Visible & Infrared Survey Telescope for Astronomy

Project Office

Tel: +44 (0)131 668 8411
Fax: +44 (0)131 668 8412
Email: 49Hvista@roe.ac.uk,
http://www.roe.ac.uk/atc/projects/vista/

Document Title:	**Active Optics and Guiding Workstation Software Design**
Document Number:	**VIS-SPE-RAL-13030-0003**
Issue:	**2.7**
Date:	**28 March 2006**

Document Prepared By:	**David Terrett RAL Software WPM**	**Signature and Date:**	
Document Approved By:	**Malcolm Stewart Software WPM**	**Signature and Date:**	
Document Released By:	**Alistair McPherson Project Manager**	**Signature and Date:**	

Change Record

Issue	Date	Section(s) Affected	Description of Change/Change Request Reference/Remarks
1.0	2004-01-04	All	New document
2.0	2004-03-03	All	Updated after e-review
2.2	2004-04-01	Auto-guiding Control	Release 0.1 of vtagws
2.3	2004-01-06	Active optics	Release 0.1 of vtactcon & vtactcal
2.4	2005-03-02	Active optics	Corrections to ao formulae
2.6	2005-04-29	All	Update to match as-built software
2.7	2006-03-27	Active optics	Update prior to ao meeting at Netpark

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Scope

This document describes the detailed design of the components of the VISTA telescope control system associated with active optics and guiding that run on the TCS workstation. Wherever possible the design follows that of the VLT TCS in order to maximise the re-use of existing software and to make maintenance of the system (by ESO staff) easier and so this description concentrates on the areas in which the VISTA design differs from the VLT design as described in [RD01] and [RD02].

1.2 Document Structure

Section 2 gives an overview of the three workstation processes involved in the control of active optics and guiding and the commands that they will be sent to implement telescope slews, offsets and dithers.

The functions of each workstation process are then described in the three following sections, including the deployment of the processes involved, the command flows that implement the various scenarios described in [AD01] and the internal structure of the workstation processes.

The sequence diagrams that illustrate the command flows have been placed together in Section 6 to make the text more readable.

The Zernike polynomial convention used in this document is the “Noll” ordering with Z1 being “piston” as defined in Table 5 of [RD06].

1.3 Acronyms and Abbreviations

aO	Active Optics
CCS	Common Control Software
CDT	Command Definition Table
ESO	European Southern Observatory
HOWFS	High Order Wave Front Sensor
LOWFS	Low Order Wave Front Sensor
SPR	Software Problem Report
TBD	To Be Defined
TCCD	Technical CCD
TCS	Telescope Control System
tif	telescope interface
VLT	Very Large Telescope

1.4 Applicable Documents

[AD01] *Active Optics and Guiding Control Functional Specification*, VIS-ATC-SPE-13030-0002, Issue 1.0.

[AD02] *M1 Control Software Functional Specification*, VIS-SPE-ACT-13080-0001, Issue 2.0.

[AD03] *Interface Control Document between the Telescope Control System and the M2 LCU*, VIS-ICD-ATC-05040-05090, Issue 1.1.

[AD04] *CCD Detectors Control Software User Manual*, VLT-MAN-ESO-17240-0672, Issue 1.6.

[AD05] *Operation of VISTA Active Optics*, VIS-TRE-VSC-13030-0001, Issue 1.0.

1.5 Reference Documents

[RD01] *VLT Software Auto Guiding and Field Stabilisation Design Description*, VLT-SPE-ESO-17230-0933 Issue 3.0.

[RD02] *VLT Software – Active Optics Design Description*, VLT-SPE-ESO-17210-1173, Draft.

[RD03] *Alignment and Number of Wavefront Sensors for VISTA*, VIS-TRE-ATC-00112-0012 Issue 1.0.

[RD04] *Low Order Wave Front Sensors Software Detailed Design*, VIS-DES-UOD-06048-0001, Issue 1.0.

[RD05] *Conversion from OWFS Astigmatism Measurements to M2 Tilt*, VIS-TRE-ATC-13030-0004, Issue 0.1.

[RD06] *Actuator Patterns, Quasi-Zernikies, and Vibration Modes on the Primary Mirror,* VIS-TRE-ATC-02020-0005, Issue 1.0.

2 Active Optics and Guiding Control Processes

The four control processes that implement active optics and guiding that are the subject of this document, all of which run on the TCS workstation, are listed in Table 1.

Environment	Process Name	Function
wvttcs	vtactconControl	control of M1 forces and M2 position
wvttcs	vtactsel1Select	selection of active optics reference star for the +Y LOWFS
wvttcs	vtactsel2Select	selection of active optics reference star for the -Y LOWFS
wvttcs	vtagwsControl	guide star selection and guiding control

Table 1 Active Optics and Guiding Control Processes

Each time the telescope is preset to a new "pointing", the preset action process[1] will send commands to all four active optics and guiding processes to select a guide-star, a pair of aO reference stars and to enable either open- or closed-loop active optics. The exact sequence of commands will depend on how far the telescope is being moved; there are four cases:

1. A slew that is large enough that:
    - The open-loop optics model must be applied to bring the optics into sufficiently good alignment for the low-order wave-front sensors to be able to analyse the wave-fronts and the active optics loop to be closed.
    - One (or possibly more) cycles of closed-loop active optics is required to bring the optics into sufficiently good alignment for a science exposure to start.
    - The accuracy of the guide star coordinates is better than the blind pointing performance of the telescope.
2. A large slew but with a small change in elevation so that the science exposure can start immediately without waiting for the first cycle of closed-loop active optics to complete.
3. A "small" offset which doesn't disturb the optics and where the offsetting accuracy of the telescope is likely to be better than the accuracy of the guide star coordinates.
4. A "dither" of only a few arc-seconds such that the current guide and aO reference stars can still be used.

The flow of commands between the preset process and the active optics and guiding control processes are illustrated in Figure 12 to Figure 14.

[1] The preset action process is a TCS workstation process that slews the telescope by sending a sequence of commands to other TCS processes.

The control processes control the set of LCU and workstation processes listed in Table 2 through the interfaces defined in the referenced documents.

Environment	Process Name	Function	Interface Doc
lvtag1	agServer	+Y auto-guider server	[RD01]
lvtag1	probeServer	virtual probe for +Y auto-guider	[RD01]
wvttcs	ccdconCI_AGPY	CCD control for +Y auto-guider	[AD04]
lvtag2	agServer	-Y auto-guider server	[RD01]
lvtag2	probeServer	virtual probe for –Y auto-guider	[RD01]
wvttcs	ccdconCI_AGNY	CCD control for –Y auto-guider	[AD04]
wvtia1	vtactServer	image analysis for +Y LOWFS	[RD04]
wvtia1	ccdconCI_WFSPY	CCD control for +Y LOWFS	[AD04]
wvtia2	vtactServer	image analysis for –Y LOWFS	[RD04]
wvtia2	ccdconCI_WSFNY	CCD control for –Y LOWFS	[AD04]
lvtm1	vtm1Server	M1 support control	[AD02]
lvtm2	vtm2Server	M2 positioning control	[AD03]

Table 2 Active Optics and Guiding processes

3 Active Optics

For an overview of the operation of the VISTA active optics see [AD05].

3.1 Process Structure

The TCS processes involved in active optics (except the selection of wave-front reference stars – described in Section 4 – and the high order wave-front sensor) are listed in Table 3 and their deployment illustrated in Figure 1.

Environment	Process Name	Function
wvttcs	vtprsAction	Preset action handling module
wvttcs	vtifControl	Telescope interface
wvttcs	vtmswControl	Mode switching
wvttcs	vtactconControl	Active optics control
wvtia1	vtactServer	Image analysis for +Y LOWFS
wvtia2	vtactServer	Image analysis for –Y LOWFS
wvtia1	ccdconCI_WFSPY	CCD control for +Y LOWFS
wvtia2	ccdconCI_WFSNY	CCD control for –Y LOWFS
lvtm1	vtm1Server	M1 support control
lvtm2	vtm2Server	M2 positioning control

Table 3 Active Optics Processes

Active optics is started and stopped by commands from the telescope interface process or the preset process; the mode switching process controls access to the M1 and M2 control systems.

The most significant difference from the VLT is that there are two LOWFS LCUs each with its own TCCD controller and image analysis processes and the WFS image analysis processes run on a dedicated workstation rather than on an LCU..

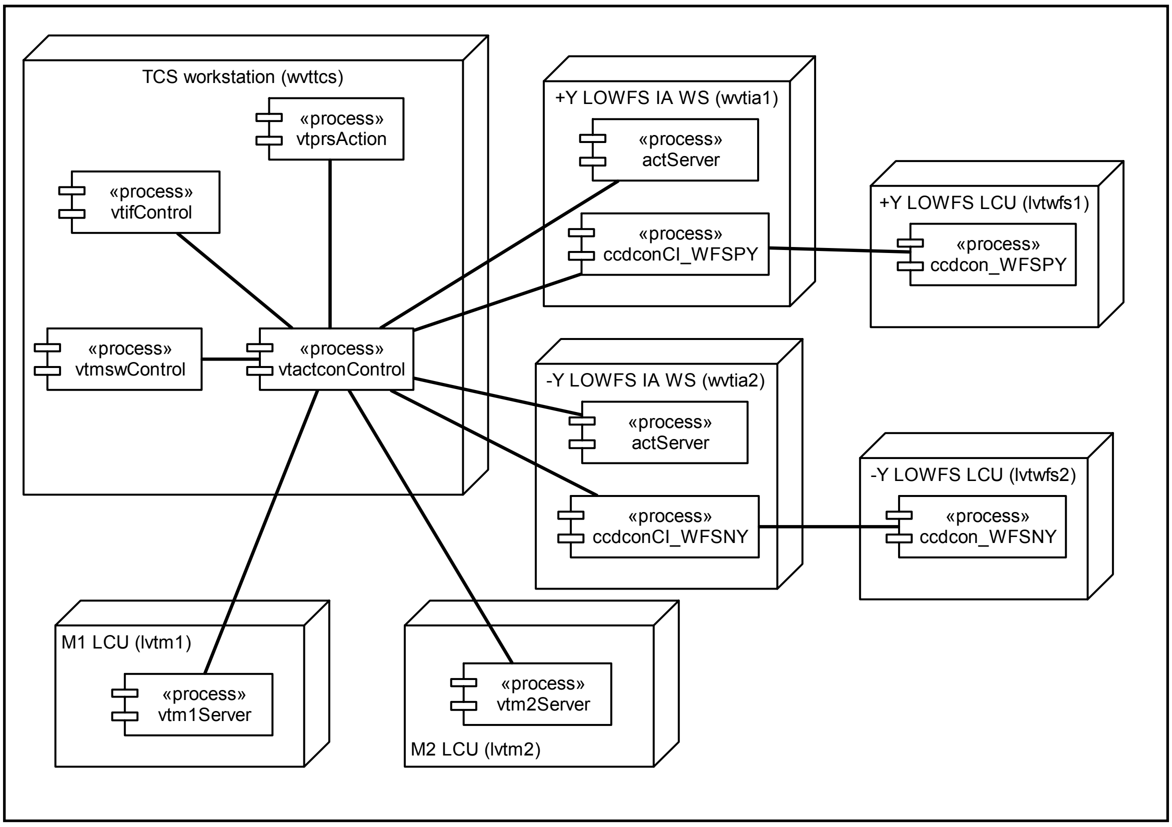

Figure 1 Active Optics Process Deployment

3.2 Internal Command Flow

3.2.1 Open Loop Active Optics

Open loop active optics is started by **vtactconControl** receiving a CYCLCAL command which causes the following actions to occur:

- **vtactconControl** sends a ONECAL command to itself.
- On receiving the ONECAL command it:

    1. Reads the current telescope altitude and temperature from the tif database.
    2. Reads the M2 look-up table coefficients from the database (6 coefficients for each of 5 Zernike coefficients).
    3. Applies the altitude and temperature to the look-up table functions (Equation 1 of [AD01]) to obtain values for each of Z4 to Z8.
    4. Adds the current focus offset to Z4.
    5. Reads the current zero-point adjustments from the data base (5 values).
    6. Adds the zero-points to Z4-Z8.

7. Reads the Zernike coefficient to M2 position conversion matrix from the database (5 values[2]).
8. Converts the Zernike coefficient values to M2 axis positions as described in section 0 and stores them in the database.
9. Reads the M1 look-up table coefficients from the database (6 coefficients for each of 18 M1 modes).
10. Applies the altitude and temperature to the look-up table functions (Equation 2 of [AD01]) to obtain values for each of the 18 modes.
11. Reads the zero point adjustments for each mode from the database (18 values) and add them to the results of step 10.
12. Reads the M1 mode to M1 forces conversion coefficients from the database (10 values for each of 4 rings of actuators).
13. Converts the mode values to a net force for each of the 84 actuators as described in section 3.4.5.4 and stores them in the database.

- It then sends itself a CORM1 command followed by a CORM2 command.

- On receiving the CORM1 command it:

1. Requests permission to update M1 from **vtmswControl** with a GETPERM command.
2. Reads the forces stored by ONECAL from the database.
3. Adds the M1 force errors stored in the database by ONEIA.[3]
4. Sends the forces to **vtm1Server** with a WRTABLE command.
5. Instructs M1 to apply the new forces with a ACTABS command.
6. Releases permission to update M1 with a RELPERM command to **vtmswControl**.

- On receiving the CORM2 command it:

1. Requests permission to update M2 from **vtmswControl** with a GETPERM command.
2. Reads the M2 axis positions stored by ONECAL from the database.
3. Adds the M2 position errors stored in the database by ONEIA.[3]
4. Reads the maximum allowed step size for each axis from the database (1 value for each of tip/tilt, focus and centring).
5. Moves M2 to the new position in steps no greater than the maximum step sizes by sending the appropriate commands to **vtm2Server**.
6. Releases permission to update M2 with a RELPERM command to **vtmswControl**.

[2] The M2 axes have been chosen such that the conversion matrix is diagonal.
[3] Note that when the system is operating open-loop, ONEIA is not being called so this step will add the difference between the open-loop model and the closed-loop forces from that last time the system was operating closed-loop. The values will be initialised to zero by INITAO.

- o After waiting a time interval (read from **:parameters.delayNext**) it sends itself another ONECAL command to begin the next cycle of open-loop correction.

Figure 15 illustrates the flow of commands sent between processes and that **vtactconControl** sends to itself when it receives the CYCLCAL command. This flow is the same as the VLT but the actions of some of the commands differ in detail because:

- o The zero-point adjustment does not exist in the VLT TCS[4]
- o The VLT TCS does not attempt to adjust the open-loop model with data from that last closed-loop cycle.
- o The number of M1 modes is different.
- o The number of M1 actuators is different.
- o The number and arrangement of temperature sensors is different.
- o The command used to apply the forces to M1 is different because the VLT M1 system does not do force balancing.
- o The VISTA will control all 5 M2 axes closed-loop whereas on the VLT coma is not actively controlled.
- o The commands to position M2 are different.
- o VISTA does not require control of the plate scale.

3.2.2 Closed Loop Active Optics

Closed loop active optics is started by **vtactconControl** receiving a CYCLAO command which causes the following process to occur (in the normal case where stars are available for both LOWFS and the wave-front analysis for both LOWFS completes sucessfully). The case of data from only one WFS being available is discussed in Section 3.2.3; if no analysis is possible for either WFS the system will alert the operator and fall back to open-loop operation.

- o **vtactconControl** sends itself a ONEIA command.

- o On receiving a ONEIA command:

    1. It sends a SETUP command to both LOWFS TCCD controllers to set the window and exposure time.
    2. Sends a START command to both controllers to start an exposure.
    3. Sends a WAIT command to both controllers to wait for the exposures to finish.
    4. Reads the positions in the focal plane of the two aO stars from the database.
    5. Calculates the distance of the aO stars from the optical axis.
    6. Reads the null aberration function coefficients from the database.

[4] The VLT is always operated with its active optics closed-loop and so the look-up tables are only required to bring the optics into sufficiently good alignment that the active optics loop can be closed. On VISTA, only the M2 position and, possibly, M1 astigmatism and trefoil, are controlled closed loop which means that the image quality is a function of the quality of the look-up tables. The zero-point adjustment allows small corrections to the look-up tables to be applied from occasional measurements made with the high-order wave-front sensor.

7. Calculates the magnitude of the null aberrations (including any requested de-focus) at the radial distances of the aO stars as described in section 3.4.5.2.
8. Adds the current focus offset to Z4.
9. "Rotates" the null aberrations into the x,y frame of the wave-front sensors.
10. Sends a SETWIN command to both **vtactServer** processes to define the position of the CCD frame.
11. Sends a STARPOS command to both **vtactServer** processes to define the position of the star image in the CCD frame.
12. Sends an STRTPNT command to both **vtactServer** processes to define the starting point for the wave-front analysis.
13. Reads an estimate of the current seeing from the database.
14. Sends the estimate to both **vtactServer** processes with a SETSEE command.
15. Sends an ANASTAR command to both **vtactServer** processes to analyse the wave-front images.
16. Sends a WAIT command to both **vtactServer** processes to wait for the image analyses to finish.
17. Checks that the analysis has completed successfully.
18. Reads the results of the wave-front analysis from the database and performs sanity checks as described in section 3.4.5.5.
19. Computes the Zernikes representing the measured error in M2 position as described in section 0.
20. "Rotates" the Zernikes from the wave-front sensor x,y coordinate frame to the telescope x,y frame.
21. Adds the new values to the running average of the difference between the open-loop model and the ideal M2 position as measured by the WFS.
22. Converts the running average Zernikes to M2 position adjustments and stores them in the database.
23. Computes the error in Z5, Z6, Z9 & Z10 (astigmatism and trefoil) as described in section 3.4.5.4.
24. Adds these new values to the running average.
25. Reads the M1 mode to M1 forces conversion coefficients from the database.
26. Converts the running average mode values to force adjustments for each of the 84 M1 supports and stores them in the database as described in section 3.4.5.4.

- It then sends itself a ONECAL command to calculate open-loop forces and positions.[5]

The subsequent command flow is the same as for the open-loop control case described in section 3.2.1.

The flow of commands for a single cycle of closed loop control is illustrated by Figure 16.

[5] At this point the VLT sends a ONECOR command as it never mixes open and closed loop control.

3.2.3 Degraded Operation

If data is available from just one low-order wave-front sensor – either because only one aO star is available or the image analysis for one sensor has failed to converge – the behaviour of ONEIA will be modified to:

- Use Z4 from the working sensor to estimate the error in focus.
- Use Z7 & Z8 as an estimate of the M2 centring error.
- Not update the M2 tilt error.
- Not update the error in M1 astigmatism and trefoil.

If both LOWFS are operating but cannot reliably measure Z5 to Z10 (because, for example, the seeing is very poor and the operator has disabled the fitting of all modes except Z4), focus will be controlled closed-loop but M2 tilt and decentring and M1 will revert to open-loop control.

3.2.4 Lookup Table Zero Point Update

The High-Order Wave-Front Sensor will be used to make occasional measurements that will be used to refine the M1 & M2 look-up tables by applying a "zero-point adjustment" – essentially an additional linear term. No such function does exists on the VLT so two new commands will be implemented.

- INCM1ZP and INCM2ZP will increment the zero points.

- SETM1ZP and SETM2ZP will set zero points.

The details of these commands are shown in the CDT listing in section 8.

The procedure for updating the zero points will be to disable all closed-loop control of the active optics, measure the resulting wave-front error (corrected for the null aberrations) and adjust the M1 and M2 focus and focus zero-points by that amount. Adjusting the M2 tilt zero-points would require two HOWFS measurements at different positions in the focal plane.

3.2.5 Focus Control

The interface for controlling the focus position on the VLT assumes that the focus is set by interactively adjusting the Z position of M2 until the instrument is in focus and then instructing the active optics to measure the focus offset seen by the wave-front sensor. This focus offset is then used in subsequent active optics calculations. On VISTA the focus may have to be adjusted each time a new filter is selected but it can be assumed that the offset for each filter can be measured during commissioning and will remain fixed thereafter.

Therefore, the VLT focus related commands will be removed and replaced by a command that simply sets the value of focus offset to be used in the active optics calculations.

3.3 Command Interface

In addition to the standard VLT commands listed in Table 4, the **vtactconControl** process will support the commands listed in Table 5. With the exception of the new commands UPDLTZP and RSTLTZP they all perform similar functions to the same commands supported by the VLT/NTT **actconControl** process.

Command	Description
BREAK	Stop all actions
EXIT	Terminate process
INIT	Initialise software and read configuration parameters
KILL	Terminate process
OFF	Switch to OFF operating state
ONLINE	Switch to ONLINE operating state
PING	Verify that process is alive
SELFTST	Perform self-test of software module
SIMULAT	Switch on simulation mode
STANDBY	Switch to STANDBY operating state
STATE	Return current state
STATUS	Return status of module
STOP	Stop all ongoing actions
STOPSIM	Switch off simulation mode
TEST	Perform complete test of software module
VERBOSE	Switch verbose mode on or off
VERSION	Return software module version number

Table 4 Commands supported by all TCS processes

Command	Description
CORM1	Set M1 forces as defined in database
CORM2	Set M2 position as defined in database
CYCLAO	Start continuous aO correction
CYCLCAL	Start continuous open-loop correction
FGETAO	Get current defocus value
INCM1ZP	Increment M1 zero points
INCM2ZP	Increment M2 zero points
INITAO	Initialise LCU and CCD
INITCOR	Initialise M1 & M2 correction system
INITIA	Initialise image analysis
OFFSFOC	Store defocus offset
ONEIA	Analyse one set of LOWFS data
ONECAL	Do one cycle of open-loop correction
ONECOR	Set M1 forces & M2 position as defined in database
SETM1ZP	Set M1 zero points
SETM2ZP	Set M2 zero points

Command	Description
SEQAO	Do a sequence of aO corrections
SEQIA	Do a sequence of image analyses
SHUTDWN	Shutdown aO
STPWAIT	Stop aO after current cycle is complete
STOP	Stop aO
TERMCOR	Terminate M1 & M2 correction system
UPDLTZP	Update lookup-table zero-points
WIPECCD	Read out CCDs to clear charge

Table 5 Commands supported by vtactconControl

3.4 vtactwsControl Process Internals

3.4.1 Module Dependencies

The dependency relationships between the modules that implement the **vtactconControl** process are shown in Figure 2.

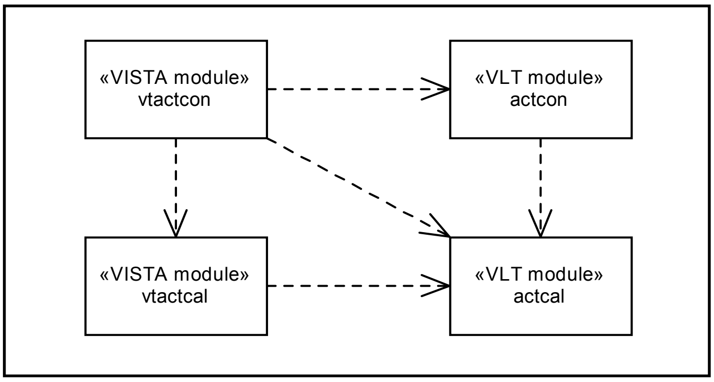

Figure 2 vtactconControl module dependencies

3.4.2 Implementation

Each **actconControl** specific command is implemented in its own class which inherits from actconSINGLE_CMD which in turn inherits from a number of CCS classes. For example, the INITAO command is implemented in actconCMD_INITAO and its relationship to other classes is illustrated by Figure 3.

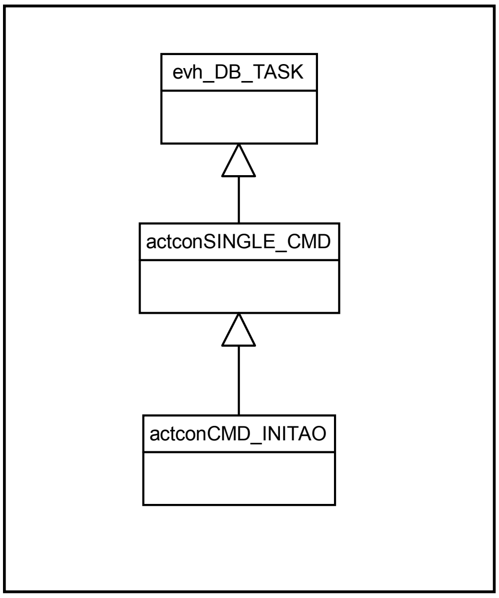

Figure 3 Inheritance Relationship for Class actconCMD_INITAO

Some of the VISTA specific commands are implemented by classes that inherit from actonSINGLE_CMD in the same way. However, actconSINGLE_CMD implements a number of methods that need to be changed for VISTA so the VISTA commands that use these methods are implemented by classes that inherit from a class called vtactconSINGLE_CMD which inherits from actconSINGLE_CMD and overrides the methods that require VISTA specific implementations.

The methods that need to be overridden are those associated with selecting the destinations for messages to the CCD controller and image analysis processes, the optics calculations implemented in the vtactcal module. The new VISTA commands that do not exist in actconControl also need to override the command state handling methods. The relationship between the command classes is show in Figure 4.

A number of vtactconControl commands have to send commands to both CCD controller and both image analysis processes in parallel which actconControl does not do. These commands use vtactconPARALLEL_CMD objects that are mostly copied from the agws module.

3.4.3 Command Object Interactions.

The flow of control through the **vtactconControl** command objects for each of the processes described in Section 3.1 are detailed in Figure 17 and Figure 18.

3.4.4 Command Specifics

This section details the modifications required to the VLT command implementation classes contained in the actcon module.

3.4.4.1 CORM1

This command will require modifications:

1. To handle a mixture of relative (closed-loop) and absolute position (open-loop) settings[6].
2. To take account of the differences in the interface to **vtm1Server.**

3.4.4.2 CORM2

This command will require modifications:

1. To handle a mixture of relative (closed-loop) and absolute position (open-loop) settings.
2. To take account of the differences in the interface to **vtm2Server.**

3.4.4.3 CYCLAO

This command will need to be modified to remove the sending of an M1MOVE command to **vtm1Server** and to send a ONECAL command **vtactconControl** instead of a ONECOR command.

3.4.4.4 CYCLCAL

This command performs a number of cycles of open-loop corrections and the VLT/NTT implementation of this command can be used unchanged.

[6] In principle this makes the CYCLCAL and SEQCAL commands redundant because the same effect can be realised by CYCLAO and SEQAO with the flags set to make all M2 position and M2 modes open-loop but these commands will be retained to preserve the interface familiar to the VLT operators.

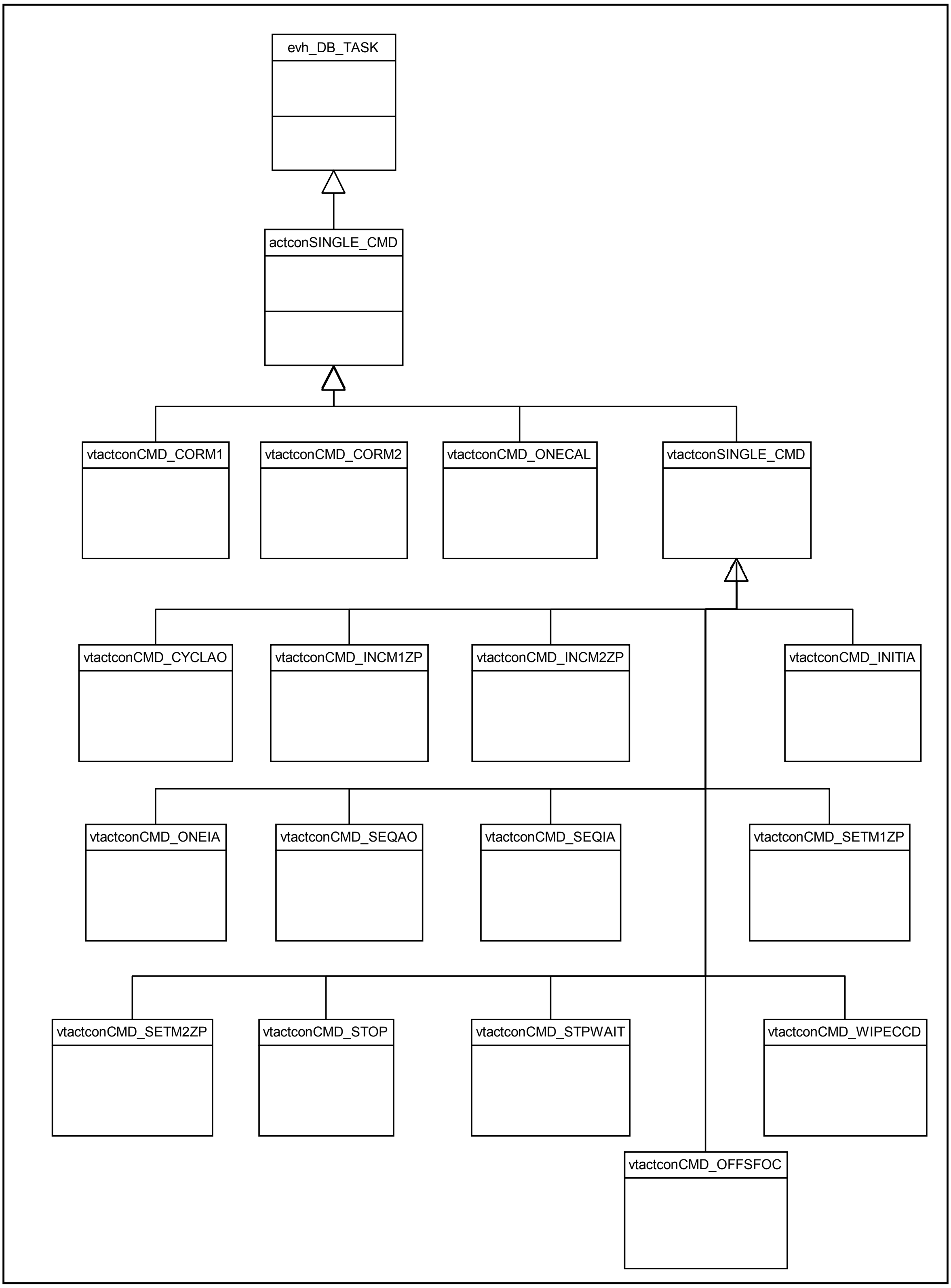

Figure 4 Active optics control command classes.

3.4.4.5 FGETAO

This command (stores the currently measured defocus) has no function on VISTA and will be removed.

3.4.4.6 INCM1ZP

This command increments the M1 look-up table zero points. It does not exist on the VLT/NTT and will have to be implemented from scratch.

3.4.4.7 INCM2ZP

This command increments the M1 look-up table zero points. It does not exist on the VLT/NTT and will have to be implemented from scratch.

3.4.4.8 INITAO

This command initialises the active optics system and VLT/NTT implementation of this command can be used unchanged.

3.4.4.9 INITCOR

This class contains code is specific to the NTT that is only executed if the database flag **:targetM1.vltTelescope** has the value zero. Provided this is set similarly on VISTA the VLT/NTT implementation can be used unchanged.

3.4.4.10 INITIA

In addition to other processing this command sends a CREATRA command to the WFS TCCD controller which is not required by the VISTA wave-front sensors.

3.4.4.11 M1PMOVE

This command (move M1 fixed points) has no function on VISTA and will be removed.

3.4.4.12 OFFSET

This command (sets a flag in the database that indicates whether the defocus value stored by FGETAO should be used) has no function on VISTA and will be removed.

3.4.4.13 OFFSFAD

This command (applies a focus offset to the WFS adapter) has no function on VISTA and will be removed.

3.4.4.14 OFFSFM2

This command move M2 in focus will need to be modified because the names of commands to control the position of M2 are different.

3.4.4.15 OFFSFOC

This command stores the focus offset in the database. It does not exist on the VLT/NTT and will have to be implemented from scratch.

3.4.4.16 ONECAL

This command will need to be modified to remove code associated with control of the plate scale.

3.4.4.17 ONECOR

This command may need modification in order to change the logging but the overall structure can remain the same.

3.4.4.18 ONEIA

This command will require modification to:

1. Remove setting the WFS adapter to see the star.
2. Send an estimate of the seeing to both **vtactServer** processes.
3. Send the wave-front analyse starting point to both **vtactServer** processes.
4. Take exposures with two TCCD controllers.
5. Send commands to analyse the image to both **vtactServer** processes.

3.4.4.19 REFIA

This command (analyse reference source image) has no function on VISTA and will be removed.

3.4.4.20 RPSCALE

This command (reset relative plate scale) has no function on VISTA and will be removed.

3.4.4.21 SETM1ZP

This command sets the M1 look-up table zero points. It does not exist on the VLT/NTT and will have to be implemented from scratch.

3.4.4.22 SETM2ZP

This command sets the M1 look-up table zero points. It does not exist on the VLT/NTT and will have to be implemented from scratch.

3.4.4.23 SEQAO

This command will need to be modified to send a ONECAL command instead of ONECOR.

3.4.4.24 SEQIA

The VLT/NTT implementation of this command can be used unchanged

3.4.4.25 SHUTDWN

This command shuts down the active optics and VLT/NTT implementation of this command can be used unchanged.

3.4.4.26 SOURCE

This command (set/unset adapter light source) has no function on VISTA and will be removed.

3.4.4.27 SPSCALE

This command (set relative plate scale) has no function on VISTA and will be removed.

3.4.4.28 STOP

This command stops all ao correction and needs to be changed to send STOP commands to both CCD controllers and both image analysis processes.

3.4.4.29 STPWAIT

This command waits for the current cycle to ao correction to complete and then stops correction and needs to be changed to send STOP commands to both CCD controllers.

3.4.4.30 TERMCOR

This class contains code is specific to the NTT that is only executed if the database flag **:targetM1.vltTelescope** has the value zero. Provided this is set similarly on VISTA the VLT/NTT implementation can be used unchanged

3.4.4.31 WIPECCD

This command must be modified to send all commands to both TCCD controllers.

3.4.5 Active Optics Calculations

The active optics calculations (table lookups and force and position calculations) are implemented in the **actcal** module and a VISTA version of this module will be required. The VISTA version will differ in detail throughout (with the exception of the Kalman filter class).

3.4.5.1 Open-Loop Models

The open loop models for both M1 figure and M2 position are described by a 4th order Legrange polynomial in zenith distance and a linear dependency on ambient temperature. The zenith distance is normalised to 30° and the temperature to 8°C so the fitting function is of the form:

$$c_o + c_1 z + c_2(2z^2 + z) + c_3(6z^3 - 6z^2 + z) + c_4(20z^4 - 30z^3 + 12z^2 - z) + tT$$

where, z = (zenith distance - 30°) and T = (temperature - 8°C).

3.4.5.2 Null Aberrations

The null aberrations (the aberrations expected at a specified point in the field of view of the WFS with the optics in optimal alignment) for the modes analysed by the low-order wave-front sensors (in nm) are described by functions of the position in the focal plane (in degrees) are listed in Table 6 and Table 7.

Mode	Function
Z4	$+625.988 - 1855.253 \times y + 1418.010 \times y^2$
Z5	$-230.666 \times x - 68.131 \times x \times y$
Z6	$-1077.017 + 3554.101 \times y - 2782.632 \times y^2$
Z7	$+233.373 - 385.774 \times y + 79.433 \times y^2$
Z8	$+1280.970 \times x - 1069.864 \times x \times y$
e(3,1)sin	$-833.256 + 2544.974 \times y - 2088.596 \times y^2$
e(3,1)cos	$+802.944 \times x - 1604.986 \times x \times y$
Z11	$+336.209 - 786.136 \times y + 414.074 \times y^2$
Z12	$-307.241 + 944.526 \times y - 840.733 \times y^2$
Z13	$-218.135 \times x + 545.542 \times x \times y$

Table 6 Formulae for +Y LOWFS Null Aberrations

Mode	Function
Z4	$+625.988 + 1855.253 \times y + 1418.010 \times y^2$
Z5	$-230.666 \times x - 68.131 \times x \times y$
Z6	$-1077.017 - 3554.101 \times y - 2782.632 \times y^2$
Z7	$-233.373 - 385.774 \times y - 79.433 \times y^2$
Z8	$+1280.970 \times x + 1069.864 \times x \times y$
e(3,1)sin	$+833.256 + 2544.974 \times y + 2088.596 \times y^2$
e(3,1)cos	$+802.944 \times x + 1604.986 \times x \times y$
Z11	$+336.209 + 786.136 \times y + 414.074 \times y^2$
Z12	$-307.241 - 944.526 \times y - 840.733 \times y^2$
Z13	$+218.135 \times x + 545.542 \times x \times y$

Table 7 Formulae for –Y LOWFS Null Aberrations

3.4.5.3 M2 Position Calculations

Each of the Zernike coefficients stored in the M2 position look-up table is related to a shift of M2 in each of its 5 axes by a constant as shown in Table 8.

Zernike	M2 axis
Z4	Z axis
Z5	rotation about coma neutral point around X axis
Z6	rotation about coma neutral point around Y axis
Z7	rotation about centre of curvature around X axis

Zernike	M2 axis
Z8	rotation about centre of curvature around Y axis

Table 8 Relationship between Zernikes and M2 position shifts

The relationship between the shifts in M2 position and the Zernike coefficients measured by the two low-order wave-front sensors (after subtraction of the null aberrations) are shown in Table 9. The subscripts 1 & 2 refer to the #1 and #2 low-order wave-front sensors respectively, r is the angular separation between the reference stars and e is the angle between a line between the two stars and the y axis as shown in Figure 5. For a derivation of these formulae see [RD05]. Note that these coefficients are in the frame of reference of the wave-front sensors and have to be rotated by the Cassegrain rotator position angle before being applied to M2.

M2 axis	function
Focus	$(Z4_1 + Z4_2)/2$
X tilt	$((Z5_1 - Z5_2)\cos(e) + (Z6_1 - Z6_2)\sin(e))/r$
Y tilt	$(-(Z5_1 - Z5_2)\sin(e) + (Z6_1 - Z6_2)\cos(e))/r$
X coma	$(Z7_1 + Z7_2)/2$
Y coma	$(Z8_1 + Z8_2)/2$

Table 9 Formulae for combining Zernikes from two WFS

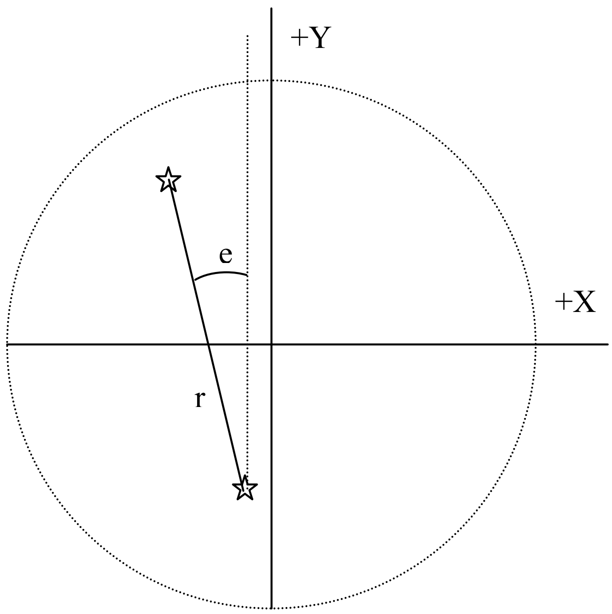

Figure 5 aO star positions in the focal plane

3.4.5.4 M1 Modes and Actuator Forces

The relationship between the Zernikes measured by the LOWFS and errors in M1 modes is shown in Table 10.

M1 mode	function
e(2,1)sin	$(Z5_1 + Z5_2)/2$
e(2,1)cos	$(Z6_1 + Z6_2)/2$
e(3,1)sin	$(Z9_1 + Z9_2)/2$
e(3,1)cos	$(Z10_1 + Z10_2)/2$

Table 10 Relationship between LOWFS Zernikes and M1 modes

The relationship of the force to be applied to an M1 actuator in ring i at azimuthal angle θ for each of the 18 M1 modes is shown in Table 11 where $G_{m,n}$ is the modulus and φ the azimuthal angle of the mode.

m	n	function
2	1	$G_{2,1} \times F_{2,1,i} \times \cos(2(\theta - \varphi_{2,1}))$
3	1	$G_{3,1} \times F_{3,1,i} \times \cos(3(\theta - \varphi_{3,1}))$
4	1	$G_{4,1} \times F_{4,1,i} \times \cos(4(\theta - \varphi_{4,1}))$
2	2	$G_{2,2} \times F_{2,2,i} \times \cos(2(\theta - \varphi_{2,2}))$
0	2	$G_{0,2} \times F_{0,2,i}$
1	2	$G_{1,2} \times F_{1,2,i} \times \cos(\theta - \varphi_{1,2})$
5	1	$G_{5,1} \times F_{5,1,i} \times \cos(5(\theta - \varphi_{5,1}))$
0	3	$G_{0,3} \times F_{0,3,i}$
6	1	$G_{6,1} \times F_{6,1,i} \times \cos(6(\theta - \varphi_{6,1}))$
3	2	$G_{3,2} \times F_{3,2,i} \times \cos(3(\theta - \varphi_{3,2}))$

Table 11 Formulae for converting M1 modes to actuator forces

3.4.5.5 LOWFS sanity checks

If the image analyses for both low-order wave-front sensors have converged ONEIA will check that each of Z7, Z8, Z9 & Z10 differ by no more than the sum of following function of the magnitudes of the aO stars and the current estimate of the seeing for the two stars before using the data to update the error estimates:

$$a \times \sqrt{1 + (seeing / b)^2} + \max(0.0, c * (magnitude - d))$$

If these checks fail the system will alert the operator and fall back to open-loop control of both M1 and M2.

3.4.6 Internal Database layout

The database points defined by **vtactconControl** are shown in Table 31 in Section 7.

3.4.7 External Database access

The database points defined by other modules that **actconControl** will access are listed in Table 12.

Name	Description
:Appl_data:TCS:act:select1:control:aoStar.usable	LOWFS #1 star available
:Appl_data:TCS:act:select1:control:aoStar.alpha	right ascension of LOWFS #1 star
:Appl_data:TCS:act:select1:control:aoStar.delta	declination of LOWFS #1 star
:Appl_data:TCS:act:select1:control:aoStar.magnitude	magnitude of LOWFS #1 star
:Appl_data:TCS:act:select1:control:aoStar.x	x position of LOWFS #1 star
:Appl_data:TCS:act:select1:control:aoStar.y	y position of LOWFS #1 star
:Appl_data:TCS:act:select1:control:aoStar.ccdx	x position of LOWFS #1 star on CCD
:Appl_data:TCS:act:select1:control:aoStar.ccdy	y position of LOWFS #1 star on CCD
:Appl_data:TCS:act:select1:control:ccd.angle	orientation of LOWFS #1 CCD
:Appl_data:TCS:act:select2:control:aoStar.valid	LOWFS #2 star available
:Appl_data:TCS:act:select2:control:aoStar.alpha	right ascension of LOWFS #2 star
:Appl_data:TCS:act:select2:control:aoStar.delta	declination of LOWFS #2 star
:Appl_data:TCS:act:select2:control:aoStar.magnitude	magnitude of LOWFS #2 star
:Appl_data:TCS:act:select2:control:aoStar.x	x position of LOWFS #2 star
:Appl_data:TCS:act:select2:control:aoStar.y	y position of LOWFS #2 star
:Appl_data:TCS:act:select2:control:aoStar.ccdx	x position of LOWFS #1 star on CCD
:Appl_data:TCS:act:select2:control:aoStar.ccdy	y position of LOWFS #1 star on CCD
:Appl_data:TCS:act:select2:control:ccd.angle	orientation of LOWFS #2 CCD
:Appl_data:TCS:agws:control:guideStar:FWHM_10	FWHM of guide star image
@wvtia1:vtact:fit:coeffs.active	mode actively fitted
@wvtia1:vtact:fit:coeffs.modulus	modulus of mode
@wvtia1:vtact:fit:coeffs.angle	angle of more
@wvtia1:vtact:fit:simplexDiag:successFlag	fit successful flag
@wvtia2:vtact:fit:coeffs.active	mode actively fitted
@wvtia2:vtact:fit:coeffs.modulus	modulus of mode
@wvtia2:vtact:fit:coeffs.angle	angle of more
@wvtia2:vtact:fit:simplexDiag:successFlag	fit successful flag

Table 12 Database points accessed by vtactconControl

Name	Description
TBD	telescope temperature

ALT	telescope altitude
ROT_ANGLE	rotator position angle

Table 13 tif data accessed by vtactconControl

3.4.8 Commands Issued

All the commands sent by vt**actconControl** to other TCS processes are listed in Table 14 to Table 18.

Command	Description
GETPERM	Get permission to control optics
RELPERM	Relinquish permission to control optics

Table 14 Commands sent by vtactconControl to vtmswControl

Command	Description
ACTREL	Set relative forces
ACTABS	Set absolute forces
WRTABLE	Write table of absolute forces

Table 15 Commands sent by vtactconControl to vtm1Server

Command	Description
SETABS	Set M2 position

Table 16 Commands sent by vtactconControl to vtm2Server

Command	Description
ABORT	Abandon processing
ANASTAR	Analyse wave-front images
SETSEE	Set estimate of seeing
SETWIN	Set CCD windows size and position
STARPOS	Set star position on CCD image
STRPNT	Set starting point for wave-front analysis
WAIT	Wait for analysis to finish

Table 17 Commands sent by vtactconControl to vtactServer

Command	Description
ABORT	Stop exposure
SETUP	Configure TCCD
START	Start exposure
WAIT	Wait for exposure to complete

Table 18 Commands sent by vtactconControl to TCCD control processes

4 Wave Front Sensor Star Selection

4.1 Process Structure

The TCS processes involved in the selection of wave-front reference stars are listed in Table 19 and their deployment illustrated in Figure 6.

Environment	Process Name	Function
wvttcs	vtprsAction	Preset action handling module
wvttcs	vtactsel1Control	Star selection for +Y LOWFS
wvttcs	ccdconCI_WFSPY	CCD control for +Y LOWFS
lvtac1	vtact1Server	+Y LOWFS image analysis
wvttcs	vtactsel2Control	Star selection for –Y LOWFS
wvttcs	ccdconCI_WFSNY	CCD control for –Y LOWFS
lvtac2	vtactServer	-Y LOWFS image analysis

Table 19 Wave-front Reference Star Selection Processes

Selection of the reference stars for the low-order wave-front sensors is done independently for the two sensors by two processes that are identical except for the database entries that specify which TCCD controller they communicate with and the positions of the CCDs in the focal plane.

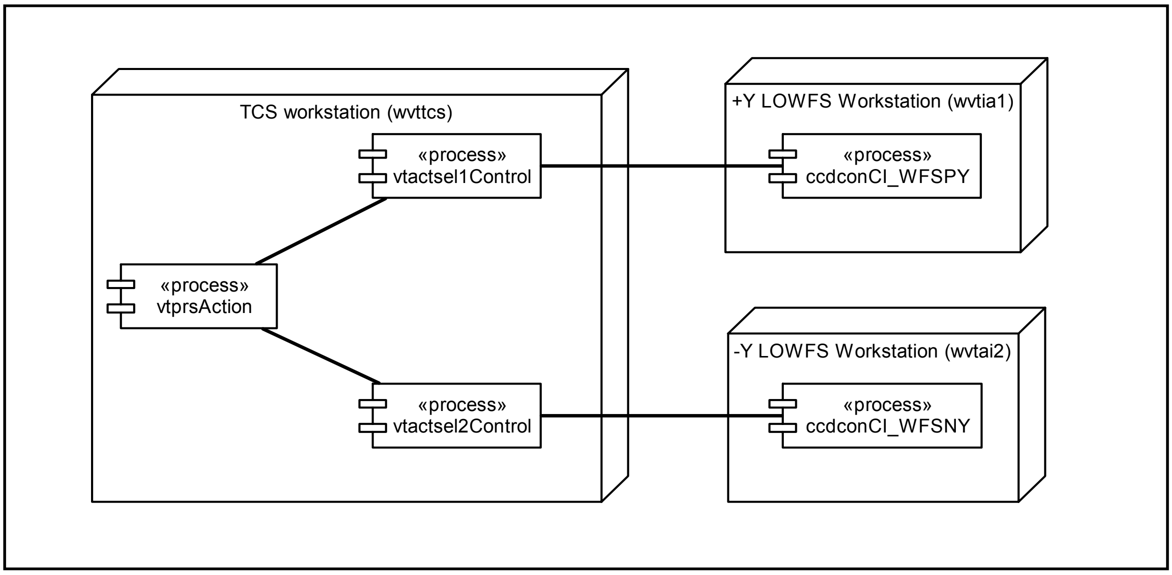

Figure 6 Wave-Front Reference Star Selection Deployment

4.2 Command Interface

In addition to the standard commands listed in Table 4 the **vtactselControl** processes will support the commands listed in Table 20. This is a subset of the commands supported by the

VLT/NTT **agwsControl** process which is responsible for selecting guide stars (in addition, to controlling auto-guiding and field stabilisation) except that the string "GS" has been replaced by "AOS" to reflect the different purpose and "PR" (probe) has been replaced by "WI" (window) in the commands that **agwsControl** passes onto the probe server.

Command	Description
CANDAOS	Add star to internal catalogue
GETAOS	Return aO star α,δ
GETAOSS	Return the contents of the internal catalogue
PRNCAT	Prints the internal catalogue
SELAOS	Select a star from the internal catalogue to be used for aO
SRCAOS	Search the standard guide-star catalogue for suitable candidate stars
SETUP	Handle new setup configuration
STRTCCD	Start continuous read-out of CCD
STOPCCD	Stop continuous read-out of CCD
WIABS	Set CCD window to absolute α,δ or x,y.
WI2STAR	Position CCD window to current CCD reference position
WIAOS	Position CCD window to next suitable entry in the internal catalogue
WITOAOS	Position CCD window to next suitable star, searching external catalogue if necessary

Table 20 Commands supported by vtactselControl processes

4.3 actselControl Process Internals

4.3.1 Module Dependencies

vtactselControl will be implemented in a module called **vtactsel** by renaming and extending the TCS template workstation application and will not depend on any other VLT application modules. However, much of the code will copy similar functions in **agws** and **vtagws**.

4.3.2 Implementation

The most significant changes to the similar functions in the **agws** module will be:

- A different algorithm for selecting suitable stars from the internal catalogue. This will be similar to that in **vtagws** but simpler because there is only one CCD.

- The read-out window on the wave-front sensor CCD needs to be set directly because there is no probe server emulating a moveable probe as there is for auto-guiding.

- The position and size of the CCD window and the estimated star position have to be sent to the associated **actServer** process so that it knows which part of the bad pixel file to use.

- Two copies of **vtactselControl** will be running in the same workstation environment so the name of the database branch being used must be determined at run-time on the basis of the process name.

4.3.3 Command Specifics

4.3.3.1 GETAOS

This command will be a copy of GETGS from **agws** but the results returned by this command will need to be modified because the wave-front sensor does not measure the star position.

4.3.3.2 GETAOSS

This command will be a copy of GETGSS from **agws**; only the command name needs to be changed.

4.3.3.3 PRNCAT

This command prints the contents of the internal catalogue; the **agws** implementation can be used unchanged.

4.3.3.4 SELAOS

This command will be a copy of SELGS from **agws**; only the command name needs to be changed.

4.3.3.5 SRCAOS

This command will be a copy of SRCGS from **agws** but the strategy for searching for suitable aO stars will be different. Instead if searching a toroidal region centred on the science target a circular region centred on the centre of the CCD will be searched instead. The results of the search will then have to be checked against the actual region of sky covered by the (rectangular) CCD.

4.3.3.6 SETUP

This command will be a much simplified copy of SETUP from **agws** as it only needs to handle parameters for the CCD controller.

4.3.3.7 STRTCCD

The **agws** implementation of this command can be used unchanged.

4.3.3.8 STOPCCD

This **agws** implementation of this command can be used unchanged.

4.3.3.9 WIABS

This command sets the CCD window to an absolute α,δ or x,y and is an analogue of the PRABS command in **agws.** Its implementation will be quite different as PRABS is just forwarded to the probe module.

4.3.3.10 WI2STAR

This command centres the TCCD window on the reference point (selected from the real-time display by picking an object or pixel position).

4.3.3.11 WIAOS

This command selects the next suitable ao star and calculates the position of the CCD window that has the star position at its centre. It is an analogue of the PRGS command in **agws**.

4.3.3.12 WITOAOS

This command is similar to WIAOS but also searches the external catalogue if the internal catalogue does not contain any suitable stars.

4.3.4 Database layout

The database points defined by **vtactSelect** are listed in Table 21 and the tiff data items accessed by **vtactselControl** listed in Table 1. The layout is essentially a copy of the relevant parts of the **agwsControl** database to facilitate the re-use of code from the **agws** and **vtagws** modules. There will be two copies of the structure with <n> set to 1 and 2.

Name *(Appl_data:TCS:act:select<n>:control:)*		***Class*/Type**	**Description**
aoStar		*actselAO_STAR*	
	.alpha	double	right ascension
	.delta	double	declination
	.magnitude	double	magnitude
	.wavelength	double	effective wavelength
	.x	double	x position in focal plane (deg)
	.y	double	y position in focal plane (deg)
	.strx	int32	x origin of window
	.stry	int32	y origin of window
	.nx	int32	x window size
	.ny	int32	y window size
	.ccdx	double	x position in window
	.ccdy	double	y position in window
	.usable	Boolean	star data is valid
aoStarTable()		*actselAO_TABLE*	
	alpha	double	right ascension
	delta	double	declination
	magnitude	double	magnitude
	wavelength	double	effective wavelength
	tryState	int32	try state

Name (*Appl_data:TCS:act:select<n>:control:*)		***Class*/Type**	**Description**
	.padding	int32	unused[7]
searchCriteria		*vtactselSRCH_CRIT*	
	.catalogueName	BYTES32	catalogue name
	.minMag	double	minimum brightness
	.maxMag	double	maximum brightness
ccd		*vtactselCCD_GEOM*	
	.useable	Boolean	Flag that can disable use of the CCD
	.xorigin	double	x coordinate of CCD in focal plane
	.yorigin	double	y coordinate of CCD
	.xsize	int32	width of CCD in pixels
	.ysize	int32	height of CCD in pixels
	.angle	double	orientation of CCD
	.pixscale	double	pixel scale of CCD
targetCCD		*vtactselPROC_ENTRY*	
	.destination	BYTES32	destination name
	.modName	BYTES32	module name
	.modSymAddr	BYTES32	module state
cmdTimeouts		*vtactselCON_SERVER*	
	.stopTimeout	int32	timeout for STOP
	.startTimeout	int32	timeout for START
ccdSetup		*vtactselCCD_SETUP*	
	.defaultSetup	BYTES32	setup file name

Table 21 Database points defined by vtactselControl

Name	**Description**
RA2000	target right ascension
DEC2000	target declination
ADA_POSANG	field orientation

Table 22 tif data accessed by vtactselControl

4.3.5 Commands issued

All the commands sent by the **vtactselControl** processes to other TCS processes are listed in Table 23.

[7] This field needs to be retained so that the records in the table are aligned in memory such that they can be read directly into an array of C structures.

Command	Description
START	Start exposure
STOP	Stop CCD exposure

Table 23 Commands sent by vtactselControl to TCCD control processes

5 Auto-guiding Control

5.1 Process Structure

The TCS processes involved in the auto-guiding star selection and control are listed in Table 24 and their deployment illustrated in Figure 7.

Environment	Process Name	Function
wvttcs	vtprsAction	Preset action handling module
wvttcs	vtagwsControl	auto-guiding control
wvttcs	trkwsControl	tracking control
lvtag1	agServer	+Y auto-guider server
lvtag1	probeServer	+Y auto-guider virtual probe
wvttcs	ccdconCI_AGPY	+Y auto-guider CCD control
lvtag2	agServer	-Y auto-guider server
lvtag2	probeServer	-Y auto-guider virtual probe
wvttcs	ccdconCI_AGNY	-Y auto-guider CCD control

Table 24 Auto-guiding Control Processes

The **vtagwsControl** process differs from the corresponding VLT/NTT implementation because:

1. The criteria for selecting guide stars are different – there are two fixed CCDs instead of a moveable probe that can be positioned in the focal plane.

2. Once a guide star has been selected the control process has to choose which of the two sets of probe control, TCCD controller and auto-guiding server process to control.

3. VISTA does not support field stabilization[8] or rapid guiding[9].

Other differences in the auto-guiding implementation are hidden by the virtual probe server process which emulates a moveable probe with a moveable read-out window on a fixed CCD.

[8] High band-width tip/tilt corrections sent directly from a wave-front sensor to M2 to compensate for wind buffeting and, if the wave-front sensor is within the isoplanatic patch, atmospheric tip/tilt of the wave-front.
[9] Direct control of M2 tip/tilt from an instrument.

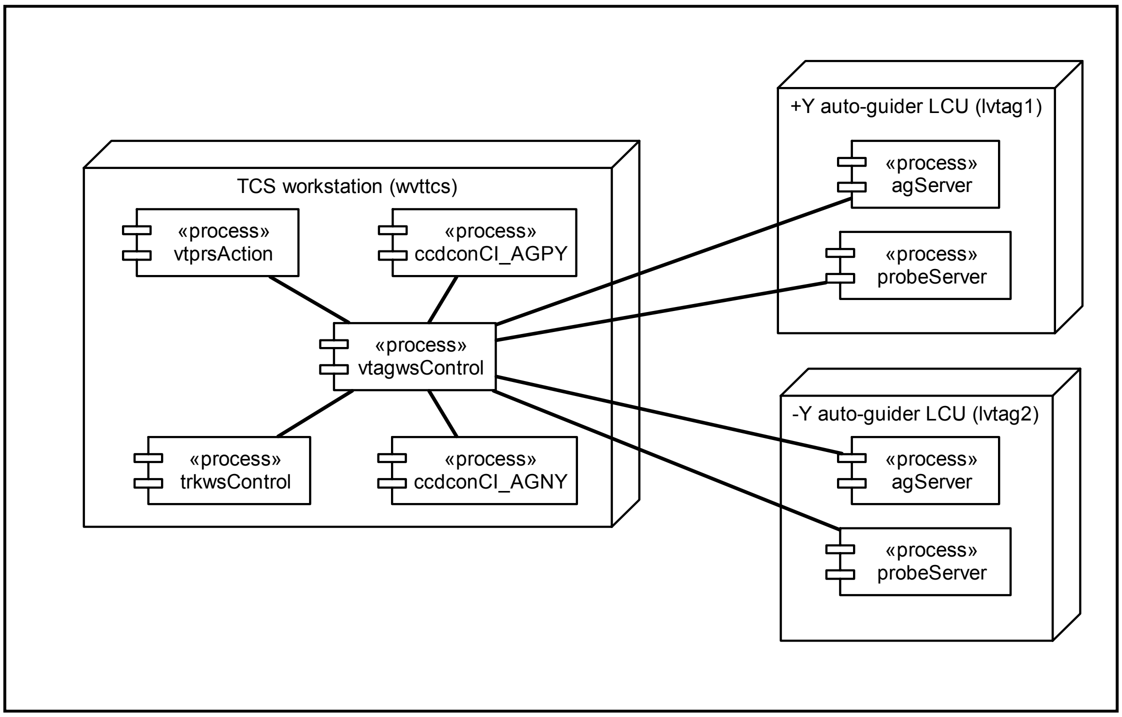

Figure 7 Guiding Process Deployment

5.2 *Command Interface*

Table 5 lists all the commands other than the standard commands and those that are relevant only to field stabilisation or fast guiding that are defined in the **agws** module's command definition table along with a brief description of their function and some remarks about the relevance of the command to VISTA.

"Internal catalogue" refers to the list of five stars held internally that either come from a SETUP command or are the result of searching a guide star catalogue.

Command	Description
BOX2GS	Moves the read-out window on the CCD to centre the guide star. This command is intended for use on the NTT. On VISTA, moving the probe has the same effect so there is no point in implementing this command.
CANDGS	Adds a star to the internal catalogue.
CNTOBJ	Moves the probe to the centre of the field. This cannot be implemented on VISTA because the probe can't move. It isn't implemented on the VLT either.
DETGS	Takes a CCD image, finds the centroid and moves the telescope with OFFSAG to centre the star on the probe.
GETGS	Returns current guide star position, magnitude etc..
GETGSS	Returns the contents of the internal catalogue.
OFFSADG	Stops guiding, offsets the telescope by changing the science target α,δ and

	restarts guiding.
OFFSGP	Moves probe by changing the guide star α,δ in tiny increments without stopping guiding. This is used by instruments to correct for flexure and is not needed for VISTA
OFFSAG	Offsets telescope by "manual" guiding. This is command is just forwarded to agServer so VISTA can use the VLT implementation.
OFFSXY	Offsets probe by x,y in CDD coordinate frame by adjusting the guide star α,δ. This command is just relayed to the probe module. However, this command is not yet implemented in probe so currently it does not work.
PR2STAR	Moves probe to α,δ selected from a guider CCD image. Sends OFFSXY to probe so currently it does not work.
PRABS	Moves the probe to an absolute α,δ or x,y. This command is just forwarded to the probe module.
PRALT	Move the probe to its "alternate" position. This command is just forwarded to the probe module but is not relevant to VISTA.
PRCHK	Check that guide star is valid. This command is just forwarded to the probe module.
PRCNT	Moves the probe to the field centre. This command is just forwarded to the probe module but is not relevant to VISTA.
PRDOBJ	This command is not used and can be ignored
PRGS	This command is the same as PRTOGS except that it just uses the internal guide star catalogue and doesn't search the external catalogue. VISTA can use the VLT implementation.
PRNCAT	Prints the internal catalogue.
PRPARK	Parks the probe. This command is just forwarded to the probe module but is not relevant to VISTA.
PRSTOP	Stops the probe tracking. This command is just forwarded to the probe module.
PRTOGS	Moves probe to next guide star in internal catalogue. Does a search of the external catalogue if there are none left.
SELGS	Selects guide star from internal catalogue. VISTA can use the VLT implementation but this command is not used.
SETUP	Processes setup keywords
SRCGS	Searches a catalogue for suitable guide star.
STAR2PR	Moves the telescope to centre the star on the probe by sending an OFFSAG command to **trkwsServer**.
STARTAG	Starts autoguiding by sending a STARTAG command to **agServer**.
STRTCCD	Should start both CCDs but otherwise the same as VLT
STOPAG	Stops Autoguiding by sending a STOPAG command to **agServer.**
STOPCCD	Should stop both CCDs but otherwise the same as VLT
STRTMC	Starts manual adjustment of probe position. This command is not used and not relevant to VISTA
STOPMC	Stops manual adjustment of probe position. This command is not used and not relevant to VISTA
TEL2OBJ	Moves the telescope with an OFFSAG command to centre a point selected on a CCD image.

Table 25 Commands supported by agwsControl

In order to support the "small" offsets described in Section 2 it will be necessary to implement a new command to be called DETXY that finds the centroid of the star on an image from a guider CCD and moves the probe to centre the image in the read-out window. This command will be similar to DETGS but will move the probe by sending the probe module an OFFSAG command instead of moving the telescope.

5.3 Mode switching table.

vtagws will require three additional entries in the table of processes in the mode switching module's database.

module name	module Id
agAlt	@lvtag2:agServer
ag_tccdAlt	ccdconCI_AGNY
adapterAlt	@lvtag2:probeServer

5.4 agwsControl Process Internals

5.4.1 Module Dependencies

The **vtagwsControl** process will be implemented by inheriting the behaviour of the **agws** module. The relationship between the **vtagws** module and other TCS modules is shown in Figure 9.

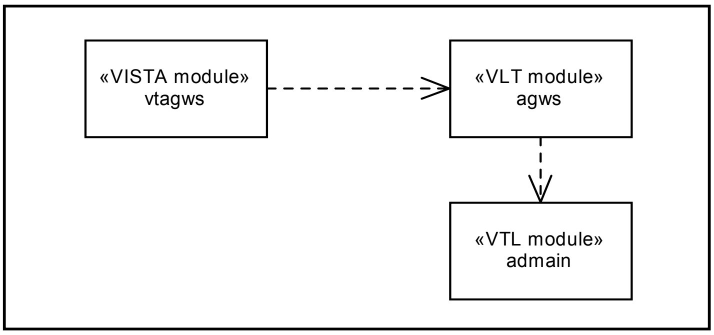

Figure 8 agwsControl module dependencies

5.4.2 Implementation

The classes that implement **vtagwsControl** and the relationships between them are shown in Figure 9 to Figure 11; for the classes specific to VISTA, the methods that the classes implement are also shown.

5.4.3 Class Specifics

5.4.3.1 vtagwsTARGET_LIST

The methods *Load* and *Select* of *agwsTARGET_LIST* will be over-ridden to load the correct destinations depending on which auto-guider the current guide star falls on and to support the retrieval of the destination for a specific CCD. The *LoadFocusData* method will be over-ridden to support the additional agServer, TCCD and adapter destinations.

5.4.3.2 vtaagwsCON_CAT

The *SearchGs* and *FilterResult* methods of *agwsCON_CAT* will be over-ridden to search for guide stars that are available to the fixed auto-guider CCDs and store the auto-guider that the star has been assigned to in the internal catalogue. *StoreCurrentGs* will be over-ridden to set the database flag in *focusData* that indicates which guider CCD is in use in addition to the functions performed by *agwsCON_CAT::StoreCurrentGs.*

SearchGsCB will be over-ridden to use ADA_POSANG instead of CCD_ON_SKY.

5.4.3.3 vtagwsCON_SERVER

The *StrtTccd*, *StopTccd* and *StopAg* methods of *agwsCON_SERVER* will be over-ridden to send commands to both TCCDs and (in *StopAg)* both agServer processes.

StartAg will be over-ridden to use ADA_POSANG instead of CCD_ON_SKY as the angle used when converting α,δ to focal plane x,y and vice-versa.

5.4.3.4 vtagwsCON_MAIN_TASK

A command handler and callback methods will be implemented to support the DETXY command.

5.4.3.5 vtagwsCON_SETUP

The *CheckWin* and *CheckBox* methods of *agwsCON_SETUP* will be over-ridden because the dimensions of the guider TCCD is hard-wired into the code of **agws** and the VISTA auto-guider CCDs are a different size.

5.4.3.6 vtagwsCON_ADAPTER

The *ProbeAbsCB* method of *agwsCON_ADAPTER* will be over-ridden to set the guider CCD window and the auto-guiding reference point before sending the PRABS command to the probe.

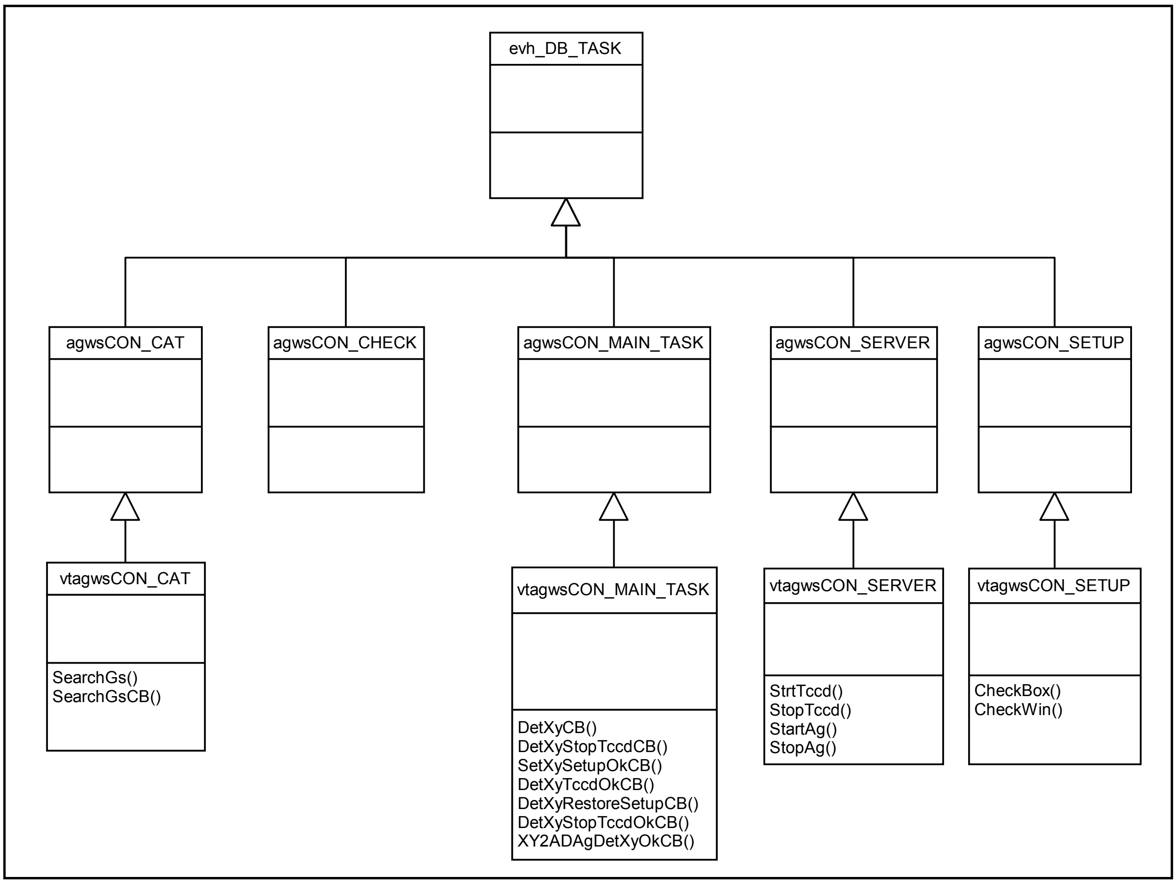

Figure 9 vtagwsControl classes

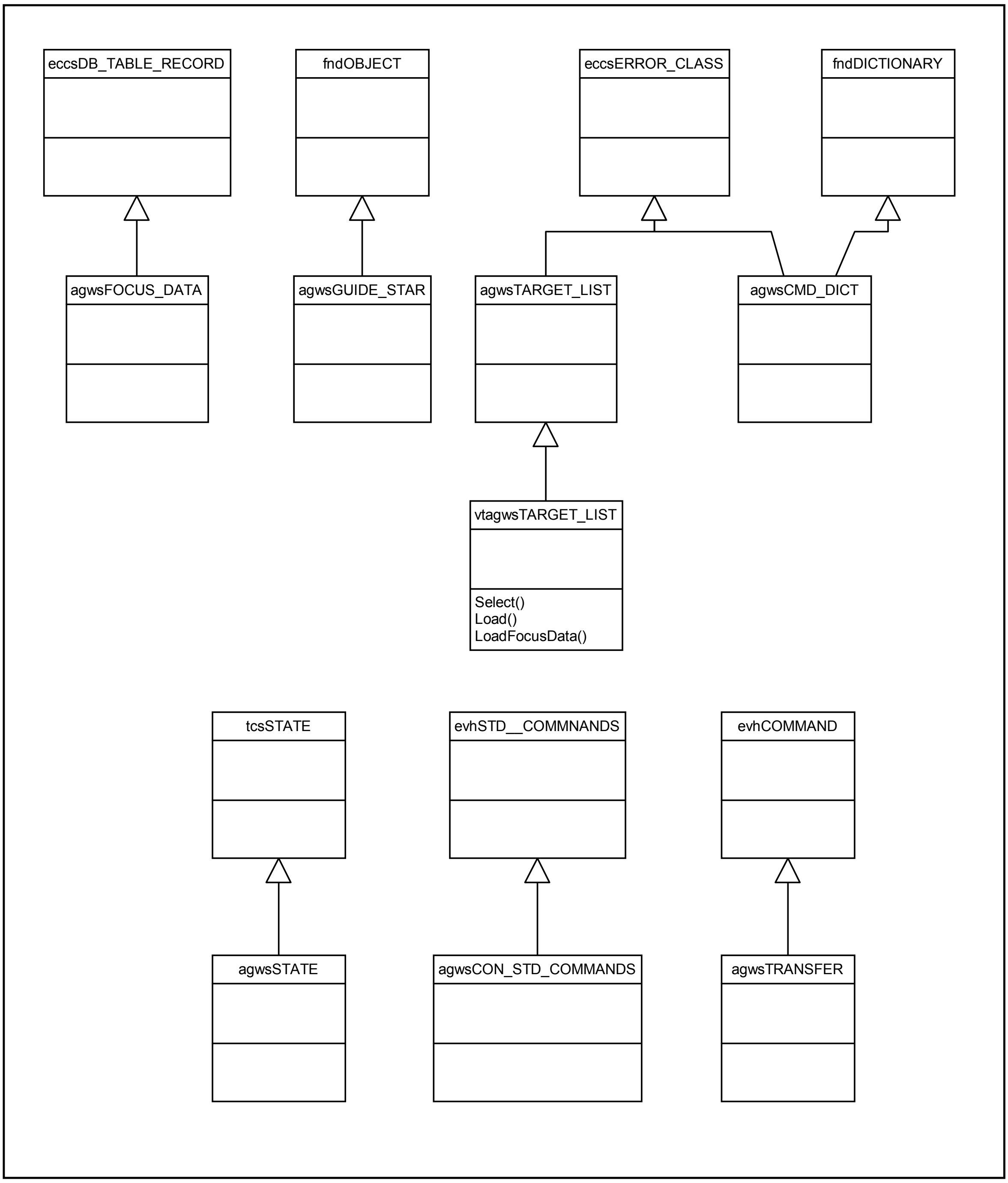

Figure 10 vtagwsControl classes

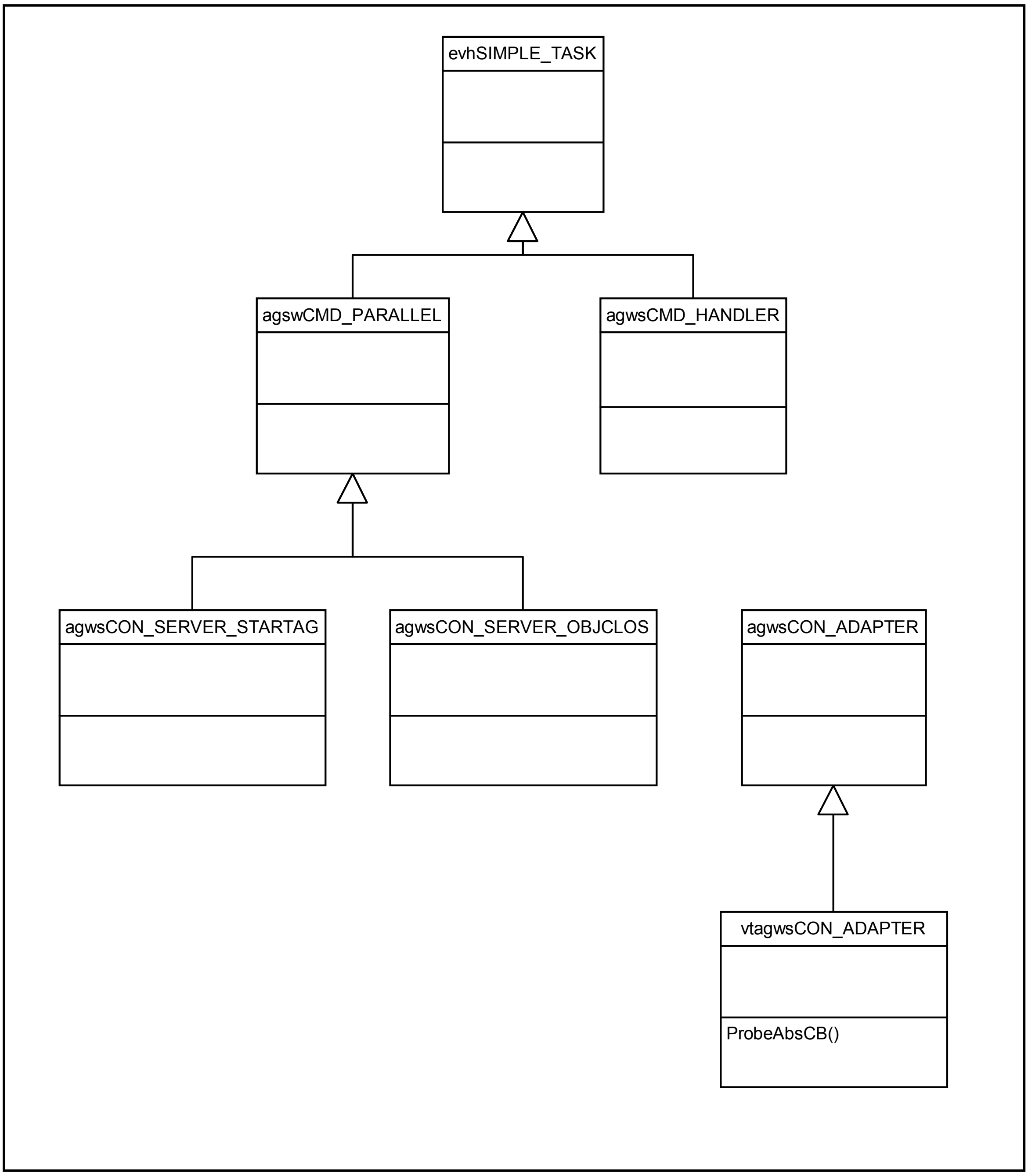

Figure 11 vtagwsControl classes

5.4.4 Database layout

The vtagws module will define additional database points for storing:

- The geometry and position in the focal plane of the two guider CCDs.

- The process names of the agServer, TCCD controller and probe server for the second guider.
- The plate scale and distortion coefficient required for converting between a guide-star α,δ and focal-plane x,y.
- Which of the two auto-guiders is in use.

These additional points are listed in Table 26.

Name (*Appl_data:TCS:agws:control:*)		***Class*/Type**	**Description**
		vtagwsCONTROL	
ccd1		*vtagwsCCD*	
	.useable	boolean	Flag that disables use of this CCD
	.xorigin	double	x coordinate of CCD in focal plane (mm)
	.yorigin	double	y coordinate of CCD (mm)
	.xsize	int32	width of CCD (pixels)
	.ysize	int32	height of CCD in (pixels)
	.angle	double	orientation of CCD (deg)
	.pixscale	double	pixel scale of CCD (pixels/mm)
ccd2		*vtagwsCCD*	
	.useable	boolean	Flag that disables use of this CCD
	.xorigin	double	x coordinate of CCD in focal plane (mm)
	.yorigin	double	y coordinate of CCD (mm)
	.xsize	int32	width of CCD (pixels)
	.ysize	int32	height of CCD (pixels)
	.angle	double	orientation of CCD (deg)
	.pixscale	double	pixel scale of CCD (pixels/mm)
focusData	.useAltGuider	Boolean	Use alternate guider
focusData:agServerAlt		*agwsPROCESS_ENTRY*	destination for alternate agServer
focusData:agTccdAlt		*agwsPROCESS_ENTRY*	destination for alternate TCCD
focusData:adapterAlt		*agwsPROCESS_ENTRY*	destination for the alternate probe server
optics		*vtagwsOPTICS*	

Name (*Appl_data:TCS:agws:control:*)		*Class*/Type	Description
	.plateScale	double	plate scale (arcsec/mm)
	.distcoeff	double	distortion coefficient

Table 26 VISTA specific vtagwsControl database points

5.4.5 Commands Issued

All the commands sent by the **agwsControl** processes to other TCS processes are listed in Table 27 to Table 30.

Command	Description
SETUP	Configure TCCD
START	Start exposure
WAIT	Wait for exposure to complete

Table 27 Commands sent by vtagwsControl to TCCD control processes

Command	Description
OFFSAD	Offset target α,δ

Table 28 Commands sent by vtagwsControl to trkwsControl

Command	Description
GPINPOS	Wait for probe to be in position
OFFSAD	Offset probe in α,δ.
OFFSGUV	Offset probe in U,V
PRABS	Move probe to absolute position (α,δ or x,y)
PRALT	Move probe to alternate position
PRCHK	Check that guide star is valid
PRCNT	Move probe to field centre
PRDOBJ	Offset the probe
PRGS	Select guide star and define CCD window
PRPARK*	Park the probe
PRSTOP	Stop virtual probe movement
SETVIGN	Set vignetting parameters
STRTMC*	Start manual probe movement
STOPMC*	Stop manual probe movement

Table 29 Commands sent by vtagwsControl to probeServer processes

Command	Description
CONFIG	Handle configuration parameters
OFFSAG	Offset telescope in α,δ

Command	Description
OFFSAG2	Offset telescope in x,y
PIXSIZE	Read pixel size
STARTAG	Start auto-guiding
STOPAG	Stop auto-guiding
XY2AD	Convert x,y to α,δ

Table 30 Commands sent by vtagwsControl to agServer processes

6 Sequence Diagrams

6.1 *Slews and Offsets*

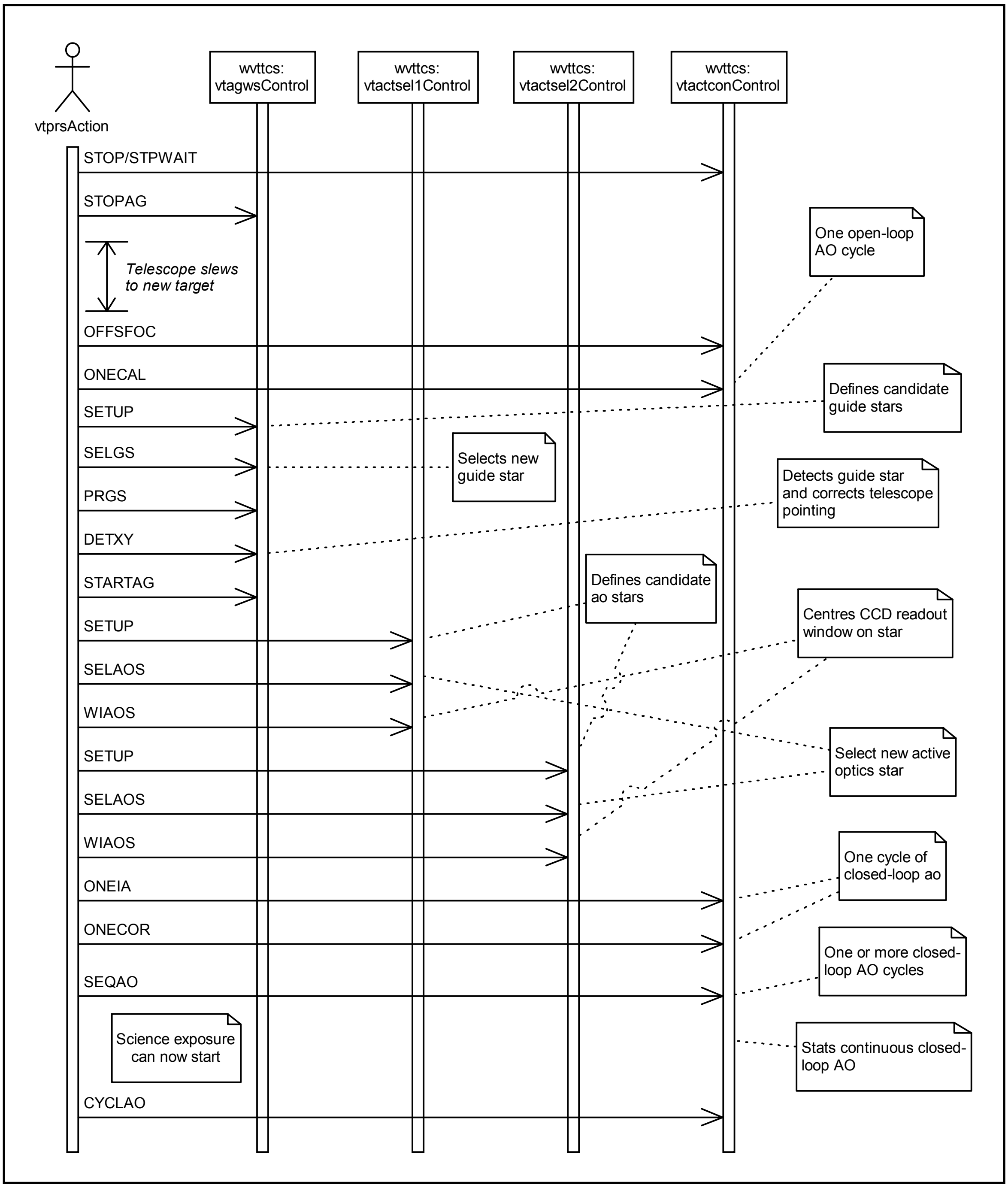

Figure 12 Long Slew

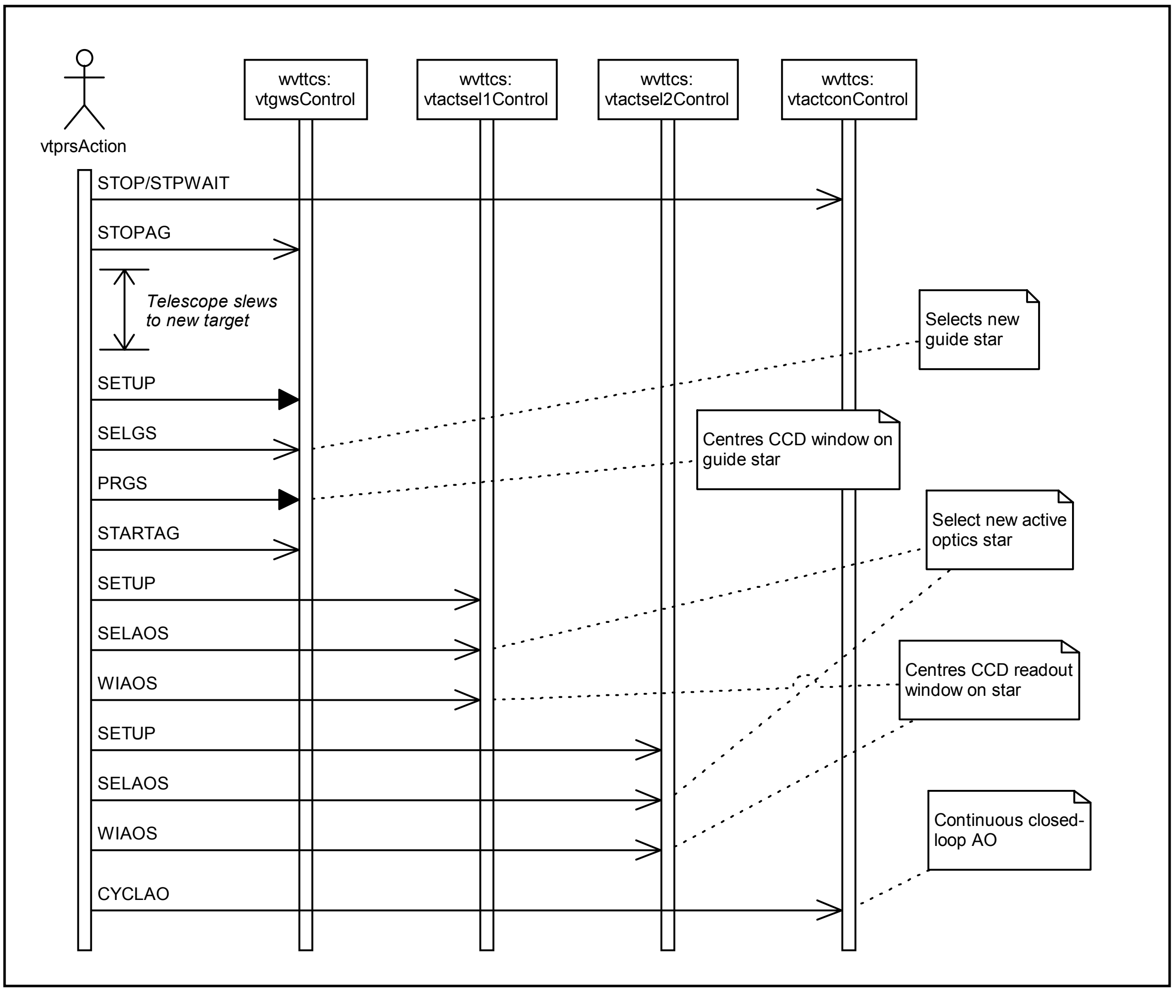

Figure 13 Offset

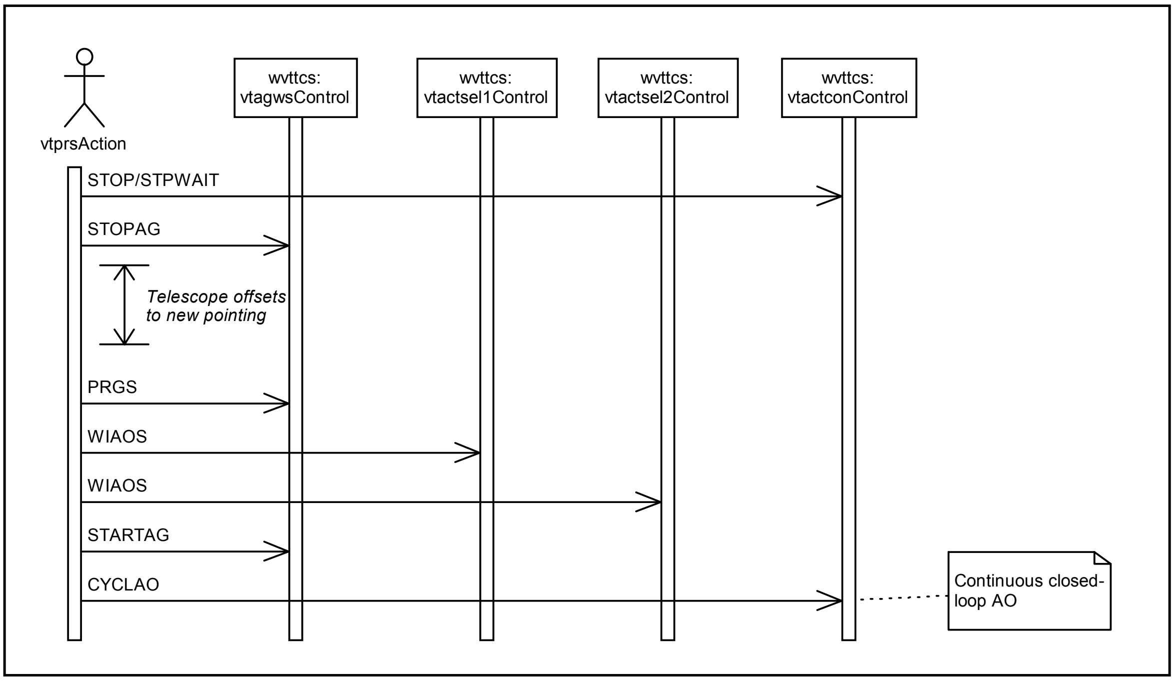

Figure 14 Dither

6.2 Active Optics Process Interactions

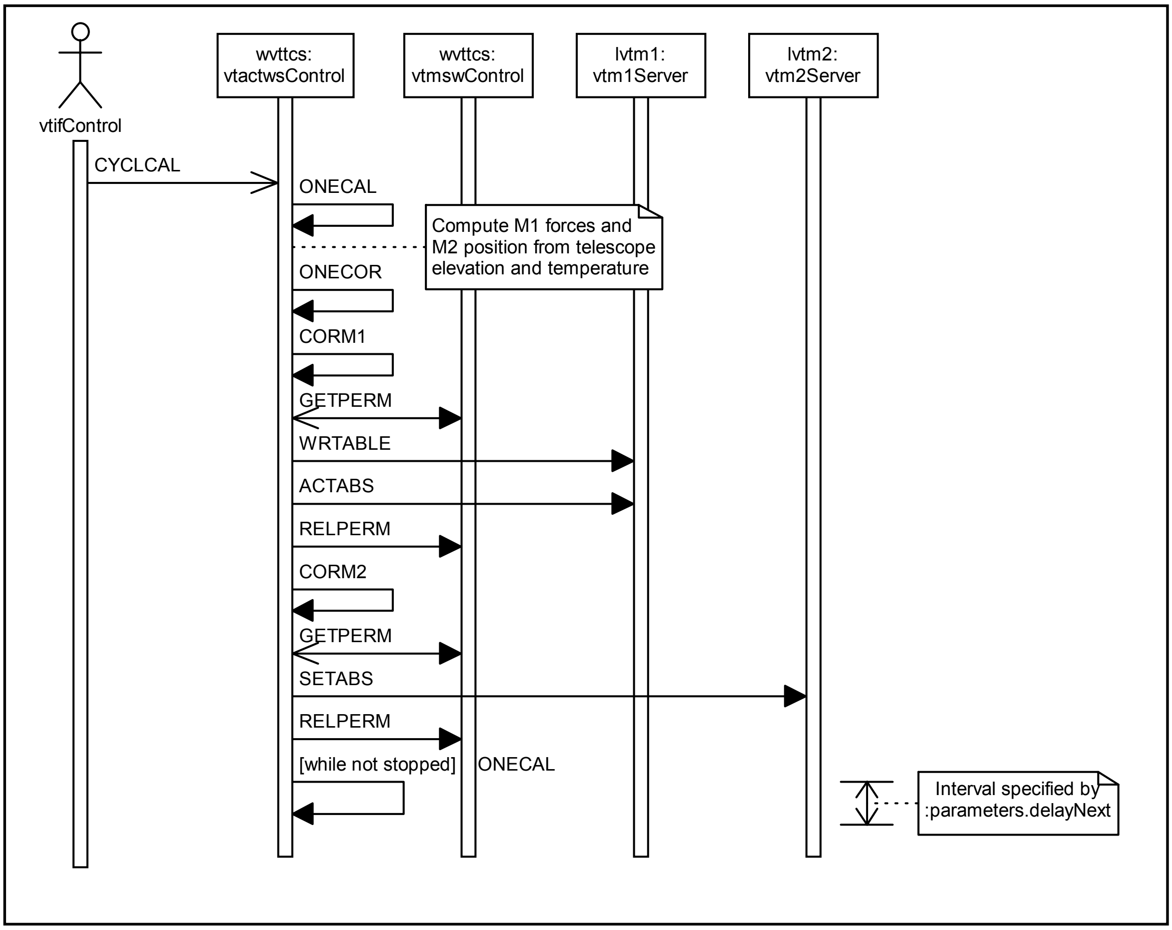

Figure 15 Open Loop Active Optics

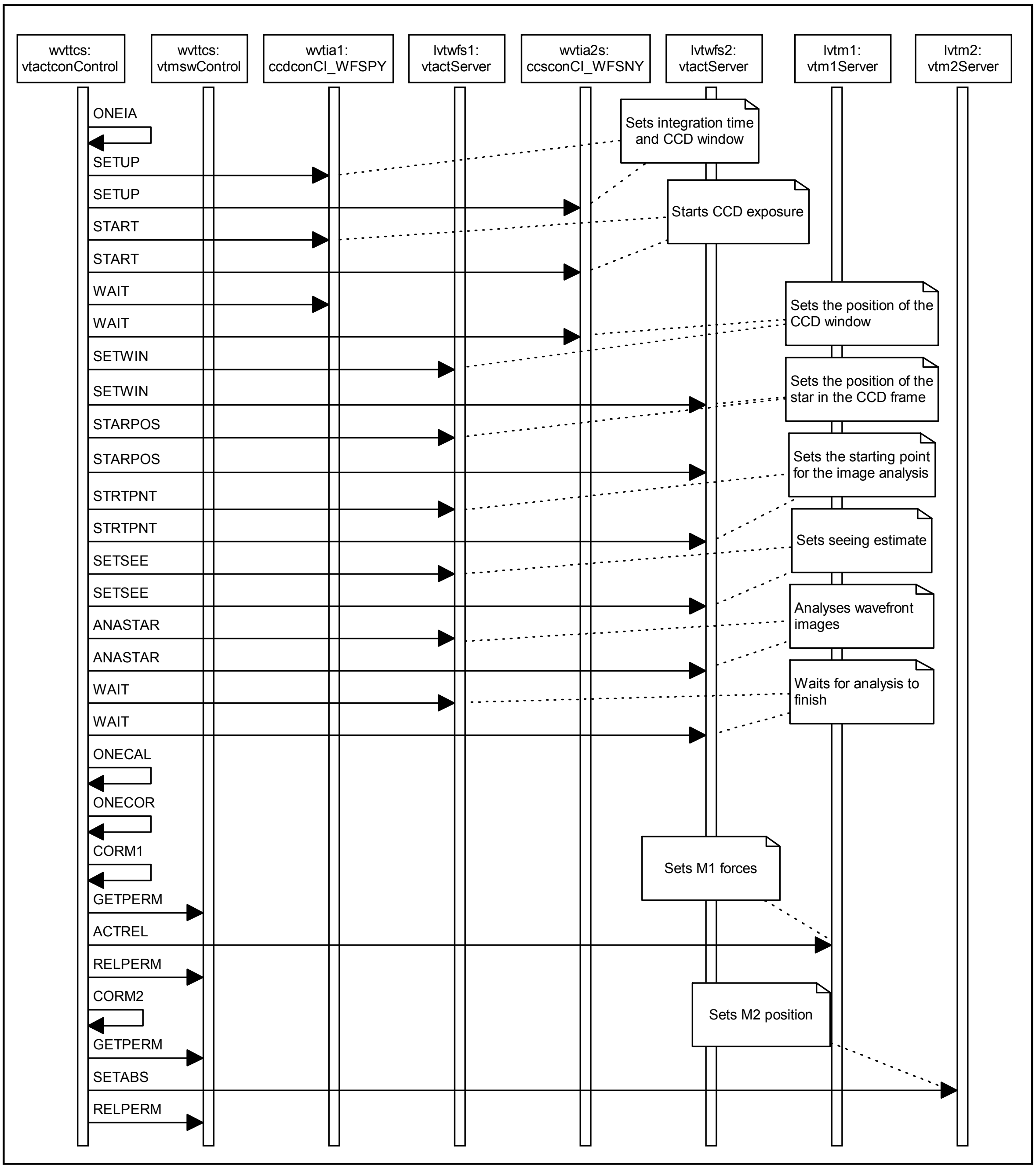

Figure 16 One Cycle of aO

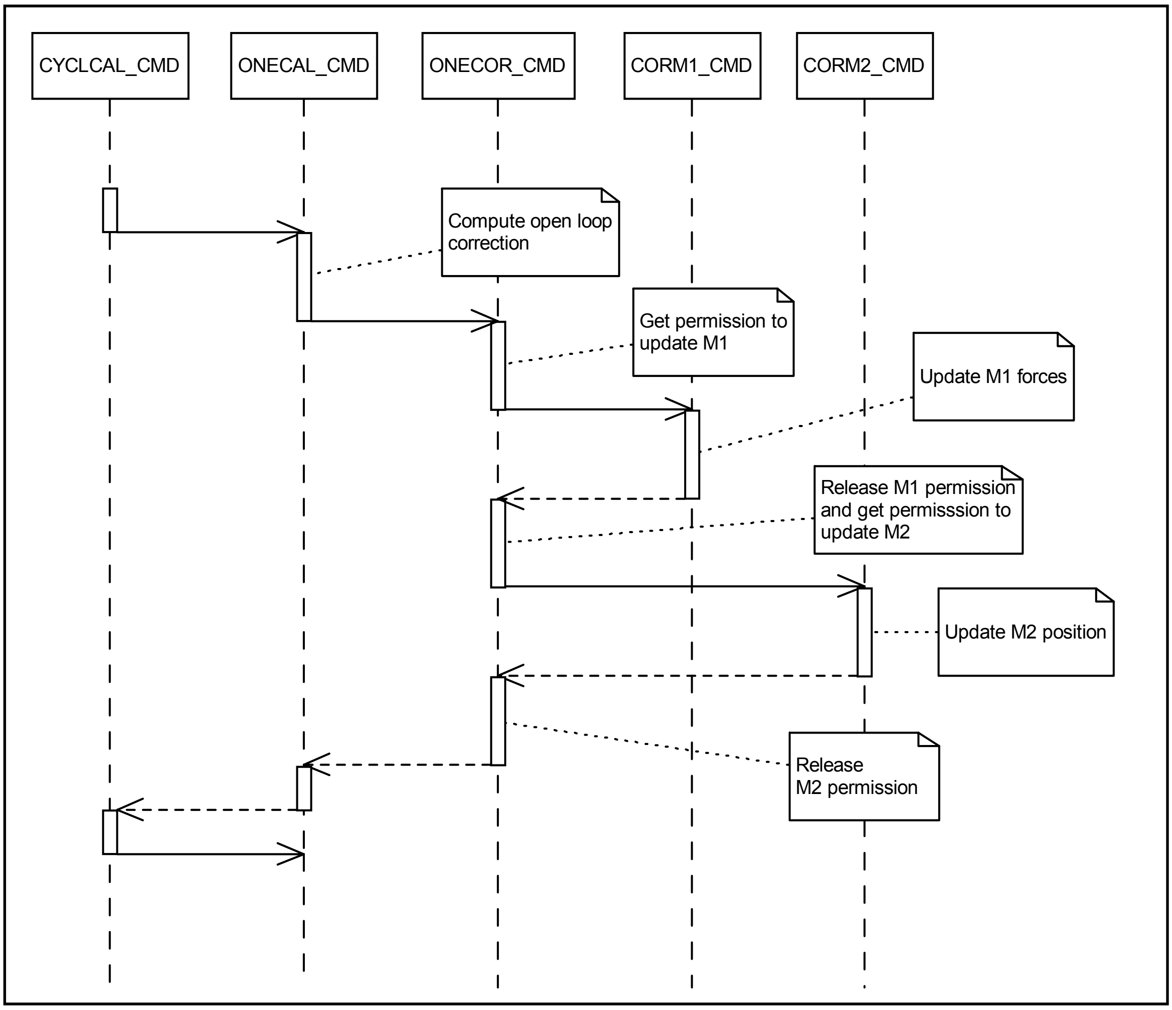

Figure 17 Open-loop Active Optics

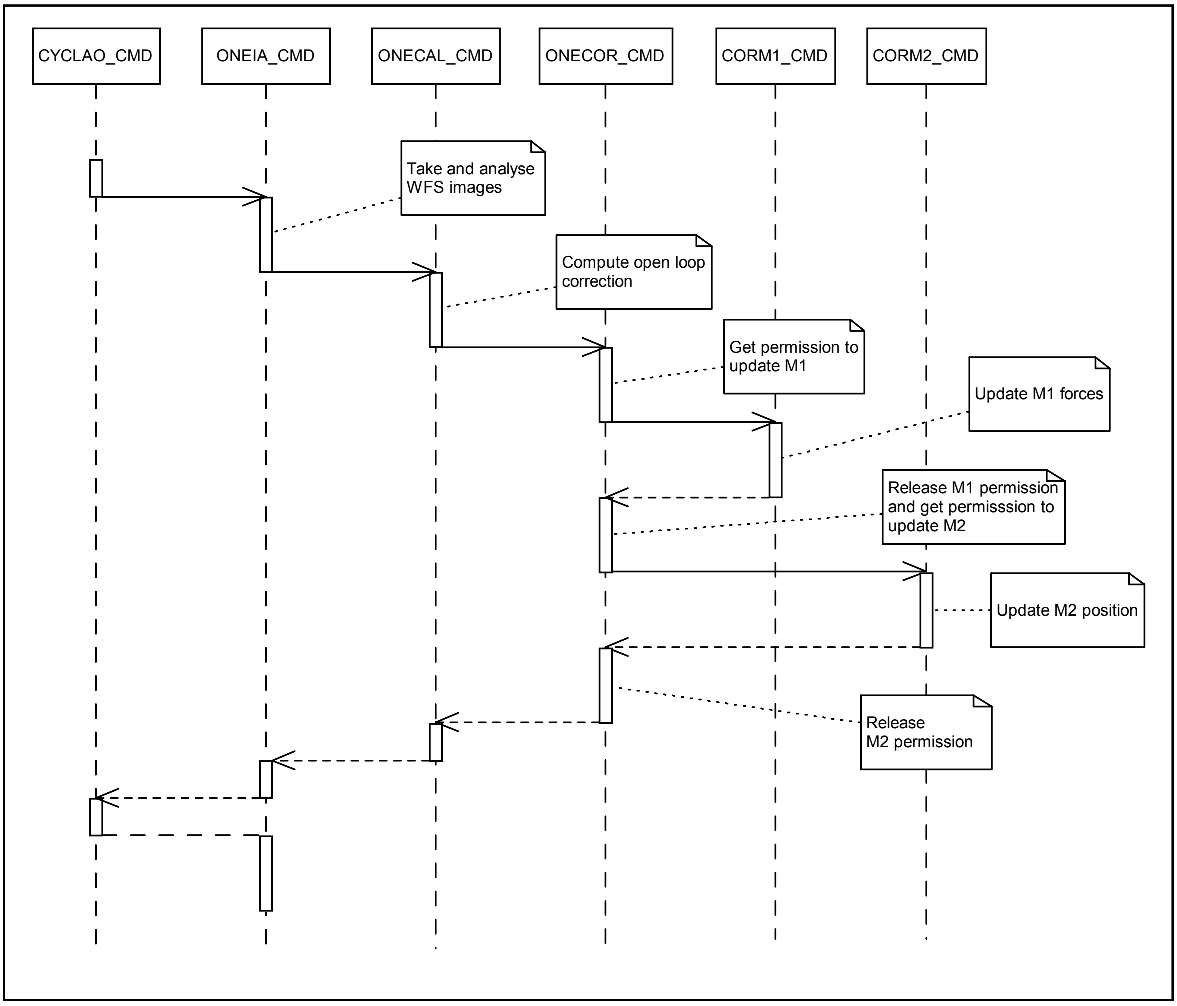

Figure 18 Closed-Loop Active Optics

7 Active optics Database

Name			***Class*/Type**	**Description**
:parameters				
		.delayNext	int32	Open loop update interval
		.maxM1Force	double	Maximum force on an M1 support
		.minM1Force	double	Minimum force on an M1 support
:calcparams				
	:lowfsSanity	.param	Vector(5,double)	Coefficients of the LOWFS sanity check function
	:m2CalibCoeffs			
		.param	Table(5,6)	M2 look-up table coefficients
	m2ZeroPoints			
		.param	Vector(5,double)	M2 zero point adjustments
	:m2ConvMatrix			
		.param	Vector(5,double)	Zernike to M2 pos conversion matrix
	:m1CalibCoeffs			
		.param	Table(18,6)	M1 look-up table coefficients
	:m1ZeroPoints			
		.param	Vector(18,double)	M1 zero point adjustments
	:m1CorrectionParams			
		.order	Vector(10,int)	Order of modes
		.symmetry	Vector(10,int)	Symmetry of modes
		.ringpar	Table(10,5)	Zernike to M1 force conversion matrix
	:nullAberrations			
		.pylowfs	Table(10,3)	Null aberration coefficients for LOWFS +Y
		.nylowfs	Table(10,3)	Null aberration coefficients for LOWFS -Y

Name			*Class*/Type	Description
	:m1SupportParams			
		.startAngle	Vector(4,double)	azimuthal angle of 1st support in ring
		.supportsPerRing	Vector(4,int)	Number of supports per ring
:coeffs				
	:m2Interpol			
		.param	Vector(5,double)	Interpolated M2 coefficients
	:m1Interpol			
		.param	Vector(18,double)	Interpolated M1 coefficients
	:filtered			
		.param	Table(5,7)	Filtered wavefront measurements
:corrections				
	:nullAbs1			
		.angle	Vector(7,double)	Null aberrations for LOWFS +Y
		.modulus	Vector(7,double)	
	:nullAbs2			
		.angle	Vector(7,double)	Null aberrations for LOWFS –Y
		.modulus	Vector(7,double)	
	:m2Absolute			
		.param	Vector(5,double)	Absolute M2 positions
	:forcesAbsolute			
		.param	Vector(84,double)	Absolute M1 forces
	:m2Diff			
		.param	Vector(5,double)	Relative M2 positions
	:forcesDiff			
		.param	Vector(84,double)	Relative M1 forces
:measure				
	:wavefront			

Name			*Class*/Type	Description
		.angle		Combined wavefront measurement
		.modulus		
		.state		
	.ia1dbRoot		BYTES256	Location of LOWFS1 result
	.ia2dbRoot		BYTES256	Location of LOWFS2 result
:kalman[10]				
	.controlActive		Boolean	
	.filterActive		Boolean	
	.controlP		double	
	.controlI1		double	
	.controlI2		double	
	.sigmaXpInit		Vector(5,double)	
	.sigmaXvInit		Vector(5,double)	
	.sigmaXp		Vector(5,double)	
	.sigmaXv		Vector(5,double)	
	.sigmaNoise		Vector(5,double)	
:focus				
	.offset		double	Current focus offset (um)
	.scale		double	Focus to wavefront scale factor
	.offsetMin		double	Minimum focus offset (um)
	.offsetMax		double	Maximum focus offset (um)
:targetMsw			*actconTARGET_MSW*	
	.destination		BYTES32	mode switching process name
	.timeoutMswGetPerm		int32	timeout for GETPERM command
	.timeoutMswRelPerm		int32	timeout for RELPERM command

[10] There are 5 Kalman filters, three for M2 and two for M1 astigmatism and trefoil.

Name		**Class/Type**	**Description**
:targetM1		*actconTARGET_M1*	
	.destination	BYTES32	m1 server process name
	.timeoutM1Actrel	int32	timeout for ACTREL command
	.timeoutM1Actabs	int32	timeout for ACTABS command
	.timeoutM1Wrtable	int32	timeout for WRTABLE command
	.vltTelescope	Boolean	true if telescope is VLT
:targetM2		*actconTARGET_M2*	
	.destination	BYTES32	m2 server process name
	.timeoutM2Setpos	int32	timeout for SETABS command
:targetIa		*actconTARGET_IA*	
	.timeoutIaAnastar	int32	timeout for ANASTAR command
	.timeoutIaWait	int32	timeout for WAIT command
	.timeoutIaStrtpnt	int32	timeout for STRPNT command
	.timeoutIaStarpos	int32	timeout for STARPOS command
	.timeoutIaSetsee	int32	timeout for SETSEE command
:targetIa1		*actconTARGET_IA1*	
	.destination	BYTES32	actServer process name
:targetIa2		*actconTARGET_IA2*	
	.destination	BYTES32	actServer process name
:targetCcd		*actconTARGET_CCD*	
	.timeoutCcdSetup	int32	timeout for SETUP command
	.timeoutCcdStart	int32	timeout for START command
	.timeoutCcdWait	int32	timeout for WAIT command
:targetCcd1		*actconTARGET_CCD1*	
	.destination	BYTES32	LOWFS #1 CCD control process name
:targetCcd2		*actconTARGET_CCD2*	
	.destination	BYTES32	LOWFS #2 CCD control process name

Name			***Class*/Type**	**Description**
:targetOwn			*actconTARGET_OWN*	
		.destination	BYTES32	own process name
		.timeoutOwnInitAo	int32	timeout for INITAO command
		.timeoutOwnInitIa	int32	timeout for INITIA command
		.timeoutOwnInitCor	int32	timeout for INITCOR command
		.timeoutOwnTermCor	int32	timeout for TEMCOR command
		.timeoutOwnStpWait	int32	timeout for STPWAIT command
		.timeoutOwnShutDwn	int32	timeout for SHUTDWN command
		.timeoutOwnCorM1	int32	timeout for CORM1 command
		.timeoutOwnCorM2	int32	timeout for CORM2 command
		.timeoutOwnOneCor	int32	timeout for ONECOR command
		.timeoutOwnOneCal	int32	timeout for ONECAL command
		.timeoutOwnOneIa	int32	timeout for ONEIA command
		.timeoutOwnStop	int32	timeout for STOP command
		.timeoutOwnSeqIa	int32	timeout for SEQIA command

Table 31 Database points defined by vtactconControl

8 Command Definition Tables

8.1 vtactconControl

```
//*******************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: vtactconControl_ZEROPOINTS.cdt,v 0.11 2005/10/05 12:09:20 vltsccm Exp $"
//
// who       when      what
// --------  --------  ----------------------------------------------
// dterrett  27/05/04  created
//
//********************************************************************************

COMMAND=  INCM2ZP

SYNONYMS=

FORMAT= A

PARAMETERS=
    PAR_NAME=    focus
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    xtilt
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    ytilt
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    xcoma
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    ycoma
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Increment M2 look-up table zero points.

Increments the zero point adjustments for the open-loop look-up tables for the
five M2 axes.
@
//-------------------------------------------------------------------------------

COMMAND=  SETM2ZP

SYNONYMS=

FORMAT= A

PARAMETERS=
    PAR_NAME=    focus
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0
```

```
    PAR_NAME=    xtilt
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    ytilt
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    xcoma
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    ycoma
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Set M2 look-up table zero points.

Sets the zero point adjustments for the open-loop look-up tables for the
five M2 axes.
@
//-------------------------------------------------------------------------------

COMMAND=  INCM1ZP

SYNONYMS=

FORMAT= A

PARAMETERS=
    PAR_NAME=    mode0
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode1
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode2
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode3
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode4
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode5
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode6
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode7
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0
```

```
    PAR_NAME=    mode8
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode9
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode10
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode11
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode12
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode13
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode14
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode15
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode16
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode17
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Increment M1 look-up table zero points.

Increments the zero point adjustments for the open-loop look-up tables for the
eighteen M1 modes.
@
//------------------------------------------------------------------------------

COMMAND=  SETM1ZP

SYNONYMS=

FORMAT= A

PARAMETERS=
    PAR_NAME=    mode0
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode1
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode2
```

```
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode3
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode4
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode5
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode6
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode7
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode8
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode9
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode10
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode11
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode12
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode13
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode14
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode15
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode16
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

    PAR_NAME=    mode17
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
```

```
Set M1 look-up table zero points.

Sets the zero point adjustments for the open-loop look-up tables for the
eighteen M1 modes.
@
//-------------------------------------------------------------------------------
//*******************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: vtactconControl_FOCUS.cdt,v 0.11 2005/10/05 12:09:20 vltsccm Exp $"
//
// who       when      what
// --------  --------  ----------------------------------------------
// dterrett  06/09/04  created
//
//********************************************************************************

COMMAND=  OFFSFOC

SYNONYMS=

FORMAT= A

PARAMETERS=
    PAR_NAME=    offset
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Set focus offset.

Sets the offset in microns from the nominal focus.
@
//-------------------------------------------------------------------------------
```

8.2 *vtactselControl*

```
//-------------------------------------------------------------------------------
//  ESO VLT              TCS vtactsel Module
//-------------------------------------------------------------------------------
//  Version    Date           Author                   Comment
//-------------------------------------------------------------------------------
//  1.0         14/01/04       Vito Graffagnino          created
//              27/01/04       Vito Graffagnino          split into sub-files
//-------------------------------------------------------------------------------
//  "@(#)$Id: vtactselControl_PUB.cdt,v 0.8+ 2005/10/26 12:42:54 vltsccm Exp $"
//-------------------------------------------------------------------------------

PUBLIC_COMMANDS

//========================================================================
// vtactsel specific commands
//========================================================================

COMMAND=  CANDAOS

FORMAT= A

PARAMETERS=
    PAR_NAME=    alpha
    PAR_UNIT=    HMS               //HHMMSS.TTT
    PAR_TYPE=    REAL
    PAR_RANGE=   INTERVAL MIN=0;MAX=240000.0
    PAR_DEF_VAL= 0.0
```

```
    PAR_NAME=    delta
    PAR_UNIT=    DMS             // [+-]DDMMSS.TTT
    PAR_TYPE=    REAL
    PAR_RANGE=   INTERVAL MIN=-900000.0;MAX=900000.0
    PAR_DEF_VAL= 0.0

    PAR_NAME=    magnitude
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0.0

    PAR_NAME=    wavelen
    PAR_UNIT=    nm
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 650.0

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Add the given AO star to the internal catalogue.

@

//-----------------------------------------------------------------------

COMMAND=  GETAOS

FORMAT= A

PARAMETERS=


REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    alpha
    PAR_UNIT=    HMS             //HHMMSS.TTT
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0.0

    PAR_NAME=    delta
    PAR_UNIT=    DMS             // [+-]DDMMSS.TTT
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0.0

    PAR_NAME=    magnitude
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0.0

    PAR_NAME=    tryState
    PAR_TYPE=    STRING

HELP_TEXT=
This returns the active-optics star position and magnitude
@

//-----------------------------------------------------------------------

COMMAND=  GETAOSS

FORMAT= A

PARAMETERS=

REPLY_FORMAT= A
```

```
REPLY_PARAMETERS=
    PAR_NAME=    alpha1
    PAR_UNIT=    HMS                //HHMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    delta1
    PAR_UNIT=    DMS                // [+-]DDMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    magnitude1
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL

    PAR_NAME=    tryState1
    PAR_TYPE=    STRING

    PAR_NAME=    alpha2
    PAR_UNIT=    HMS                //HHMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    delta2
    PAR_UNIT=    DMS                // [+-]DDMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    magnitude2
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL

    PAR_NAME=    tryState2
    PAR_TYPE=    STRING

    PAR_NAME=    alpha3
    PAR_UNIT=    HMS                //HHMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    delta3
    PAR_UNIT=    DMS                // [+-]DDMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    magnitude3
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL

    PAR_NAME=    tryState3
    PAR_TYPE=    STRING

    PAR_NAME=    alpha4
    PAR_UNIT=    HMS                //HHMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    delta4
    PAR_UNIT=    DMS                // [+-]DDMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    magnitude4
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL

    PAR_NAME=    tryState4
    PAR_TYPE=    STRING

    PAR_NAME=    alpha5
    PAR_UNIT=    HMS                //HHMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    delta5
    PAR_UNIT=    DMS                // [+-]DDMMSS.TTT
    PAR_TYPE=    REAL

    PAR_NAME=    magnitude5
```

```
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL

    PAR_NAME=    tryState5
    PAR_TYPE=    STRING

HELP_TEXT=
Returns the contents of the internal  AO star catalogue.
For each AO star the same parameters as accepted by the CANDAOS
command are returned.
@

//-----------------------------------------------------------------------

COMMAND=  SELAOS

FORMAT= A

PARAMETERS=
    PAR_NAME=    starnum
    PAR_TYPE=    INTEGER
    PAR_RANGE=   INTERVAL MIN=1; MAX = 5
    PAR_DEF_VAL= 1

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Select which of the AO stars in the list of candidates should be used
for active optics and moves it on top of the list
@

//-----------------------------------------------------------------------

COMMAND=  SRCAOS

FORMAT= A

PARAMETERS=
    PAR_NAME=    noDefCriteria
    PAR_TYPE=    LOGICAL
    PAR_DEF_VAL= FALSE

    PAR_NAME=    minMag
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 8

    PAR_NAME=    maxMag
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 16

    PAR_NAME=    vignettingLimit
    PAR_TYPE=    STRING
    PAR_RANGE=   ENUM "NONE", "INSTRUMENT"
    PAR_DEF_VAL= "NONE"

    PAR_NAME=    takeActualPos
    PAR_TYPE=    STRING
    PAR_RANGE=   ENUM "FALSE", "TRUE", "AUTO"
    PAR_DEF_VAL= "AUTO"

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
```

```
    PAR_DEF_VAL= "OK. Query guide-star catalogue"

HELP_TEXT=
Query standard guide-star catalogue (name defined in database) for AO stars
and put them in the candidates list, sorted by priority.

Optionally, specific search criteria can be given, otherwise the according
setup values are used.

<minMag> and <maxmag> are the minimum and maximum brightness in mag of the
guide-stars extracted from the catalog.
@
//------------------------------------------------------------------------

COMMAND=  WIABS
FORMAT= A
PARAMETERS=
    PAR_NAME=    type
    PAR_UNIT=
    PAR_TYPE=    STRING
    PAR_RANGE=   ENUM "AD" , "XY"
    PAR_DEF_VAL= "AD"

    PAR_NAME=    alpha
    PAR_UNIT=    HMS
    PAR_TYPE=    REAL
    PAR_RANGE=   INTERVAL MIN=0.0;MAX=240000.0
    PAR_DEF_VAL= 0

    PAR_NAME=    delta
    PAR_UNIT=    DMS
    PAR_TYPE=    REAL
    PAR_RANGE=   INTERVAL MIN=-900000.0;MAX=900000.0
    PAR_DEF_VAL= 0


    PAR_NAME=    X
    PAR_UNIT=    PIXELS
    PAR_TYPE=    REAL
    PAR_RANGE=   INTERVAL MIN=0.0;MAX=2048.0
    PAR_DEF_VAL= 0

    PAR_NAME=    Y
    PAR_UNIT=    PIXELS
    PAR_TYPE=    REAL
    PAR_RANGE=   INTERVAL MIN=0.0;MAX=2048.0
    PAR_DEF_VAL= 0

REPLY_FORMAT= A
HELP_TEXT=
Set AO star to a given position. The position can be Ra/Decl (Apparent
coordinates, Epoch 2000) or X/Y in the CCD coordinate system in CCD pixels.

alpha/delta: probe position for type AD

X/Y: probe position for type XY

@

//------------------------------------------------------------------------

COMMAND=  WI2STAR

FORMAT= A

PARAMETERS=

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
```

```
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Centres the TCCD window on the reference point

@
//----------------------------------------------------------------------

COMMAND=  WITOAOS

FORMAT= A

PARAMETERS=
    PAR_NAME=    noDefCriteria
    PAR_TYPE=    LOGICAL
    PAR_DEF_VAL= FALSE

    PAR_NAME=    minMag
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 8

    PAR_NAME=    maxMag
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 16

    PAR_NAME=    vignettingLimit
    PAR_TYPE=    STRING
    PAR_RANGE=   ENUM "NONE", "INSTRUMENT"
    PAR_DEF_VAL= "NONE"

    PAR_NAME=    takeActualPos
    PAR_TYPE=    STRING
    PAR_RANGE=   ENUM "FALSE", "TRUE", "AUTO"
    PAR_DEF_VAL= "AUTO"

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK. Window centred on guide star"

HELP_TEXT=
Select the next suitable AO star in the internal catalogue and
centres the CCD window on it.
An automatic catalogue search for an AO star is done if the list is empty.
The AO stars are put in the internal catalogue for AO star
candidates and the window is positioned onto the most suitable one.
Optionally, specific criteria can be given. If not the setup values are used.
@


//----------------------------------------------------------------------

COMMAND=  WIAOS

FORMAT= A

PARAMETERS=

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Select the next suitable AO star in the internal catalogue and
```

```
and set the CCD window If no more un-tried entry is available
an error is returned.
@


//-----------------------------------------------------------------------
// Setup commands
//-----------------------------------------------------------------------

COMMAND=  SETUP

FORMAT= A

PARAMETERS=
    PAR_NAME=     expoId
    PAR_TYPE=     INTEGER
    PAR_OPTIONAL=YES

    PAR_NAME=     file
    PAR_TYPE=     STRING
    PAR_OPTIONAL=YES
    PAR_MAX_REPETITION=999

    PAR_NAME=     function
    PAR_TYPE=     STRING
    PAR_OPTIONAL=YES
    PAR_MAX_REPETITION=999

    PAR_NAME=     noMove
    PAR_TYPE=     LOGICAL
    PAR_OPTIONAL=YES
    PAR_DEF_VAL= FALSE

    PAR_NAME=     check
    PAR_TYPE=     LOGICAL
    PAR_DEF_VAL= FALSE

REPLY_FORMAT = A
REPLY_PARAMETERS=
    PAR_NAME=     expoId
    PAR_TYPE=     INTEGER

    PAR_NAME=     done
    PAR_TYPE=     STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Handle a new setup configuration
@




//-----------------------------------------------------------------------
// TCCS Pass-through commands
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------

COMMAND=  STRTCCD

FORMAT= A

PARAMETERS=
    PAR_NAME=     winMode
    PAR_TYPE=     STRING
    PAR_RANGE=    ENUM "FULL" , "WIN"
    PAR_DEF_VAL= "FULL"
```

```
REPLY_FORMAT = A
REPLY_PARAMETERS=
    PAR_NAME=     done
    PAR_TYPE=     STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
This is just a pass-through command to the TCCD software.
Start a loop of TCCD exposures with the current TCCD setup configuration
in full screen or window mode.
@

//----------------------------------------------------------------------

COMMAND=  STOPCCD

FORMAT= A

PARAMETERS=

REPLY_FORMAT = A
REPLY_PARAMETERS=
    PAR_NAME=     done
    PAR_TYPE=     STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
This is just a pass-through command to the TCCD software.
Stop TCCD exposures, typically startd with STRTCCD.
@



//======================================================================
// Standard commands
//======================================================================

COMMAND=  STATE

FORMAT= A

PARAMETERS=

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=     done
    PAR_TYPE=     STRING
    PAR_DEF_VAL= "ONLINE"

HELP_TEXT=
Get state of vtactsel module.
@

//----------------------------------------------------------------------

COMMAND=  PING

FORMAT= A

PARAMETERS=

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=     done
    PAR_TYPE=     STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Get lifesign from ecsws module.
@
```

```
//----------------------------------------------------------------------


COMMAND=  STATUS

SYNONYMS= getStatus, GSTS

FORMAT= A

PARAMETERS=
    PAR_NAME=    function
    PAR_TYPE=    STRING
    PAR_OPTIONAL=YES
    PAR_MAX_REPETITION=999

    PAR_NAME=    global
    PAR_TYPE=    LOGICAL
    PAR_DEF_VAL= FALSE

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "State: ONLINE - Default CMD reply"

HELP_TEXT=
Get complete status of vtactsel module.
@

//----------------------------------------------------------------------

COMMAND= STOP

SYNONYMS=

FORMAT= A

PARAMETERS=
    PAR_NAME=    function
    PAR_TYPE=    STRING
    PAR_OPTIONAL= YES
    PAR_MAX_REPETITION=999

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Stop any on-going operation.
@









// __oOo__
```

8.3 vtagwsControl

```
//***********************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: vtagwsControl_PUB.cdt,v 0.9 2005/10/27 15:33:32 vltsccm Exp $"
//
// who       when      what
// --------  --------  ----------------------------------------------
// dterrett  08/03/04  First version
//
//***********************************************************************************


//========================================================================
// VISTA specific commands
//========================================================================

//------------------------------------------------------------------------

COMMAND=  DETXY

FORMAT= A

PARAMETERS=
    PAR_NAME=    noDefCriteria
    PAR_TYPE=    LOGICAL
    PAR_DEF_VAL= FALSE

    PAR_NAME=    minMag
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 8

    PAR_NAME=    maxMag
    PAR_UNIT=    magnitude
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 16

    PAR_NAME=    windowSize
    PAR_UNIT=    pixel
    PAR_TYPE=    INTEGER
    PAR_DEF_VAL= 150

    PAR_NAME=    intTime
    PAR_UNIT=    msec
    PAR_TYPE=    REAL
    PAR_DEF_VAL= 0.001

REPLY_FORMAT= A
REPLY_PARAMETERS=
    PAR_NAME=    done
    PAR_TYPE=    STRING
    PAR_DEF_VAL= "OK"

HELP_TEXT=
Detect the best object in the detection field and take it as guide star,
if acceptable. Then it centers the CCD window. Optionally, specific criteria
can be given. If not the setup values are used.
@

// __oOo__
```

__oOo__