `POSITION.INSTRUMENTAL` will be used (see section 8.2).
`TARGETDISTANCE`	f!		-1	The RMS distance of all axes from the target [°]

Name	T	R	W	Description
GLOBAL_DATA	i!		50	On write: Switch between a connection-local storage (see section 3.5) and a global storage for the data from the OBJECT and POINTING.SETUP module. Possible values are: **0** Use private (e.g. connection-local) data **1** Use global data SET will return when the data area has been switched. On read: Return currently active data area
COPY_CURRENT	i!	-1	50	Copy the object that is currently tracked into the OBJECT and/or POINTING.SETUP variables. Possible values are: **1** Copy OBJECT data **2** Copy POINTING.SETUP data **3** Copy OBJECT and POINTING.SETUP data
SETUP	M			Configuration of the tracking/slewing mode (see section 6.1)
TRAJECTORY	M			Readout of points of the trajectory that the object will make in different coordinate systems (see section 6.2)

6.1. The POINTING.SETUP submodule

This module allows the configuration of how the target should be acquired. Most of the settings will however be ignored if OBJECT.TYPE is 1.

Name	T	R	W	Description
REFRACTION	i!		50	Consider atmospheric refraction (refer to appendix C for a description of the different refraction types) . Possible values are: **0** No refraction correction **1** Use refraction correction. The temperature and pressure will be taken from the TELESCOPE.CONFIG.ENVIRONMENT module (see section 4.2.2) **101. . .** Use a custom refraction correction. Temperature and pressure (see above) might be used in this case as well. If and how many different custom refraction modes are available can be checked by using the OpenTPL !MAX property.

Name	T	R	W	Description
ORIENTATION	i!		50	Desired orientation for tracking: **0** Normal pointing (e.g. ZD $\geq 0°$, DEC, ZD[1] $\leq 90°$)) **1** Reverse pointing (e.g. ZD $< 0°$, DEC, ZD[1] $> 90°$). If selected and the telescope hardware does not support reverse pointing (see section 4.2) tracking will fail. **2** Allow automatic selection
OPTIMIZATION	i!		50	Optimization mode for slewing to/tracking of objects: **0** No optimization. An extended azimuth/derotator range will not be used. **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. **2** Minimize slew time. As above, all features will be used, to allow fastest acquisition of the target from the current telescope position. The optimization will also consider MIN_TRACKTIME (see below).
MIN_TRACKTIME	f!		50	The minimal required track time. This will be considered, if OPTIMIZATION is used. If the selected target cannot be tracked at least for the given time, tracking will fail. [s]
USE_PORT	i!		50	Selection of optical port (refer to [1] for a list of defined ports). This will determine the used derotator, filter wheel and instrument selector and will also rotate M3 (if supported by the telescope hardware).
LOCAL	M			Location and time settings (see section 6.1.1)
ENVIRONMENT	M			Environmental conditions setup (see section 6.1.2)
SIDEROSTAT	M			Siderostat specific options (see section 6.1.3)
FOCUS	M			Focus specific options (see section 6.1.4)
DEROTATOR	M			Derotator specific options (see section 6.1.5)
FILTER	M			Filter wheel specific options (see section 6.1.6)
INSTRUMENT	M			Instrument specific options (see section 6.1.7)
DOME	M			Dome specific options (see section 6.1.8)

6.1.1. The POINTING.SETUP.LOCAL submodule

The global location and time settings from TELESCOPE.CONFIG.LOCAL (see section 4.2.1) can be overridden here, which can be used to simulate previous conditions, e.g. for adding old pointing model measurements (see section 4.3.1.0.1).

Name	T	R	W	Description
SYNCMODE	i		50	Source for telescope location and time information: **0** Use custom values (see below) **1** Use global values from TELESCOPE.CONFIG.LOCAL Writing to any of the variables below will set this variable to 0.
LATITUDE	f		50	Latitude of telescope site, positive for northern hemisphere [°]
LONGITUDE	f		50	Longitude of telescope site, positive east of Greenwich [°]
HEIGHT	f		50	Height of telescope site above sea level [m]
UT1-UTC	f		50	Difference between UT1 and UTC [s]
TAI-UTC	f		50	Difference between TAI and UTC [s]

6.1.2. The POINTING.SETUP.ENVIRONMENT submodule

The global environmental settings from TELESCOPE.CONFIG.ENVIRONMENT (see section 4.2.2) can be overridden here, which can be used to simulate previous conditions, e.g. for adding old pointing model measurements (see section 4.3.1.0.1).

Name	T	R	W	Description
SYNCMODE	i		50	Source for environmental data: **0** Use custom values (see below) **1** Use global values from TELESCOPE.CONFIG.ENVIRONMENT Writing to any of the variables below will set this variable to 0.
TEMPERATURE	f		50	Temperature [°C]
PRESSURE	f		50	Air pressure [mBar]

6.1.3. The POINTING.SETUP.SIDEROSTAT submodule

Allows configuration of siderostat related settings.

Name	T	R	W	Description
AZ_CORRECTION	f!		50	Azimuth correction of the port selection. If the port selection has a remaining error, this can be corrected here. [°]

6.1.4. The POINTING.SETUP.FOCUS submodule

Allows configuration of focus related settings.

Name	T	R	W	Description
SYNCMODE	i!		50	Synchronize focus movement to various conditions. This variable is bit coded (bits not listed below are reserved and need to be zero): **Bit 0** Synchronize with focus position (see below) **Bit 1** reserved **Bit 2** Synchronize with port specific offset **Bit 3** Synchronize with filter specific offset **Bit 4** Turn off focus motor during tracking **Bit 5** Synchronize with instrument specific offset **Bit 6** Synchronize with offset from active M1 model (see section 4.2.5.1)
POSITION	f!		50	Focus position, taken into account if Bit 0 of SYNCMODE is set [mm]

6.1.5. The POINTING.SETUP.DEROTATOR submodule

Allows configuration of derotator related settings (if available on the used port).

Name	T	R	W	Description
SYNCMODE	i!		50	Synchronize derotator movement. Possible values: **0** Do not synchronize **1** Absolute position, given as OFFSET (or DEROTATOR for trajectories) **2** True orientation **3** As 2, but use OFFSET (or DEROTATOR for trajectories) as an additional offset **4** True orientation plus a 0, 90, 180 or 270 degree offset (to allow for a longer tracking time or shorter slewing time) **5** As 4, but use OFFSET (or DEROTATOR for trajectories) as an additional offset **6** True orientation plus arbitrary offset (for even better optimization)
OFFSET	f!		50	Derotator position (SYNCMODE 1) or additional offset on calculated position (SYNCMODE 3 and 5). The counting direction is always the same as the position angle (not the instrumental counting direction of the derotator). The pointing model of the derotator will be after this value has been used/added. [°]

6.1.6. The POINTING.SETUP.FILTER submodule

Allows the selection of a filter (if a filter wheel is available on the used port).

Name	T	R	W	Description
INDEX	i!		50	On write: Selection of filter by index On read: Index of currently selected filter
NAME	s!		50	On write: Selection of filter by name On read: Name of currently selected filter

6.1.7. The POINTING.SETUP.INSTRUMENT submodule

Allows the selection of the used instrument (if a instrument selector is available on the used port).

Name	T	R	W	Description
INDEX	i!		50	On write: Selection of instrument by index On read: Index of currently selected instrument
NAME	s!		50	On write: Selection of instrument by name On read: Name of currently selected instrument

6.1.8. The POINTING.SETUP.DOME submodule

Allows the configuration of dome related settings.

Name	T	R	W	Description
SYNCMODE	i!		50	Synchronize dome movements with telescope.This variable is bit coded (bits not listed below are reserved and need to be zero): **Bit 0** Use OFFSET either as fixed dome position (if none of the next two bits are set) or as an additional offset. **Bit 1** Synchronize discrete (allow a position deviation from the calculated position of up to TELESCOPE.CONFIG.DOME.MAX_DEVIATION (see section 4.2.10) before moving the dome again) **Bit 2** Synchronize continuous (may not be suited for all domes) **Bit 3** Use MAX_DEVIATION instead of the global value, when synchronizing discrete If no bits are set, the dome will not be moved automatically during tracking.
MAX_DEVIATION	f!		50	For discrete dome synchronization (SYNCMODE Bit 1), the globally configured maximum position deviation can be overridden (SYNCMODE Bit 3). [°]

Name	T	R	W	Description
OFFSET	f!		50	Dome position or position offset (taken into account when SYNCMODE Bit 0 is set). The dome model will be applied after this value has been used/added. [°]

6.2. The POINTING.TRAJECTORY submodule

With this module, the path (trajectory) of the object can be retrieved. Depending on the submodule used, the path can be acquired in different coordinate systems:

Name	T	R	W	Description
STARTTIME	f!		50	Start time of the requested path segment. A value of NULL will be replaced by the current time. [UTC seconds since 01.01.1970 00:00:00]
STEPSIZE	f!		50	Time between two points of the path segment [s]
HORIZONTAL[]	M[]			Access object path in horizontal coordinates (see section 6.3)
EQUATORIAL[]	M[]			Access object path in equatorial coordinates (see section 6.4)

6.3. The POINTING.TRAJECTORY.HORIZONTAL[] submodule

Each element of this module array represents one point of the object trajectory. Its coordinates are available in the horizontal coordinate system.

Name	T	R	W	Description
UTC	f!		-1	Time of the trajectory point [UTC seconds since 01.01.1970 00:00:00]
AZ	f!		-1	Azimuth in true horizontal coordinates [°]
ALT	f!		-1	Elevation/Altitude in true horizontal coordinates [°]
REFRACTION	f!		-1	Altitude offset due to atmospheric refraction (calculated as currently selected in POINTING.SETUP.REFRACTION), adding it to ALT will yield the refracted altitude [°]
DEROTATOR	f!		-1	Derotator position [°]
DOME	f!		-1	Dome position [°]

6.4. The POINTING.TRAJECTORY.EQUATORIAL[] submodule

Each element of this module array represents one point of the object trajectory. Its coordinates are available in the equatorial coordinate system, both in the current equinox and in the J2000.0 reference system.

Name	T	R	W	Description
UTC	f!		-1	Time of the trajectory point [UTC seconds since 01.01.1970 00:00:00]
RA_J2000	f!		-1	Right ascension in the J2000.0 reference system [h]
DEC_J2000	f!		-1	Declination in the J2000.0 reference system [°]
RA_CURRENT	f!		-1	Right ascension in current equinox [h]
DEC_CURRENT	f!		-1	Declination in current equinox [°]

7. The CURRENT module

This module provides information on the currently running tracking, giving access to the exact object definition and the setup used for this tracking. Additionally, it offers current information on tracking limits and remaining tracking time etc.. Many variables are identical in meaning to those discussed in sections 5 and 6, but allow only reading.

Name	T	R	W	Description
TRACK	i		-1	Current tracking status: **0** Telescope currently not tracking > 0 Telescope currently tracking or slewing to target. The returned number will change always (and only) if a new tracking is started, uniquely identifying the currently running tracking.
ORIENTATION	i		-1	Used tracking orientation: **0** Normal pointing (e.g., ZD ≥ 0) **1** Reverse pointing (e.g., ZD < 0) If reverse tracking is available depends on the hardware (see section 4.2). Also, the POINTING.SETUP module allows to activate or deactivate this feature.
DEROTATOR_OFFSET	f		-1	Offset for the derotator (in relative mode) or derotator position (in absolute mode). This will include both user settings and automatic settings (if this optimization is selected (see section 6.1)). [°]
FOCUS_OFFSET	f		-1	Automatically selected offset for the focus (composed from active M2 model, port, filter, instrument and active M1 model offset) [mm]
SLEWTIME	f		-1	Remaining time needed to reach the target [s]
TRACKTIME	f		-1	Minimum remaining time available for tracking the target before mount limits will be reached [s]

Name	T	R	W	Description
TRACKLIMITS	s		-1	Comma separated list of limits that will be reached eventually: **<*axis*>_PosMin** Axis will hit minimum position limit **<*axis*>_PosMax** Axis will hit maximum position limit **<*axis*>_SpeedMax** Axis will go over maximum speed limit **TRAJECTORY_EndOfData** User-defined trajectory data will end **OBJECT_BelowHorizon** Object will go below horizon **OBJECT_Invisible** Object is invisible at current geographic location For <*AXIS*>, the same names as in POSITION.INSTRUMENTAL will be used.
TARGETDISTANCE	f		-1	The RMS distance of all axes from the target [°]
OBJECT	M			Information on the currently tracked object. Refer to section 5 for a list of variables and their description. In this module, all variables are read-only and no TRAJECTORY sub-modules exist. It is possible to retrieve INSTRUMENTAL, HORIZONTAL and EQUATORIAL coordinates also for objects with different type. For TRAJECTORY objects, the current position on the trajectory will be returned.
SETUP	M			Information on the setup of the current tracking. Refer to section 6.1 for a list of variables and their description. In this module, all variables are read-only.
TRAJECTORY	M			Readout of points of the trajectory that the currently tracked object will make in different coordinate systems. This module has the same entries as POINTING.TRAJECTORY (see section 6.2)

8. The POSITION module

This module provides the telescope position (including geographic position and time) in several coordinate systems.

Name	T	R	W	Description
LOCAL	M			Time and position information (see section 8.1)
INSTRUMENTAL	M			Instrumental coordinates (see section 8.2)
HORIZONTAL	M			Horizontal coordinates (see section 8.3)
EQUATORIAL	M			Equatorial coordinates (see section 8.4)

8.1. The POSITION.LOCAL submodule

This module gives access to the geographic position of the telescope site and the current time.

Name	T	R	W	Description
SIDEREAL_TIME	f		-1	Current local sidereal time [h]
UTC	f		-1	Current telescope time in UTC [seconds since 01.01.1970 00:00:00]
UT1	f		-1	Current telescope time in UT1 [seconds since 01.01.1970 00:00:00]
TAI	f		-1	Current telescope time in TAI [seconds since 01.01.1970 00:00:00]
UT1-UTC	f		-1	Difference between UT1 and UTC [s]
TAI-UTC	f		-1	Difference between TAI and UTC [s]
LATITUDE	f		-1	Latitude of telescope site, positive for northern hemisphere [°]
LONGITUDE	f		-1	Longitude of telescope site, positive east of Greenwich [°]
HEIGHT	f		-1	Height of telescope site above sea level [m]

8.2. The POSITION.INSTRUMENTAL submodule

The modules gives access to position, offset and status of many telescope axes. Refer to [1] for a detailed description of the OpenTCI axes, their variables and their coordinate systems. The following telescope axes can be available, depending on the actual telescope hardware. Depending on the telescope mount type (TELESCOPE.CONFIG.MOUNTTYPE, see section 4.2), the available main telescope axes differ. The correlation between main axes and mount type is listed in the table below.

Name	T	R	W	Description
FOCUS[[]]	M[[]]			Focus (can be an array in case of a hexapod system)
COVER	M			Main mirror cover
PORT_SELECT	M			Port selection (e.g. turnable M3)
PORT_COVER[]	M[]			Port cover
DEROTATOR[]	M[]			Derotators
FILTER[]	M[]			Filter wheels
INST_SELECT[]	M[]			Instrument selectors
AO_BENDER[]	M[]			Active optics actuators
DOME[]	M[]			Astrodome
				Only for mount type 1
AZ	M			Azimuth
ZD	M			Zenith distance
				Only for mount type 2
HA	M			Hour angle
DEC	M			Declination
				Only for mount type 3
ZD[]	M			Zenith distance, array of two axes

PORT_COVER[], DEROTATOR[], FILTER[] and INST_SELECT[] are arrays and the index corresponds to the optical port number where the devices is mounted on. FOCUS can be an array if a hexapod system is used. The `DOME[]` array will consist of all dome axes. Please refer to [1] for the port numbering, numbers of the hexapod axes and dome axes.

For all these axes, the following information is provided:

Name	T	R	W	Description
`POWER_STATE`	f		-1	Power state of the component. Possible values are: **-1.0** Emergency stop **0.0** Off **>0.0 . . . <1.0** Power state changing **1.0** On
`REFERENCED`	f		-1	Referenced state of this component. Possible values are: **0.0** Not referenced **>0.0 . . . <1.0** Axis is currently referencing **1.0** Referenced
`ERROR_STATE`	i		-1	If this variable is $\neq 0$, then the axis is currently having errors, possibly preventing proper operation. The severity is bit coded (bits not listed below are reserved and need to be zero): **Bit 0** *PANIC*: Module does not work anymore; the error is so severe that the whole telescope cannot operate anymore **Bit 1** *ERROR*: Module has errors and is not working **Bit 2** *WARNING*: Module is still working but performance may be influenced **Bit 3** *INFO*: Module is working; some information on the status have been saved **Bit 4** *DEBUG*: Module is working; some debug information is available

Name	T	R	W	Description
LIMIT_STATE	i		-1	If this variable is $\neq 0$, then the axis is currently at some limit. The limits are bit coded (bits not listed below are reserved and need to be zero): **Bit 0** Minimum position (hardware limit) **Bit 1** Maximum position (hardware limit) **Bit 7** Hardware limit is blocking current movement **Bit 8** Minimum position (software limit) **Bit 9** Maximum position (software limit) **Bit 15** Software limit is blocking current movement
MOTION_STATE	i		-1	If this variable is $\neq 0$, then the axis is currently moving. The exact movement situation is bit coded (bits not listed below are reserved and need to be zero): **Bit 0** Axis is moving **Bit 1** Trajectory is running **Bit 2** Movement is blocked for some reason (e.g. by limit switch, see also LIMIT_STATE) **Bit 3** Axis has acquired current axis target position or position deviation during trajectory execution is below threshold **Bit 4** Axis movement is limited by maximum speed, acceleration and/or jerk. This bit indicates for instance that trajectory points are to quickly changing the axis target position **Bit 5** Axis is "unparked", i.e. is configured to and has been moved on its startup position by TELESCOPE.READY=1 or TELESCOPE.PARK=0 **Bit 6** Axis is "parked", i.e. is configured to and has been moved on its park position by TELESCOPE.READY=0 or TELESCOPE.PARK=1
REALPOS[1]	f		-1	True current position in component specific units. Refer to [1] for an explanation of the coordinate systems. With the properties !MIN and !MAX the valid position range can be queried. If either of the properties is NULL this will indicate that there is no limit for the position in that direction. E.g. the azimuth of a telescope might rotate infinitely.

[1] The relationship between CURRPOS, REALPOS and OFFSET is as follows: CURRPOS+OFFSET=REALPOS.

Name	T	R	W	Description
CURRPOS[1]	f		-1	The current position, not including the OFFSET (in the same units as REALPOS).
CURRSPEED	f		-1	The current speed of the axis. With the property !MAX, the maximum speed of the axis can be read. [units/s]
TARGETPOS[2]	f		20	The position the axis is currently moving to, not including the OFFSET (in the same units as REALPOS). While a trajectory is executed, the variable will constantly be updated to the current position on the trajectory. By writing to this variable, the axis will switch into direct positioning mode, canceling a possibly running trajectory. SET will return after the position has been reached (i. e. bit 3 of MOTION_STATE is set), the target was reset by another request or an error occurred, so the execution of this command may take a while. Meanwhile it is possible to set another target position. However this will terminate any previous TARGETPOS commands. Note that setting this variable for axes the OpenTSI is currently using for a running tracking will terminate the tracking.
OFFSET[1,2]	f		40	Additional offset (in the same units as REALPOS) that will be added to *all* positioning requests (both direct positioning and trajectories) for that component. This can be used for a guiding system and for centering the stars when building a pointing model. SET will only return after the component has reached the given offset (i. e. bit 3 of MOTION_STATE is set), another offset was requested or an error occurred. Meanwhile it is possible to set a different offset. Again, this will terminate previous offset commands.
TARGETDISTANCE	f		-1	The distance between CURRPOS and TARGETPOS (in the same units as REALPOS), also valid during trajectory execution.

8.3. The POSITION.HORIZONTAL submodule

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

Name	T	R	W	Description
AZ	f		-1	Azimuth in true horizontal coordinates [°]
ALT	f		-1	Elevation/Altitude in true horizontal coordinates [°]
ZD	f		-1	Zenith distance in true horizontal coordinates [°]
AIR_MASS	f		-1	Air mass (calculated according to Rozenberg (1966) for sea level elevation, taking refraction into account)

[2] The relationship between TARGETPOS, CURRPOS, REALPOS and OFFSET is as follows: On target holds TARGETPOS=CURRPOS and TARGETPOS+OFFSET=REALPOS

Name	T	R	W	Description
REFRACTION	f		-1	Altitude offset due to atmospheric refraction (calculated as currently selected in POINTING.SETUP.REFRACTION), adding it to ALT will yield the refracted altitude [°]
DEROTATOR	f		-1	Derotator position [°]
DOME	f		-1	Dome position [°]

8.4. The POSITION.EQUATORIAL submodule

Provides the telescope position in equatorial coordinates.

Name	T	R	W	Description
RA_J2000	f		-1	Right ascension in the J2000.0 reference system [h]
DEC_J2000	f		-1	Declination in the J2000.0 reference system [°]
RA_CURRENT	f		-1	Right ascension in current equinox [h]
DEC_CURRENT	f		-1	Declination in current equinox [°]
PARALLACTIC_ANGLE	f		-1	Parallactic angle [°]
POSITION_ANGLE	f		-1	Position angle [°]

9. The AUXILIARY module

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

Name	T	R	W	Description
EXECUTE	s	0	0	On write: Execute custom command (implementation specific) On read: Status of last executed command
COVER	M			Main mirror covers (see section 9.1)
PORT_COVER[]	M			Optical port covers (see section 9.2)
DOME	M			Astrodome (see section 9.3)
PADDLE	M			Manual control unit (see section 9.4)
SENSOR	M			Telescope sensors (see section 9.5)
SWITCH	M			Telescope switches (see section 9.6)

9.1. The AUXILIARY.COVER submodule

This module provides functions to move the mirror covers.

Name	T	R	W	Description
REALPOS	f		-1	Position of the cover: **0** Closed **1** Open
TARGETPOS	f		30	Target position of the cover (see above) SET will return once the position has been reached.

9.2. The AUXILIARY.PORT_COVER[] submodule

This module array provides functions to move covers on the optical ports, the array index specifies the port number.

Name	T	R	W	Description
REALPOS	f		-1	Position of the port cover: **0** Closed **1** Open
TARGETPOS	f		30	Target position of the port cover (see above) SET will return once the position has been reached.

9.3. The AUXILIARY.DOME submodule

This module provides functions for opening and closing the dome. The rotation of the dome (if supported) is handled by the POINTING module (see section 6.1.8). If direct access to the individual dome axes is needed, POSITION.INSTRUMENTAL.DOME[] can be used (it is however not recommended to do so).

Name	T	R	W	Description
REALPOS	f		-1	Position of the dome enclosure(s): **0** Closed **>0.0 ... <1.0** Dome is moving and/or some of the enclosure axes are open, others are closed. **1** Open
TARGETPOS	f		30	Target position of the dome enclosure(s): **0** Close all axes **1** Fully open enclosure/roof/slit/flap **>1** Bit coded value of axes that should be open (bits not listed below are reserved and need to be zero): **Bit 1** Fold enclosure/roof part 1/slit **Bit 2** Roof part 2/flap **Bit 4** Ventilation flaps
OPEN_MASK	f		-1	The currently fully open dome axes as bit coded value. See TARGETPOS for the meaning of the individual bits.
CLOSED_MASK	f		-1	The currently fully closed dome axes as bit coded value. See TARGETPOS for the meaning of the individual bits.

9.4. The `AUXILIARY.PADDLE` submodule

This module provides access to the manual control unit of the telescope.

Name	T	R	W	Description
`ACTIVE`	i		30	Manual control software activation state (current state on read, new state on write)[3]. Possible values: **0** Disabled **1** Enabled
`BRIGHTNESS`	f		30	Brightness of buttons. Possible values range from zero (off) to one (brightest setting).
`SPEED`	f		-1	Setting of the speed dial on the manual control unit in relative units from zero to one, with one representing the maximum axis speed
`MODE`	i		-1	Mode of the manual control unit (e.g. state of the key switch on the unit)[3]. Possible values are: **0** Disabled **1** Enabled
`SELECTION`	i		-1	A number representing a selection (e.g. selected device) that was made with the manual control unit, usually corresponding to the number on a selector switch
`SELECTION_TEXT`	s		-1	The OpenTCI module name(s) of the selected axis or axes as a comma-separated list[4]: <*module*>[,<*module*>[,...]] (e.g. `AZ,DEROTATOR[2]`)

9.5. The `AUXILIARY.SENSOR[]` submodule

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

Name	T	R	W	Description
`DESCRIPTION`	s		-1	A textual description of the sensor position
`UNIT`	s		-1	The unit of the sensor device like Pa, °C etc.
`VALUE`	f		-1	Sensor value in the specified unit

9.6. The `AUXILIARY.SWITCH[]` submodule

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

[3] The actual state of the manual control unit is an "AND" conjunction of `MODE` and `ACTIVE`, e.g. it needs to be activated both in software and by its key switch to be operational.

[4] This can be used to visualize the layout of the manual control unit. Therefore, also an empty string can be returned for one axis to show that it currently has no selection, e.g. `,DEROTATOR[2]`.

Name	T	R	W	Description
DESCRIPTION	s		-1	A textual description of the switch position
UNIT	s		-1	The unit of the switch device like Pa, °C etc.
VALUE	f		30	Switch value in the specified unit

A. Quick guide for the impatient

- 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 5 and 6).
- 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 4).
- 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 a manual control unit, these will be accessible in the `AUXILIARY` module (see section 9).

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. Altazimuth mounted telescopes (`MOUNTTYPE 1`)

B.1.1. The `CLASSIC` model

The classic model consists of offsets for both telescope axes (`AOFF`, `ZOFF`) and additionally of corrections that take geometric errors of the mount into account. An error in the leveling of the mount towards north (`AN`) and towards east (`AE`), a non-perpendicularity of the azimuth and zenith distance axis (`NPAZ`), a non-perpendicularity of the zenith distance and optical axis (`NPZO`) and a sagging of the telescope tube (`TF`). The table below gives a systematic overview on all coefficients.

Coefficient	Description
AOFF	Azimuth offset (+: clockwise) [°]
ZOFF	Zenith distance offset (+: towards horizon) [°]
AN	Tilt of azimuth axis (+: towards north) [°]
AE	Tilt of azimuth axis (+: towards east) [°]

Coefficient	Description
NPAZ	Error in perpendicularity of azimuth and zenith distance axis [°]
NPZO	Error in perpendicularity of zenith distance and optical axis [°]
TF	Sagging of tube [°]

The formulas for the calculation of the correction offsets for this model are as follows:

$$
\begin{aligned}
\Delta_{\mathtt{AZ}} &= c_{\mathtt{AOFF}} \\
&- c_{\mathtt{AN}} \cdot \sin A \cdot \cot Z \\
&+ c_{\mathtt{AE}} \cdot \cos A \cdot \cot Z \\
&+ c_{\mathtt{NPAZ}} \cdot \cot Z \\
&+ c_{\mathtt{NPZO}} \cdot \csc Z \\
\Delta_{\mathtt{ZD}} &= c_{\mathtt{ZOFF}} \\
&+ c_{\mathtt{AN}} \cdot \cos A \\
&+ c_{\mathtt{AE}} \cdot \sin A \\
&+ c_{\mathtt{TF}} \cdot \sin Z
\end{aligned}
$$

B.1.2. The EXTENDED model

The extended model includes, in addition to the classic model (see appendix B.1.1), four coefficients that take the eccentricity of the encoders into account (AES, AEC for azimuth, ZES, ZEC for zenith distance). The constant TF of the classic model is thereby included in ZEC, so there are only three new coefficients. Furthermore the extended model adds several empirical 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 (AAN, AAE) and zenith distance (ZAN, ZAE). The table below gives a systematic overview on all coefficients.

Coefficient	Description
AOFF	Azimuth offset (+: clockwise) [°]
ZOFF	Zenith distance offset (+: towards horizon) [°]
AAN	Tilt of azimuth axis (+: towards north) (correction for azimuth) [°]
ZAN	Tilt of azimuth axis (+: towards north) (correction for zenith distance) [°]
AAE	Tilt of azimuth axis (+: towards east) (correction for azimuth) [°]
ZAE	Tilt of azimuth axis (+: towards east) (correction for zenith distance [°]
NPAZ	Error in perpendicularity of azimuth and zenith distance axis [°]
NPZO	Error in perpendicularity of zenith distance and optical axis [°]
AES	Eccentricity of azimuth encoder, sine part [°]
AEC	Eccentricity of azimuth encoder, cosine part [°]
ZES	Eccentricity of zenith distance encoder, sine part [°]
ZEC	Eccentricity of zenith distance encoder, cosine part (this coefficient also includes the sagging of the tube, TF in classic model) [°]
AS2A	Empirical correction of azimuth due to 2× azimuth, sine part [°]
AC2A	Empirical correction of azimuth due to 2× azimuth, cosine part [°]
AS3A	Empirical correction of azimuth due to 3× azimuth, sine part [°]

Coefficient	Description
AC3A	Empirical correction of azimuth due to 3× azimuth, cosine part [°]
ZS2A	Empirical correction of zenith distance due to 2× azimuth, sine part [°]
ZC2A	Empirical correction of zenith distance due to 2× azimuth, cosine part [°]
ZS3A	Empirical correction of zenith distance due to 3× azimuth, sine part [°]
ZC3A	Empirical correction of zenith distance due to 3× azimuth, cosine part [°]
ZS4A	Empirical correction of zenith distance due to 4× azimuth, sine part [°]
ZC4A	Empirical correction of zenith distance due to 4× azimuth, cosine part [°]
ZCSZ	Empirical correction of zenith distance due to cosecant of zenith distance [°]

The formulas for the calculation of the correction offsets for this model are as follows:

$$
\begin{aligned}
\Delta_{\mathtt{AZ}} &= c_{\mathtt{AOFF}} \\
&- c_{\mathtt{AAN}} \cdot \sin A \cdot \cot Z \\
&+ c_{\mathtt{AAE}} \cdot \cos A \cdot \cot Z \\
&+ c_{\mathtt{NPAZ}} \cdot \cot Z \\
&+ c_{\mathtt{NPZO}} \cdot \csc Z \\
&+ c_{\mathtt{AES}} \cdot \sin A \\
&+ c_{\mathtt{AEC}} \cdot \cos A \\
&+ c_{\mathtt{AS2A}} \cdot \sin(2A) \cdot \cot Z \\
&+ c_{\mathtt{AC2A}} \cdot \cos(2A) \cdot \cot Z \\
&+ c_{\mathtt{AS3A}} \cdot \sin(3A) \cdot \cot Z \\
&+ c_{\mathtt{AC3A}} \cdot \cos(3A) \cdot \cot Z \\
\Delta_{\mathtt{ZD}} &= c_{\mathtt{ZOFF}} \\
&+ c_{\mathtt{ZAN}} \cdot \cos A \\
&+ c_{\mathtt{ZAE}} \cdot \sin A \\
&+ c_{\mathtt{ZES}} \cdot \sin Z \\
&+ c_{\mathtt{ZEC}} \cdot \cos Z \\
&+ c_{\mathtt{ZS2A}} \cdot \sin(2A) \\
&+ c_{\mathtt{ZC2A}} \cdot \cos(2A) \\
&+ c_{\mathtt{ZS3A}} \cdot \sin(3A) \\
&+ c_{\mathtt{ZC3A}} \cdot \cos(3A) \\
&+ c_{\mathtt{ZS4A}} \cdot \sin(4A) \\
&+ c_{\mathtt{ZC4A}} \cdot \cos(4A) \\
&+ c_{\mathtt{ZCSZ}} \cdot \csc Z
\end{aligned}
$$

B.2. Equatorial mounted telescopes (`MOUNTTYPE 2`)

B.2.1. The `CLASSIC` model

The classic model consists of offsets for both mount axes (`HOFF`, `DOFF`) and additionally of corrections that take geometric errors of the telescope into account. An error of the polar axis alignment in azimuth (PA) and elevation (PE), a non-perpendicularity of the hour angle and declination axis (NPHD), a non-perpendicularity of the declination and optical axis (NPDO) and a sagging of the tube (TF). The table below gives a systematic overview on all coefficients.

Coefficient	Description
HOFF	Hour angle offset [°]
DOFF	Declination offset [°]
PA	Polar axis misalignment in azimuth (+: towards east (northern hemisphere)/west (southern hemisphere)) [°]
PE	Polar axis misalignment in elevation (+: towards horizon (northern hemisphere)/zenith (southern hemisphere)) [°]
NPHD	Error in perpendicularity of hour angle and declination axis [°]
NPDO	Error in perpendicularity of declination and optical axis [°]
TF	Sagging of tube [°]

The formulas for the calculation of the correction offsets for this model are as follows (with Φ being the telescope's site latitude):

$$
\begin{aligned}
\Delta_{\mathtt{HA}} &= c_{\mathtt{HOFF}} \\
&- c_{\mathtt{PA}} \cdot \cos H \cdot \tan D \\
&+ c_{\mathtt{PE}} \cdot \sin H \cdot \tan D \\
&+ c_{\mathtt{NPHD}} \cdot \tan D \\
&+ c_{\mathtt{NPDO}} \cdot \sec D \\
&+ c_{\mathtt{TF}} \cdot \cos \Phi \cdot \sin H \cdot \sec D \\
\Delta_{\mathtt{DEC}} &= c_{\mathtt{DOFF}} \\
&+ c_{\mathtt{PA}} \cdot \sin H \\
&+ c_{\mathtt{PE}} \cdot \cos H \\
&+ c_{\mathtt{TF}} \cdot (\cos \Phi \cdot \cos H \cdot \sin D - \sin \Phi \cdot \cos D)
\end{aligned}
$$

B.2.2. The EXTENDED model

The extended model includes, in addition to the classic model (see appendix B.2.1), four coefficients that take the eccentricity of the encoders into account (HES, HEC for hour angle, DES, DEC for declination). Furthermore, terms for mount flexure (MF) and position-dependent non-perpendicularity of hour angle and declination axis (DNPHD) are added. The table below gives a systematic overview on all coefficients.

Coefficient	Description
HOFF	Hour angle offset [°]
DOFF	Declination offset [°]
PA	Polar axis misalignment in azimuth (+: towards east (northern hemisphere)/west (southern hemisphere)) [°]
PE	Polar axis misalignment in elevation (+: towards horizon (northern hemisphere)/zenith (southern hemisphere)) [°]
NPHD	Error in perpendicularity of hour angle and declination axis [°]

Coefficient	Description
DNPHD	Error in perpendicularity of hour angle and declination axis, dependent on hour angle [°]
NPDO	Error in perpendicularity of declination and optical axis [°]
TF	Sagging of tube [°]
HES	Eccentricity of hour angle encoder, sine part [°]
HEC	Eccentricity of hour angle encoder, cosine part [°]
DES	Eccentricity of declination encoder, sine part [°]
DEC	Eccentricity of declination encoder, cosine part [°]

The formulas for the calculation of the correction offsets for this model are as follows (with Φ being the telescope's site latitude):

$$
\begin{aligned}
\Delta_{\mathtt{HA}} &= c_{\mathtt{HOFF}} \\
&- c_{\mathtt{PA}} \cdot \cos H \cdot \tan D \\
&+ c_{\mathtt{PE}} \cdot \sin H \cdot \tan D \\
&+ c_{\mathtt{NPHD}} \cdot \tan D \\
&+ c_{\mathtt{DNPHD}} \cdot \sin H \cdot \tan D \\
&+ c_{\mathtt{NPDO}} \cdot \sec D \\
&+ c_{\mathtt{TF}} \cdot \cos \Phi \cdot \sin H \cdot \sec D \\
&+ c_{\mathtt{HES}} \cdot \sin H \\
&+ c_{\mathtt{HEC}} \cdot \cos H \\
\Delta_{\mathtt{DEC}} &= c_{\mathtt{DOFF}} \\
&+ c_{\mathtt{PA}} \cdot \sin H \\
&+ c_{\mathtt{PE}} \cdot \cos H \\
&+ c_{\mathtt{TF}} \cdot (\cos \Phi \cdot \cos H \cdot \sin D - \sin \Phi \cdot \cos D) \\
&+ c_{\mathtt{DES}} \cdot \sin D \\
&+ c_{\mathtt{DEC}} \cdot \cos D
\end{aligned}
$$

B.3. Alt-Alt mounted telescopes (MOUNTTYPE 3)

B.3.1. The CLASSIC model

The classic model consists of offsets for both mount axes (Z0OFF, Z1OFF) and additionally of corrections that take geometric errors of the mount into account. An error of the mount alignment in azimuth (MA) and elevation (ME), a non-perpendicularity of the two mount axes (NPZZ), a non-perpendicularity of the upper zenith distance axis and optical axis (NPZO) and a sagging of the tube (TF). The table below gives a systematic overview on all coefficients.

Coefficient	Description
Z0OFF	Zenith distance 0 offset [°]
Z1OFF	Zenith distance 1 offset [°]
MA	Mount orientation misalignment in azimuth (+: clockwise) [°]
ME	Mount orientation misalignment in elevation (+: towards horizon) [°]

Coefficient	Description
NPZZ	Error in perpendicularity of zenith distance 0 and zenith distance 1 axis [°]
NPZO	Error in perpendicularity of zenith distance 1 and optical axis [°]
TF	Sagging of tube [°]

The formulas for the calculation of the correction offsets for this model are as follows:

$$
\begin{aligned}
\Delta_{\mathtt{Z0}} &= c_{\mathtt{Z0OFF}} \\
&- c_{\mathtt{MA}} \cdot \cos Z_0 \cdot \tan Z_1 \\
&+ c_{\mathtt{ME}} \cdot \sin Z_0 \cdot \tan Z_1 \\
&+ c_{\mathtt{NPZZ}} \cdot \tan Z_1 \\
&+ c_{\mathtt{NPZO}} \cdot \sec Z_1 \\
&+ c_{\mathtt{TF}} \cdot \sin Z_0 \cdot \sec Z_1 \\
\Delta_{\mathtt{Z1}} &= c_{\mathtt{Z1OFF}} \\
&+ c_{\mathtt{MA}} \cdot \sin Z_0 \\
&+ c_{\mathtt{ME}} \cdot \cos Z_0 \\
&+ c_{\mathtt{TF}} \cdot \cos Z_0 \cdot \sin Z_1
\end{aligned}
$$

B.3.2. The EXTENDED model

The extended model includes, in addition to the classic model (see appendix B.3.1), four coefficients that take the eccentricity of the encoders into account (Z0ES, Z0EC for zenith distance 0 (lower), Z1ES, Z1EC for zenith distance 1 (upper)). These also correct for the mount flexure that needed an extra term in equatorial mounts. Furthermore, a term for position-dependent non-perpendicularity of the mount axes (DNPZZ) is added. The table below gives a systematic overview on all coefficients.

Coefficient	Description
Z0OFF	Zenith distance 0 offset [°]
Z1OFF	Zenith distance 1 offset [°]
MA	Mount orientation misalignment in azimuth (+: clockwise) [°]
ME	Mount orientation misalignment in elevation (+: towards horizon) [°]
NPZZ	Error in perpendicularity of zenith distance 0 and zenith distance 1 axis [°]
DNPZZ	Error in perpendicularity of zenith distance 0 and zenith distance 1 axis, dependent on zenith distance 0 [°]
NPZO	Error in perpendicularity of zenith distance 1 and optical axis [°]
TF	Sagging of tube [°]
Z0ES	Eccentricity of zenith distance 0 encoder, sine part [°]
Z0EC	Eccentricity of zenith distance 0 encoder, cosine part [°]
Z1ES	Eccentricity of zenith distance 1 encoder, sine part [°]
Z1EC	Eccentricity of zenith distance 1 encoder, cosine part [°]

The formulas for the calculation of the correction offsets for this model are as follows:

$$
\begin{aligned}
\Delta_{\texttt{Z0}} &= c_{\texttt{Z0OFF}} \\
&- c_{\texttt{MA}} \cdot \cos Z_0 \cdot \tan Z_1 \\
&+ c_{\texttt{ME}} \cdot \sin Z_0 \cdot \tan Z_1 \\
&+ c_{\texttt{NPZZ}} \cdot \tan Z_1 \\
&+ c_{\texttt{DNPZZ}} \cdot \sin Z_0 \cdot \tan Z_1 \\
&+ c_{\texttt{NPZO}} \cdot \sec Z_1 \\
&+ c_{\texttt{TF}} \cdot \sin Z_0 \cdot \sec Z_1 \\
&+ c_{\texttt{Z0ES}} \cdot \sin Z_0 \\
&+ c_{\texttt{Z0EC}} \cdot \cos Z_0 \\
\Delta_{\texttt{Z1}} &= c_{\texttt{Z1OFF}} \\
&+ c_{\texttt{MA}} \cdot \sin Z_0 \\
&+ c_{\texttt{ME}} \cdot \cos Z_0 \\
&+ c_{\texttt{TF}} \cdot \cos Z_0 \cdot \sin Z_1 \\
&+ c_{\texttt{Z1ES}} \cdot \sin Z_1 \\
&+ c_{\texttt{Z1EC}} \cdot \cos Z_1
\end{aligned}
$$

B.4. Corrections for other axes

B.4.1. Active M1

Only one model type (`SAMPLE`) is defined, which uses directly the correction of the nearest measurement.

B.4.2. Active M2

B.4.2.1. The `Z_T` model The OpenTSI includes a simple thermal focus correction which is based on the telescope 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 OpenTCI ordering of temperature sensors (see [1]) which can be assumed fixed unless the hardware configuration is updated.

Then the temperature dependent focus offset will be calculated using the following formula:

$$
\Delta_{\texttt{FOCUS}} = \sum_{i=0}^{N-1} \left(c_{\texttt{A2\_}i} \vartheta_i^2 + c_{\texttt{A1\_}i} \vartheta_i \right) + c_{\texttt{A0}}
$$

B.4.2.2. The `XYZUV_GRAV_T` model An extended model for active M2 systems can be selected. It uses a gravitational correction together with a thermal correction with linear and quadratic terms.

For the thermal correction, the values T_i of the N temperature sensors of the telescope are combined linearly by applying an individual weight $c_{\texttt{T\_}<i>}$ on them (the order of the sensors is based on the OpenTCI ordering of temperature sensors (see [1]) which can be assumed fixed unless the hardware configuration is updated):

$$T_{comb} = \sum_{i=0}^{N-1} (c_{\texttt{T\_}<i>} \cdot T_i)$$

From the instrumental telescope position, the direction of the gravitation is calculated.

The position of each axis of the active M2 system ($axis \in \{X, Y, Z, U, V\}$) is then calculated using individual coefficients for the three components of the gravitation and the linear and quadratic temperature influence:

$$\begin{aligned} P_{<axis>} &= c_{<axis>\texttt{\_C}} \\ &+ c_{<axis>\texttt{\_GX}} \cdot G_x \\ &+ c_{<axis>\texttt{\_GY}} \cdot G_y \\ &+ c_{<axis>\texttt{\_GZ}} \cdot G_z \\ &+ c_{<axis>\texttt{\_T1}} \cdot T_{comb} \\ &+ c_{<axis>\texttt{\_T2}} \cdot T_{comb}^2 \end{aligned}$$

The set of coefficients are:

Coefficient	Description
<*axis*>_C	Constant offset [mm for X, Y and Z / ° for U and V]
<*axis*>_GX	Component of gravitation along x [mm for X, Y and Z / ° for U and V]
<*axis*>_GY	Component of gravitation along y [mm for X, Y and Z / ° for U and V]
<*axis*>_GZ	Component of gravitation along z [mm for X, Y and Z / ° for U and V]
<*axis*>_T1	Linear thermal correction [mm for X, Y and Z / ° for U and V]
<*axis*>_T2	Quadratic thermal correction [mm for X, Y and Z / ° for U and V]
T_<*i*>	Weight of temperature sensor with index i.

B.4.3. Derotator

The derotator position is corrected by an axis offset (DOFF) and terms for the encoder eccentricity (DES, DEC).

Coefficient	Description
DOFF	Derotator offset [°]
DES	Eccentricity of derotator encoder, sine part [°]
DEC	Eccentricity of derotator encoder, cosine part [°]

The formula for the calculation of the correction offset for the derotator are as follows:

$$\begin{aligned} \Delta_{\texttt{DEROTATOR}} &= c_{\texttt{DOFF}} \\ &+ c_{\texttt{DES}} \cdot \sin D \end{aligned}$$

$$+ \quad c_{\mathtt{DEC}} \cdot \cos D$$

B.4.4. Dome

The dome position is corrected by an axis offset in azimuth and zenith distance (e.g. windshield) (`CAOFF`, `CZOFF`). Additionally, a parallax correction for the dome for non-fork mounts is performed (denoted as function f below), which takes the dome diameter into account and for equatorial mounts also the site latitude.

Coefficient	Description
`CAOFF`	Astrodome azimuth offset[°]
`CZOFF`	Astrodome zenith distance offset[°]
`CDMCN`	Distance of mount center (intersection of mount axes) from astrodome center towards north [m]
`CDMCE`	Distance of mount center (intersection of mount axes) from astrodome center towards east [m]
`CDMCZ`	Distance of mount center (intersection of mount axes) from astrodome center towards zenith [m]
`CDOMCA`	Distance of optical axis from mount center (intersection of mount axes) in direction of second mount axis [m]
`CDOMCP`	Distance of optical axis from mount center (intersection of mount axes) in direction perpendicular to second mount axis [m]

The formulas for the calculation of the correction offsets for the dome axes are as follows (with Φ being the telescope's site latitude for equatorial mounts, or 90° for altazimuth resp. 0° for alt-alt mounted telescopes):

$$\begin{aligned}
\Delta_{\mathtt{DOME[0]}} &= c_{\mathtt{CAOFF}} \\
&+ f_{\mathrm{Dome,Az}}(A_0, A_1, \Phi, c_{\mathtt{CDMCN}}, c_{\mathtt{CDMCE}}, c_{\mathtt{CDMCZ}}, c_{\mathtt{CDOMCA}}, c_{\mathtt{CDOMCP}}, d_{\mathrm{Dome}}) \\
\Delta_{\mathtt{DOME[5]}} &= c_{\mathtt{CZOFF}} \\
&+ f_{\mathrm{Dome,ZD}}(A_0, A_1, \Phi, c_{\mathtt{CDMCN}}, c_{\mathtt{CDMCE}}, c_{\mathtt{CDMCZ}}, c_{\mathtt{CDOMCA}}, c_{\mathtt{CDOMCP}}, d_{\mathrm{Dome}})
\end{aligned}$$

C. Refraction types

The built-in correction for atmospheric refraction uses the method proposed by Bennett. The refraction R_{sealevel} (in arc minutes) is calculated for the apparent altitude h_a in degrees):

$$R_{\mathrm{sealevel}} = \frac{1\,\mathrm{arcmin}}{\tan\left(h_a + \frac{7.31°^2}{h_a + 4.4°}\right)}$$

The apparent altitude needs to be calculated iteratively with the above formula from the true altitude.

A conversion must be applied to correct for the local pressure P and local temperature T (in kelvin):

$$R = R_{\text{sealevel}} * \frac{P}{1010\,\text{mPa}} * \frac{283\,\text{K}}{T}$$

D. Events

This paragraph will be completed in the future.

E. Sample communication

The following sample communication explains how to accomplish some basic tasks using the OpenTSI. Data the client receives from the server is marked with "⇐" and data the client sends to the server is marked with "⇒". (Even though the command id could be reused once the respective command has been completed, this is not done here for clarity. For more details on the OpenTPL protocol, please refer to [2])

- Check the power state of the telescope (see section 4) — the OpenTSI server states here that the telescope is currently not powered:

  ⇒ `101 GET TELESCOPE.READY_STATE`

  ⇐ `101 COMMAND OK`

  ⇐ `101 DATA INLINE TELESCOPE.READY_STATE=0.0`

  ⇐ `101 COMMAND COMPLETE`

- Request to power up the telescope (see section 4) — the command returns successfully when the telescope was powered up without problems. This may take a while. In the meantime `TELESCOPE.READY_STATE` and other variables can be read for ongoing status information.

  ⇒ `102 SET TELESCOPE.READY=1.0`

  ⇐ `102 COMMAND OK`

  ⇐ `102 DATA OK TELESCOPE.READY`

  ⇐ `102 COMMAND COMPLETE`

- Prepare the telescope to track an object (see section 6.1) — the sent options depend on the installed telescope hardware, e.g. if there is a derotator (see section 6.1.5) or filter wheel (see section 6.1.6) installed, more `POINTING.SETUP` variables should be set beforehand. Check if all commands were reported as successfully set (`DATA OK` was received for each value). To detect if certain hardware is available on the telescope, the variables in `TELESCOPE.CONFIG` (see section 4.2) can be used.

  ⇒ `201 SET POINTING.SETUP.REFRACTION=1`

  ⇒ `202 SET POINTING.SETUP.ORIENTATION=2`

  ⇒ `203 SET POINTING.SETUP.OPTIMIZATION=1`

  ⇒ `204 SET POINTING.SETUP.MIN_TRACKTIME=1800.0`

  ⇒ `205 SET POINTING.SETUP.USE_PORT=3`

⇒ 206 SET POINTING.SETUP.LOCAL.SYNCMODE=1

⇒ 207 SET POINTING.SETUP.ENVIRONMENT.SYNCMODE=1

⇒ 208 SET POINTING.SETUP.FOCUS.SYNCMODE=111

⇒ 209 SET POINTING.SETUP.FOCUS.POSITION=1.25

⇐ 201 COMMAND OK

⇐ 201 DATA OK POINTING.SETUP.REFRACTION

⇐ 201 COMMAND COMPLETE

⇐ ...

Set the information about the tracked target, in this example of the type equatorial (see section 5.5). Other object types can be selected as well (see section 5 for an overview) — check if all commands were reported as successfully set:

⇒ 211 SET OBJECT.EQUATORIAL.NAME="Arcturus"

⇒ 212 SET OBJECT.EQUATORIAL.RA=14.26

⇒ 213 SET OBJECT.EQUATORIAL.DEC=19.18

⇒ 214 SET OBJECT.EQUATORIAL.RA_PM=-2.02E-5

⇒ 215 SET OBJECT.EQUATORIAL.DEC_PM=-5.55E-4

⇒ 216 SET OBJECT.EQUATORIAL.RA_RATE=0.0

⇒ 217 SET OBJECT.EQUATORIAL.DEC_RATE=0.0

⇒ 218 SET OBJECT.EQUATORIAL.RATE_START=NULL

⇒ 219 SET OBJECT.EQUATORIAL.EPOCH=1991.25

⇒ 220 SET OBJECT.EQUATORIAL.EQUINOX=2000.0

⇒ 221 SET OBJECT.DIFFERENTIAL.TYPE=0

⇐ 211 COMMAND OK

⇐ 211 DATA OK OBJECT.EQUATORIAL.NAME

⇐ 211 COMMAND COMPLETE

⇐ ...

Check if the set up target can be tracked — the track-time should be at least the value set for POINTING.SETUP.MIN_TRACKTIME (see section 6.1), POINTING.TRACKLIMITS (see section 6) helps to find the cause for the tracking end.

⇒ 231 GET POINTING.TRACKTIME;POINTING.TRACKLIMITS

⇐ 231 COMMAND OK

⇐ 231 DATA INLINE POINTING.TRACKTIME=3600.0

⇐ 231 DATA INLINE POINTING.TRACKLIMITS=""

⇐ 231 COMMAND COMPLETE

Start tracking the set up target (see section 6), the command returns immediately:

⇒ 232 SET POINTING.TRACK=1

⇐ 232 COMMAND OK

⇐ 232 DATA OK POINTING.TRACK

⇐ 232 COMMAND COMPLETE

Check if the tracking is still running. To get more information on the currently running tracking, the CURRENT object can be used (see section 7).

⇒ 233 GET POINTING.TRACK

⇐ 233 COMMAND OK

⇐ 233 DATA INLINE POINTING.TRACK=1

⇐ 233 COMMAND COMPLETE

When the new tracking is started, check the progress of the slewing — use TELESCOPE.MOTION_STATE (see section 4) to check for the end of the slewing process.

⇒ 234 GET POINTING.TARGETDISTANCE;TELESCOPE.MOTION_STATE

⇐ 234 COMMAND OK

⇐ 234 DATA INLINE POINTING.TARGETDISTANCE=0.00168

⇐ 234 DATA INLINE TELESCOPE.MOTION_STATE=11

⇐ 234 COMMAND COMPLETE

- Apply an instrumental offset to center a target in the view/camera (see section 8.2), in case of bad pointing and to prepare a new pointing model measurement. Check for success of the sent commands.

  ⇒ 301 SET POSITION.INSTRUMENTAL.AZ.OFFSET=-0.0012

  ⇒ 302 SET POSITION.INSTRUMENTAL.ZD.OFFSET=0.0015

  ⇐ 301 COMMAND OK

  ⇐ 301 DATA OK POSITION.INSTRUMENTAL.AZ.OFFSET

  ⇐ 301 COMMAND COMPLETE

  ⇐ ...

- Check regularily if the telescope reports any events (see section 4.4) — here for the ZD axis an event is reported. After the cause is resolved (in this case: a too large offset moving the axis into the software limit), the event can be cleared.

  ⇒ 401 GET TELESCOPE.STATUS.LIST;TELESCOPE.STATUS.GLOBAL

  ⇐ 401 COMMAND OK

  ⇐ 401 DATA INLINE TELESCOPE.STATUS.LIST=→
  "DRIVES|8:ZD|8:ERR_Soft_Limit_min||8|ZD,SYSTEM|0::,AUXILIARY|0::,UNKNOWN|0::"

  ⇐ 401 DATA INLINE TELESCOPE.STATUS.GLOBAL=8

  ⇐ 401 COMMAND COMPLETE

  Clear the reported events by writing the value from TELESCOPE.STATUS.GLOBAL (it could also be derived from TELESCOPE.STATUS.LIST, see section 4.4) to the appropriate clear variable — check for success of the event clearing command, and check also the event list again (see above), as the event may reappear if the cause was not removed successfully.

⇒ `402 SET TELESCOPE.STATUS.CLEAR_INFO=8`

⇐ `402 COMMAND OK`

⇐ `402 DATA OK TELESCOPE.STATUS.CLEAR_INFO`

⇐ `402 COMMAND COMPLETE`

- Park the telescope and switch off power (see section 4) — the telescope acknowledges the request and the command returns after all shutdown actions are finished.

  ⇒ `501 SET TELESCOPE.READY=0.0`

  ⇐ `501 COMMAND OK`

  ⇐ `501 DATA OK TELESCOPE.READY`

  ⇐ `501 COMMAND COMPLETE`

References

[1] M. Ruder, M. Velten, and D. Plasa. *OpenTCI, Open Telescope Control Interface — An open specification of an OpenTPL based interface to provide generic, low-level control of telescopes.* tau-tec GmbH, Hintere Grabenstr. 30, 72070 Tübingen, Germany, 2012. `tclm-tci:spec-en`.

[2] M. Ruder, M. Velten, and D. Plasa. *OpenTPL, Open Transfer Protocol Language — A protocol for client-server based exchange of data and commands over a TCP/IP network connection.* tau-tec GmbH, Hintere Grabenstr. 30, 72070 Tübingen, Germany, 2012. `opentpl:spec-en`.