Pilar-TSI

Telescope Software Interface

Specifications of the Interface to the high level Telescope Software as provided by Pilar 2 (Pilar-TSI)

Version 1.1

Document Id:

4PI-DOC-03-008-06-1

Michael Ruder, Peter Kroll

Revision 49 as of 20th August 2004 (michael)

Revision History:

08.11.2003	Initial Version (mr,pk)
04.03.2004	Modifications (mr)
27.07.2004	Updates (filter wheel, flip mirror and cover added) (mr)
03.08.2004	Updates (optimization modes, full pointing model) (pk)

$Id: Pilar-TSI.tex 49 2004-08-20 14:45:24Z michael$

Contents

1 Preface

This document covers the interface to the high level telescope software as provided by Pilar 2. This interface will eventually be replaced by the "Telescope Software Interface (TSI)" [4], which is currently under development. It is therefore to be considered as a temporary solution only to provide a set of advanced functions such as pointing and tracking in equatorial coordinates, calculation and application of a pointing model etc. as this is not covered by the TCI ("Telescope Control Interface" [2]).
Since the final TSI will provide a much more general, hardware independent approach and will include a broader set of functions, the both interfaces cannot be assumed to be compatible.
For the communication with both the current Pilar-TSI and the final TSI, the same protocol as for the TCI is used ("Transfer Protocol Language, V2" [3]).
The graphical user interface (GUI) of Pilar is not covered in this document. It is however described in detail in [1].

2 Basics

- The TPL-Server of Pilar does not understand the full TPL2 syntax as described in [3]. It only understands inline assignments (it is however possible to do multiple assignments in one command) and it only returns data in the `DATA INLINE` format. Also, properties are not known and there is no support for authentication, encryption, different privilege levels and accessing modules and variables by their number. All connections are authenticated implicitly.
- At the moment, it is not possible to connect to the TCI through the TSI. To access variables of the TCI which are not provided by the Pilar-TSI, a separate connection to the TCI has to be opened.
- All coordinates are to be given in degrees unless otherwise noted. The right ascension is to be given in hours. All equatorial coordinates refer to the equinox J2000.0.

3 List of Modules

The following modules are currently supported by the Pilar-TSI.

Name	Type	Access	Description
`PILAR`	MODULE	—	Control of Pilar specific functions, f.e. the display of the GUI.

Name	Type	Access	Description
LOC	MODULE	—	Location specific settings of the telescope.
ENV	MODULE	—	Environment specific settings.
FILTER	MODULE	—	Control of the filter wheel.
FOCUS	MODULE	—	Control of the focus system.
MIRROR	MODULE	—	Control of the flip mirror.
COVER	MODULE	—	Control of the mirror covers.
POINTING	MODULE	—	All pointing related functions.
TCC	MODULE	—	Direct information from the TCC.

3.1 Module PILAR

This module allows control over some pilar settings. At the moment it is used to control whether the GUI is shown or not.

Name	Type	Access	Description
GUI	MODULE	—	Control of the Pilar GUI.
VERSION	STRING	RO	Version of Pilar.
VERSION_DATE	STRING	RO	Date when this version was compiled.
LICENSE	STRING	RO	License information.

3.1.1 Submodule PILAR.GUI

This module allows control of the Pilar GUI.

Name	Type	Access	Description
SHOW	INT	WO	Change the state of the Pilar GUI: 1=show Pilar GUI, 0=hide Pilar GUI. *Attention:* If the Pilar GUI is shown, the user at the TSC may disable/enable the TSI server and may also disable/enable the write access of the remote control. A more detailed explanation of the functions of the Pilar GUI can be found in [1].
SHOW_STATE	INT	RO	Check state of the Pilar GUI: 1=Pilar GUI is currently shown, 0=GUI is hidden.

3.2 Module LOC

In this module, all location dependent settings are grouped together.

Name	Type	Access	Description
`LONG`	FLOAT	RW	Returns the currently active longitude or sets it [°]. Positive west of greenwich.
`LAT`	FLOAT	RW	Returns the currently active latitude for calculations or sets it [°]. Positive for northern hemisphere.
`HEIGHT`	FLOAT	RW	Returns the currently active height or sets it [m].
`TIME`	FLOAT	RO	If `USE_GPS_TIME` is 0, UTC time from the TSC in seconds since 1970 (millisecond resolution) otherwise the same `LOC.GPS.TIME`.
`SIDERIAL_TIME`	FLOAT	RO	Sidereal time at the currently active position (see above) [h].
`GPS`	MODULE	—	Access to the coordinates and time measured by the GPS system of the TCC.
`USE_GPS_LONG`	INT	RW	Use the longitude measured by the GPS system (=1), or the one set with `LOC.LONG` (=0).
`USE_GPS_LAT`	INT	RW	Use the latitude measured by the GPS system (=1) or the one set with `LOC.LAT` (=0).
`USE_GPS_HEIGHT`	INT	RW	Use the height measured by the GPS system (=1), or the one set with `LOC.HEIGHT` (=0). *Attention:* The TCC GPS receiver might not return a valid height information!

3.2.1 Submodule LOC.GPS

This module contains the data as it is obtained from the GPS receiver.

Name	Type	Access	Description
`LONG`	FLOAT	RO	The geographic longitude, as measured by the GPS system [°]. Positive west of greenwich.
`LAT`	FLOAT	RO	The geographic latitude, as measured by the GPS system [°]. Positive for northern hemisphere.
`HEIGHT`	FLOAT	RO	The geographic height, as measured by the GPS system (not always available, see [2]) [m].

Name	Type	Access	Description
TIME	FLOAT	RO	Precise UTC time source from the TCC (usually synchronized via GPS) in seconds since 1970 (millisecond resolution).

3.3 Module ENV

With this module it is possible to set weather dependent data which is required for the correct calculation of the refraction.

Name	Type	Access	Description
TEMPERATURE	FLOAT	RW	Current temperature, used for the refraction model [°C].
PRESSURE	FLOAT	RW	Current barometric pressure, used for the refraction model [Pa].

3.4 Module FILTER

This module allows to control the filter wheel.

Name	Type	Access	Description
POWER	INT	WO	Turn on (=1) or turn off (=0) the power to the filter wheel drive.
POWER_STATE	INT	RO	Check if the filter wheel is turned on (=1) or turned off (=0).
TARGET.POS	FLOAT	RW	Move the filter wheel to the specific filter holder index (integer value) [index]. For this command to work, the filter wheel has to be powered on.
CURRENT.POS	FLOAT	RO	Readout the index of the currently selected filter [index].

3.5 Module FOCUS

This module allows to control the focus motor.

Name	Type	Access	Description
POWER	INT	WO	Turn on (=1) or turn off (=0) the power to the focus system.
POWER_STATE	INT	RO	Check if the focus system is turned on (=1) or turned off (=0).

Name	Type	Access	Description
`TARGET.POS`	FLOAT	RW	Move the focus to the specified position [mm], smaller values mean that M2 is closer to M1. For this command to work, the focus has to be powered on.
`CURRENT.POS`	FLOAT	RO	Readout the current focus position [mm].
`OFFSET.POS`	FLOAT	RW	Set or returns the additional position offset for the focus system [mm]. The focus will only start to move to the new position, if it is powered on.

3.6 Module MIRROR

This module allows to control the flip mirror.

Name	Type	Access	Description
`POWER`	INT	WO	Turn on (=1) or turn off (=0) the power to the flip mirror drive.
`POWER_STATE`	INT	RO	Check if the flip mirror drive is turned on (=1) or turned off (=0).
`TARGET.POS`	FLOAT	RW	Move the flip mirror to the specified position in degrees. Zero specifies the primary nasmyth focus [°]. For this command to work, the flip mirror has to be powered on.
`CURRENT.POS`	FLOAT	RO	Readout the current flip mirror position [°].

3.7 Module COVER

This module allows to control the mirror covers.

Name	Type	Access	Description
`POWER`	INT	WO	Turn on (=1) or turn off (=0) the power to the cover motors.
`POWER_STATE`	INT	RO	Check if the cover motors are turned on (=1) or turned off (=0).
`TARGET.POS`	FLOAT	RW	Close the covers (=1) or open them (=0). For this command to work, the cover motors have to be powered on.

Name	Type	Access	Description
`CURRENT.POS`	FLOAT	RO	Check if the covers are closed (=1) or open (=0). Some hardware may return positions between 0 and 1 to indicate the progress.

3.8 Module POINTING

In this module, all functions for pointing, tracking and calculation and application of pointing models are included.

Name	Type	Access	Description
`HOME`	MODULE	—	Defines the telescope home position.
`OFFSET`	MODULE	—	Sets or reads the offset on the main telescope axes (azimuth, altitude and derotator).
`CURRENT`	MODULE	—	Reads the current position of the telescope axes in different coordinate systems.
`TARGET`	MODULE	—	Sets or reads the coordinates of the desired target in equatorial coordinates.
`GO`	MODULE	—	Starts/stops tracking and allows movement to the home position.
`MODEL`	MODULE	—	Activates the pointing model and also allows the calculation of a new model.

3.8.1 Submodule POINTING.HOME

It is possible to define a special home position for the telescope. By setting `POINTING.GO.HOME` to 1, the telescope will then move to that position.

Name	Type	Access	Description
`AZ`	FLOAT	RW	Azimuth home position [°] (the telescope will move to that position if `POINTING.GO.HOME` is set to 1)
`ZD`	FLOAT	RW	Zenith distance home position [°] (the telescope will move to that position if `POINTING.GO.HOME` is set to 1)
`DEROT`	FLOAT	RW	Derotator home position [°] (the telescope will move to that position if `POINTING.GO.HOME` is set to 1)

3.8.2 Submodule POINTING.OFFSET

All axes can have an offset which is applied to all telescope positions. This can be used to center a star during pointing measurements or to allow a position updates from a guiding system.

Name	Type	Access	Description
AZ	FLOAT	RW	Offset in azimuth [°] (which is always added to the telescope position, also during tracking).
ALT	FLOAT	RW	Offset in altitude [°] (which is always added to the telescope position, also during tracking).
DEROT	FLOAT	RW	Offset for the derotator [°] (which is always added to the telescope position, also during tracking).

3.8.3 Submodule POINTING.CURRENT

This module allows the readout of the telescope position in different coordinate systems.

Name	Type	Access	Description
TARGET_RA	FLOAT	RO	Right ascension of the currently selected target [h].
TARGET_DEC	FLOAT	RO	Declination of the currently selected target [°].
RA	FLOAT	RO	Right ascension of the position the telescope is currently pointing at [h].
DEC	FLOAT	RO	Declination of the position the telescope is currently pointing at [°].
PA	FLOAT	RO	Position angle for the current telescope position [°].
DEROT	FLOAT	RO	Current derotator angle [°].

Name	Type	Access	Description
DEROT_PRE_OFFSET	FLOAT	RO	Offset between the theoretical derotator angle (calculated from position angle and altitude) and the used derotator angle [°]. Since the derotator may not be able to move the whole needed 540 degrees, it may be neccessary to set this offset to be able to track and derotate some positions correctly. If the angle is set to any other value than zero, the image will of course be rotated by that angle in comparison to reference images.
DEROT_OFF	INT	RO	Current mode of the derotator: 0=derotator will move during tracking, 1=derotator stays at the position defined by POINTING.TARGET.DEROT
DEROT_OPT	INT	RO	Current use of derotator optimization (on=1, off=0). Optimization means that a derotator offset of 0° or ±90° or ±180° is automatically added to the derotator angle for the next chosen target in order to achieve minimum slewing time of the telescope. DEROT_OPT indicates if the current target was pointed to with or without such an optimization.
DEROT_OPT_MIN_... ...TRACKING_TIME	FLOAT	RO	Current minimum tracking time in [min] for the derotator optimization. When automatically choosing the derotator optimization angle (provided DEROT_OPT = 1) there may be ambiguities. In general the different possible derotator offsets differ from each other in their available tracking time. DEROT_OPT_MIN_TRACKING_TIME gives the minimum tracking time which has to be guaranteed by a possible derotator offset of the current target. Remaining ambiguities are resolved by choosing the derotator offset which means the lowest slewing time.

Name	Type	Access	Description
FRONT	INT	RO	Current orientation of the telescope tube. Normal orientation is given by 1, reverse orientation by 0.
FRONT_AUTO	INT	RO	Current setting of the automatic choosing of the tube orientation. If 0, the orientation was not chosen automatically, if 1 it was. This means, that the telescope uses the normal (front) orientation whenever possible. If this orientation is not possible (this may occur due to an insufficient range of the azimuth axis hardware of some telescopes), the software decided to use instead the reverse orientation of the tube in order to reach the target.
HA	FLOAT	RO	Current hour angle [h].
AZ	FLOAT	RO	Azimuth coordinate of the current telescope position [°]. (This is the real azimuth, not the instrumental coordinate which will differ due to the pointing model.)
ALT	FLOAT	RO	Altitude coordinate of the current telescope position [°]. (This is the real altitude, not the instrumental coordinate which will differ due to the pointing model.)

3.8.4 Submodule POINTING.TARGET

This module is used to define a new target which should be tracked. After setting all necessary variables, the tracking can be started by setting `POINTING.GO.TRACK` to 1.

Name	Type	Access	Description
RA	FLOAT	RW	Right ascension of the next target to be pointed at [h]. To actually move the telescope, `POINTING.GO.TO` or `POINTING.GO.TRACK` needs to be set to 1.

Name	Type	Access	Description
DEC	FLOAT	RW	Declination of the next target to be pointed at [°]. To actually move the telescope, POINTING.GO.TO or POINTING.GO.TRACK needs to be set to 1.
OFFSET.RA	FLOAT	RW	Additional offset in right ascension [h]. This offset is only taken into account when POINTING.GO.TO or POINTING.GO.TRACK is set to one. So if this is changed during tracking, nothing happens until the tracking is restarted.
OFFSET.DEC	FLOAT	RW	Additional offset in declination [°]. This offset is only taken into account when POINTING.GO.TO or POINTING.GO.TRACK is set to one. So if this is changed during tracking, nothing happens until the tracking it restarted.
DEROT	FLOAT	RW	Define the derotator position for the fixed derotator mode (see below) [°].
DEROT_PRE_OFFSET	FLOAT	RW	Define an additional offset for the derotator which will be added to theoretical position, which is calculated from position angle and altitude [°]. Since the derotator cannot move the whole 540 degrees, it is neccessary to set this offset to be able to track and derotate some positions correctly. If the angle is set to any other value than zero, the image will of course be rotated by that angle in comparison to reference images.
DEROT_FIXED	INT	RW	Select derotator mode: 0=derotator will move during tracking, 1=derotator stays at the position defined by POINTING.TARGET.DEROT

Name	Type	Access	Description
DEROT_OPT	INT	RW	Switches the derotator optimization on (=1) or off (=0). Optimization means that a derotator offset of 0° or ±90° or ±180° is automatically added to the derotator angle for the next chosen target in order to achieve minimum slewing time of the telescope.
DEROT_OPT_MIN_... ...TRACKING_TIME	FLOAT	RW	Sets the minimum tracking time in [min] for the derotator optimization. When automatically choosing the derotator optimization angle (provided DEROT_OPT = 1) there may be ambiguities. In general the different possible derotator offsets differ from each other in their available tracking time. DEROT_OPT_MIN_TRACKING_TIME selects the minimum tracking time which should be guaranteed by a possible derotator offset for the next target. Remaining ambiguities are resolved by choosing the derotator offset which means the lowest slewing time.
FRONT	INT	RW	Select the orientation of the telescope tube to be either normal/front (=1) or reverse (=0).
FRONT_AUTO	INT	RW	Activates (=1) or deactivates (=0) the automatic choosing of the tube orientation. In automatic mode, the telescope uses the normal (front) orientation whenever possible. If this orientation is not possible (this may occur due to an insufficient range of the azimuth axis hardware of some telescopes), the software decides to use instead the reverse orientation of the tube in order to reach the target.
CHECK.TRACKING_TIME	FLOAT	RO	After setting the above fields, this variable will gives the approximate time the selected object can be tracked without that at least one axis runs into some limit [s].

Name	Type	Access	Description
`CHECK.TRACKING_LIMIT`	STRING	RO	After setting the above fields, this variable will return a bit mask, telling which axes will limit the time the selected object can be tracked. The bits have the following meanings (bit 0 is the lowest one): 0 = azimuth minimum, 1 = azimuth maximum, 2 = zenith distance minimum, 3 = zenith distance maximum, 4 = derotator minimum, 5 = derotator maximum.

3.8.5 Submodule POINTING.GO

This module allows to move the telescope and to start or stop the tracking as well as inquire the current status of the telescope.

Name	Type	Access	Description
`TO`	INT	WO	By setting this variable to 1, the telescope will move towards the coordinates set in `POINTING.TARGET`. It will however not track this position. *Attention:* The offset set in `POINTING.TARGET.OFFSET` will also be applied!
`TRACK`	INT	WO	By setting this variable to 1, the telescope will move towards the coordinates set in `POINTING.TARGET` and tracks on these coordinates. `POINTING.GO.TO` does not need to be set before setting this variable to 1. *Attention:* The offset set in `POINTING.TARGET.OFFSET` will also be applied!
`STOP`	INT	WO	Setting this variable to 1 will stop an possibly running tracking.
`HOME`	INT	WO	By setting this variable to 1, the telescope will move to the position defined with `POINTING.HOME`.
`STATE`	STRING	RO	Reading this variable will return the current movement state of the telescope. Possible values are: 0 (unknown), 1 (resting), 2 (slewing), 3 (tracking).

Name	Type	Access	Description
TRACKING_TIME	INT	RO	This variable can only be read while the telescope is tracking and it will return the remaining time before at least one telescope axis will run into some limit [s].
TRACKING_EXECUTE	INT	RO	If this variable is 1, the telescope is tracking or trying to track. If POINTING.GO.STATE is not 3, the telescope is still slewing to the position (POINTING.GO.STATE=2), has reached a limit or encountered a some problem (POINTING.GO.STATE=1).

3.8.6 Submodule POINTING.MODEL

This module gives access to three different pointing models (see appendix A for mathematical descriptions of each model) and allows calculation and loading of different sets of coefficients. (To define a standard pointing model which is automatically loaded when Pilar starts, the configuration files can be used. Refer to [1] for more details.)

Name	Type	Access	Description
TYPE	INT	RW	Select which pointing model type to use or read which model is currently used (0=none, 1=simple, 2=classic, 3=full).
FILE	STRING	RW	Set or get the configuration filename containing the pointing coefficients.
LOAD	INT	WO	Load pointing coefficients from the configuration file.
SAVE	INT	WO	Save currently active pointing coefficients to the configuration file.
SIMPLE	MODULE	—	Access to the coefficients of the simple pointing model.
CLASSIC	MODULE	—	Access to the coefficients of the classic pointing model.
FULL	MODULE	—	Access to the coefficients of the extended pointing model.
LIST	MODULE	—	List of measurements and functions for calculation a new model.

Submodule POINTING.MODEL.SIMPLE

The simple pointing model consists only of two offsets, one for the azimuth axis and one for altitude axis. With this model only very simple errors in the mounting of the telescope can be corrected, but it is well suited as a starting point for more advanced models.

Name	Type	Access	Description
`AOFS`	FLOAT	RW	Azimuth offset [°].
`EOFS`	FLOAT	RW	Altitude offset [°].

Submodule POINTING.MODEL.CLASSIC

This is the classic model which is known from the literature. It consists of simple geometric corrections which take into account, that the telescope might not be well leveled (AN/AE), that the altitude axis is not perpendicular to the azimuth axis (NPAE), that the optical axis is not perpendicular to the altitude axis (BNP), the offsets of the simple model (AOFS/EOFS) and that the tube will bent, depending on the altitude (TF).

Name	Type	Access	Description
`AN`	FLOAT	RW	Tilt of azimuth axis toward north [°].
`AE`	FLOAT	RW	Tilt of azimuth axis toward east [°].
`NPAE`	FLOAT	RW	Error in perpendicularity of azimuth and altitude axis [°].
`BNP`	FLOAT	RW	Error in perpendicularity of optical and altitude axis [°].
`AOFS`	FLOAT	RW	Azimuth offset [°].
`EOFS`	FLOAT	RW	Altitude offset [°].
`TF`	FLOAT	RW	Sagging of tube [°].

Submodule POINTING.MODEL.FULL

In comparison to the classic model, this pointing model has been extended by four coefficients which take into account that the encoders are not mounted precisely in the center of the axis rotation (AES/AEC for azimuth, EES/EEC for altitude) as well as by some phenomenological coefficients for higher order effects (ES2A, EC2A, ES3A, EC3A, C5). Additionally, the coefficients for azimuth tilt correction are now separate for azimuth and altitude (AAN, EAN, AAE, EAE).

Name	Type	Access	Description
Name	Type	Access	Description
`AAN`	FLOAT	RW	Tilt of azimuth axis toward north (correction for azimuth axis) [°].
`EAN`	FLOAT	RW	Tilt of azimuth axis toward north (correction for altitude axis) [°].
`AAE`	FLOAT	RW	Tilt of azimuth axis toward east (correction for azimuth axis) [°].
`EAE`	FLOAT	RW	Tilt of azimuth axis toward east (correction for altitude axis) [°].
`NPAE`	FLOAT	RW	Error in perpendicularity of azimuth and altitude axis [°].
`BNP`	FLOAT	RW	Error in perpendicularity of optical and altitude axis [°].
`AOFS`	FLOAT	RW	Azimuth offset [°].
`EOFS`	FLOAT	RW	Altitude offset [°].
`AES`	FLOAT	RW	Eccentricity of azimuth encoder, sine part [°].
`AEC`	FLOAT	RW	Eccentricity of azimuth encoder, cosine part [°].
`EES`	FLOAT	RW	Eccentricity of altitude encoder, sine part [°].
`EEC`	FLOAT	RW	Eccentricity of altitude encoder, cosine part (this coefficient also includes the sagging of the tube) [°].
`ES2A`	FLOAT	RW	Phenomenological correction of altitude due to double azimuth argument, sine part [°].
`EC2A`	FLOAT	RW	Phenomenological correction of altitude due to double azimuth argument, cosine part [°].
`ES3A`	FLOAT	RW	Phenomenological correction of altitude due to triple azimuth argument, sine part [°].
`EC3A`	FLOAT	RW	Phenomenological correction of altitude due to triple azimuth argument, cosine part [°].
`C5`	FLOAT	RW	Phenomenological correction of altitude due to inverse cosine of azimuth [°].

Submodule POINTING.MODEL.LIST

With this module, measurements for a new pointing model can be acquired and the calculation of the different pointing models can be performed. Furthermore, the newly acquired pointing model can be activated.

Name	Type	Access	Description
`ADD`	INT	WO	Add a new measurement. By writing a 1 to this variable, the current real azimuth and altitude coordinates will be saved together with the current offsets (including both the offset due to the currently active pointing model as well any user applied offset) of azimuth and altitude.
`N`	INT	RO	Returns the number of already acquired measurements.
`CLEAR`	INT	WO	Writing a 1 to this variable will clear the entire list of measurements.
`FILE`	STRING	RW	The filename from which or to which the measurements (not the coefficients) will be loaded resp. saved. A default filename can be set in the configuration files (see [1]).
`LOAD`	INT	WO	Writing a 1 to this variable will read all measurements from the defined file and adds them to the measurements already in memory.
`SAVE`	INT	WO	Writing a 1 to this variable will save the current list of measurements.
`SIMPLE`	MODULE	—	Access to calculation functions and new coefficients of the simple pointing model.
`CLASSIC`	MODULE	—	Access to calculation functions and new coefficients of the classic pointing model.
`FULL`	MODULE	—	Access to calculation functions and new coefficients of the full pointing model.

Submodule POINTING.MODEL.LIST.SIMPLE

Name	Type	Access	Description
`CALC`	INT	WO	Writing a 1 will start the calculation of the coefficients for the simple pointing model.

Name	Type	Access	Description
`APPLY`	INT	WO	The calculated pointing model will become the active model when writing a 1.
`RES`	FLOAT	RO	The total residual of pointing error for the calculated model [°].
`RES_AZ`	FLOAT	RO	The residual of azimuth pointing error [°].
`RES_ALT`	FLOAT	RO	The residual of altitude pointing error [°].
`AOFS`	FLOAT	RO	Calculated azimuth offset [°].
`EOFS`	FLOAT	RO	Calculated altitude offset [°].

Submodule POINTING.MODEL.LIST.CLASSIC

Name	Type	Access	Description
`CALC`	INT	WO	Writing a 1 will start the calculation of the coefficients for the classic pointing model.
`APPLY`	INT	WO	The calculated pointing model will become the active model when writing a 1.
`RES`	FLOAT	RO	The total residual of pointing error for the calculated model [°].
`RES_AZ`	FLOAT	RO	The residual of azimuth pointing error [°].
`RES_ALT`	FLOAT	RO	The residual of altitude pointing error [°].
`AN`	FLOAT	RO	Calculated tilt of azimuth axis toward north [°].
`AE`	FLOAT	RO	Calculated tilt of azimuth axis toward east [°].
`NPAE`	FLOAT	RO	Calculated error in perpendicularity of azimuth and altitude axis [°].
`BNP`	FLOAT	RO	Calculated error in perpendicularity of optical and altitude axis [°].
`AOFS`	FLOAT	RO	Calculated azimuth offset [°].
`EOFS`	FLOAT	RO	Calculated altitude offset [°].
`TF`	FLOAT	RO	Calculated sagging of tube [°].

Submodule POINTING.MODEL.LIST.FULL

Name	Type	Access	Description
`CALC`	INT	WO	Writing a 1 will start the calculation of the coefficients for the full pointing model.
`APPLY`	INT	WO	The calculated pointing model will become the active model when writing a 1.

Name	Type	Access	Description
RES	FLOAT	RO	The total residual of pointing error for the calculated model [°].
RES_AZ	FLOAT	RO	The residual of azimuth pointing error [°].
RES_ALT	FLOAT	RO	The residual of altitude pointing error [°].
AAN	FLOAT	RW	Calculated tilt of azimuth axis toward north (correction for azimuth axis) [°].
EAN	FLOAT	RW	Calculated tilt of azimuth axis toward north (correction for altitude axis) [°].
AAE	FLOAT	RW	Calculated tilt of azimuth axis toward east (correction for azimuth axis) [°].
EAE	FLOAT	RW	Calculated tilt of azimuth axis toward east (correction for altitude axis) [°].
NPAE	FLOAT	RW	Calculated error in perpendicularity of azimuth and altitude axis [°].
BNP	FLOAT	RW	Calculated error in perpendicularity of optical and altitude axis [°].
AOFS	FLOAT	RW	Calculated azimuth offset [°].
EOFS	FLOAT	RW	Calculated altitude offset [°].
AES	FLOAT	RW	Calculated eccentricity of azimuth encoder, sine part [°].
AEC	FLOAT	RW	Calculated eccentricity of azimuth encoder, cosine part [°].
EES	FLOAT	RW	Calculated eccentricity of altitude encoder, sine part [°].
EEC	FLOAT	RW	Calculated eccentricity of altitude encoder, cosine part (this coefficient also includes the sagging of the tube) [°].
ES2A	FLOAT	RW	Calculated phenomenological correction of altitude due to double azimuth argument, sine part [°].
EC2A	FLOAT	RW	Calculated phenomenological correction of altitude due to double azimuth argument, cosine part [°].
ES3A	FLOAT	RW	Calculated phenomenological correction of altitude due to triple azimuth argument, sine part [°].
EC3A	FLOAT	RW	Calculated phenomenological correction of altitude due to triple azimuth argument, cosine part [°].

Name	Type	Access	Description
C5	FLOAT	RW	Calculated phenomenological correction of altitude due to inverse cosine of azimuth [°].

3.9 Module TCC

This module gives direct access to the most needed functions from the TCC like power, error status, status of the referencing and the positioning limits of all axes. It also includes functions to monitor and close/establish the connection to the TCC itself.
It does, however, not provide a direct ("pass through") access to the TCC. Instead, the readonly variables are cached in Pilar and are regularly updated. Writeonly variables trigger an action inside Pilar and are immediately transferred to the TCC.

Name	Type	Access	Description
POWER	INT	WO	By setting this variable to 1, the main power is turned on. Setting it to 0 will turn off the main power.
POWER_STATE	INT	RO	Returns the current power state of the system: 0=off, 1=on. It will take a while until changes made with TCC.POWER are executed and shown in this variable.
CONNECT	INT	WO	Opens (=1) or closes (=0) the connection to the TCC.
CONNECT_STATE	INT	RO	Checks the state of the connection: 0=disconnected, 1=connected.
REFERENCED	INT	RO	Returns if all components are referenced (will only work when TCC.POWER_STATE is 1): 0=some or all unreferenced, 1=all are referenced.
ERROR_STATE	INT	RO	Returns the overall error status of the TCC: 0=OK, 1=Warning, 2=Error, 3=Panic. At least for values greater than 1 there will be some components not working. A value of 3 usually means that the entire system was halted and put into a safe condition.
ERROR_MESSAGE	STRING	RO	Returns a string with all currently active error messages, separated by ",". A description of all errors is given in [2].

Name	Type	Access	Description
ERROR_RESET	INT	WO	By writing to this variable, all currently active errors will be reset. However if the condition persists, the error will not be reset.
OFFSET_MODE	INT	RO	This variable specifies, if offsets are taken into account (=1) or not (=0) (if supported by the telescope). By settings this variable to 2, the offsets can be modified with the paddles (if those exist and are connected).
AZ.REFERENCED	INT	RO	Returns, if the azimuth axis is referenced: 0=not referenced, 1=referenced.
AZ.LIMIT.MIN	FLOAT	RO	Returns the minimum position that the azimuth axis can reach.
AZ.LIMIT.MAX	FLOAT	RO	Returns the maximum position that the azimuth axis can reach.
AZ.POS	FLOAT	RO	Returns the current position of the azimuth axis in raw telescope coordinates [°].
ZD.REFERENCED	INT	RO	Returns, if the zenith distance axis is referenced: 0=not referenced, 1=referenced.
ZD.LIMIT.MIN	FLOAT	RO	Returns the minimum position that the zenith distance axis can reach.
ZD.LIMIT.MAX	FLOAT	RO	Returns the maximum position that the zenith distance axis can reach.
ZD.POS	FLOAT	RO	Returns the current position of the zenith distance axis in raw telescope coordinates [°].
DEROT.REFERENCED	INT	RO	Returns, if the derotator is referenced: 0=not referenced, 1=referenced.
DEROT.LIMIT.MIN	FLOAT	RO	Returns the minimum position that the derotator can reach.
DEROT.LIMIT.MAX	FLOAT	RO	Returns the maximum position that the derotator can reach.
DEROT.POS	FLOAT	RO	Returns the current position of the derotator in raw telescope coordinates [°].
FILTER.REFERENCED	INT	RO	Returns, if the filter wheel is referenced: 0=not referenced, 1=referenced.
FILTER.LIMIT.MIN	FLOAT	RO	Returns the minimum position that the filter wheel can reach.

Name	Type	Access	Description
`FILTER.LIMIT.MAX`	FLOAT	RO	Returns the maximum position that the filter wheel can reach.
`FILTER.POS`	FLOAT	RO	Returns the current position of the filter wheel in raw telescope coordinates [index].
`FOCUS.REFERENCED`	INT	RO	Returns, if the focus drive is referenced: 0=not referenced, 1=referenced.
`FOCUS.LIMIT.MIN`	FLOAT	RO	Returns the minimum position that the focus drive can reach.
`FOCUS.LIMIT.MAX`	FLOAT	RO	Returns the maximum position that the focus drive can reach.
`FOCUS.POS`	FLOAT	RO	Returns the current position of the focus drive in raw telescope coordinates [mm].
`MIRROR.REFERENCED`	INT	RO	Returns, if the flip mirror is referenced: 0=not referenced, 1=referenced.
`MIRROR.LIMIT.MIN`	FLOAT	RO	Returns the minimum position that the flip mirror can reach.
`MIRROR.LIMIT.MAX`	FLOAT	RO	Returns the maximum position that the flip mirror can reach.
`MIRROR.POS`	FLOAT	RO	Returns the current position of the flip mirror in raw telescope coordinates [°].
`COVER.REFERENCED`	INT	RO	Returns, if the mirror covers are referenced: 0=not referenced, 1=referenced.
`COVER.LIMIT.MIN`	FLOAT	RO	Returns the minimum position that the mirror covers can reach.
`COVER.LIMIT.MAX`	FLOAT	RO	Returns the maximum position that the mirror covers can reach.
`COVER.POS`	FLOAT	RO	Returns the current position of the mirror covers in raw telescope coordinates [a.c.].
`SENSOR[0].VALUE`	FLOAT	RO	Value of the first (temperature/pressure/...) sensor. [depends on sensor type]
`SENSOR[1].VALUE`	FLOAT	RO	Value of the second (temperature/pressure/...) sensor. [depends on sensor type]
`SENSOR[2].VALUE`	FLOAT	RO	Value of the third (temperature/pressure/...) sensor. [depends on sensor type]
`SENSOR[3].VALUE`	FLOAT	RO	Value of the fourth (temperature/pressure/...) sensor. [depends on sensor type]
`SENSOR[4].VALUE`	FLOAT	RO	Value of the fifth (temperature/pressure/...) sensor. [depends on sensor type]
`SENSOR[5].VALUE`	FLOAT	RO	Value of the sixth (temperature/pressure/...) sensor. [depends on sensor type]

Name	Type	Access	Description
`SENSOR[6].VALUE`	FLOAT	RO	Value of the seventh (temperature/pressure/...) sensor. [depends on sensor type]
`SENSOR[7].VALUE`	FLOAT	RO	Value of the eighth (temperature/pressure/...) sensor. [depends on sensor type]

A Formulas of the different Pointing Models

The formulas in the following sections make use of the usual astronomical quantities and refer to the coordinate systems defined in [1].

A.1 The Simple Model

The simple pointing model only takes one offset per axis into account. With these offsets wrong encoder zero points and a rotation of the telescope around the azimuth axis can be corrected, so that 90 degrees altitude is in the zenith and 0 degrees of the azimuth is in the north.

$$\begin{aligned} d_{\mathrm{A}} &= c_{\mathrm{AOFS}} \\ d_{\mathrm{E}} &= c_{\mathrm{EOFS}} \end{aligned}$$

A.2 The Classic Model

The classic model takes several geometric errors of the telescopes into account and also includes the offsets of the simple model.
Additional errors corrected by this model in comparison to the simple model are: An error in the leveling of the telescope toward north (AN) and toward east (AE), a non-perpendicularity of the azimuth and elevation axis (NPAE), a non-perpendicularity of the optical and the elevation axis (BNP) and a sagging of the tube (TF).

$$\begin{aligned} d_{\mathrm{A}} &= c_{\mathrm{AN}} \cdot \sin A \cdot \tan E \\ &- c_{\mathrm{AE}} \cdot \cos A \cdot \tan E \\ &+ c_{\mathrm{NPAE}} \cdot \tan E \\ &- c_{\mathrm{BNP}} / \cos E \\ &+ c_{\mathrm{AOFS}} \\ d_{\mathrm{E}} &= c_{\mathrm{AN}} \cdot \cos A \\ &+ c_{\mathrm{AE}} \cdot \sin A \\ &+ c_{\mathrm{TF}} \cdot \cos E \end{aligned}$$

$$+ \quad c_{\mathrm{EOFS}}$$

A.3 The Full Model

The full model includes, in addition to the classic model, four coefficients that take the eccentricity of the encoders into account (AES, AEC, EES, EEC). The constants TF of the classic model is included in EEC, so there are only three new coefficients. Furthermore it adds several phenomenological coefficients (ES2A, EC2A, ES3A, EC3A, C5) for higher order effects of unknown origin. Additionally, the coefficients for azimuth tilt correction are now separate for azimuth and altitude (AAN, AAE, EAN, EAE).

$$\begin{aligned}
d_{\mathrm{A}} &= c_{\mathrm{AAN}} \cdot \sin A \cdot \tan E \\
&- c_{\mathrm{AAE}} \cdot \cos A \cdot \tan E \\
&+ c_{\mathrm{NPAE}} \cdot \tan E \\
&- c_{\mathrm{BNP}} / \cos E \\
&+ c_{\mathrm{AOFS}} \\
&+ c_{\mathrm{AES}} \cdot \sin A \\
&+ c_{\mathrm{AEC}} \cdot \cos A \\
d_{\mathrm{E}} &= c_{\mathrm{EAN}} \cdot \cos A \\
&+ c_{\mathrm{EAE}} \cdot \sin A \\
&+ c_{\mathrm{EOFS}} \\
&+ c_{\mathrm{EES}} \cdot \sin E \\
&+ c_{\mathrm{EEC}} \cdot \cos E \\
&+ c_{\mathrm{ES2A}} \cdot \sin(2A) \\
&+ c_{\mathrm{EC2A}} \cdot \cos(2A) \\
&+ c_{\mathrm{ES3A}} \cdot \sin(3A) \\
&+ c_{\mathrm{EC3A}} \cdot \cos(3A) \\
&+ c_{\mathrm{C5}} / \cos A
\end{aligned}$$

B Abbreviations

Abbreviation	Description
TPL	Transfer Protocol Language

Abbreviation	Description
TCC	Telescope Control Computer, computer in cabinet, running the TCS
TCI	Telescope Control Interface, connection between TCS and TSS
TCS	Telescope Control System, software running on TCC
TSC	Telescope Software Computer, computer close to the telescope running TSS
TSI	Telescope Software Interface, connection between TSS and user applications interface or applications
TSS	Telescope Software System, software running on TSC

References

[1] M. Ruder and P. Kroll. *Pilar 2 — A Graphical User Interface for the manual control of telescopes of "Teleskoptechnik Halfmann" company.* 4pi systeme GmbH. 4PI-DOC-03-008-07.

[2] M. Ruder and D. Plasa. *TCI, Telescope Control Interface — Specifications of the Interface to the Telescope Hardware's Control Software (TCS).* 4pi systeme GmbH. 4PI-DOC-03-008-02.

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

[4] M. Ruder, D. Plasa, and P. Kroll. *TSI, Telescope Software Interface — Specifications of the Interface between Application Programs and the Telescope Software System (TSS).* 4pi systeme GmbH. 4PI-DOC-03-008-03.