Vane end control software user manual

COLLABORATORS

	TITLE : Vane end control software user manual		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Charlie Hull, Greg Bredthauer, Silvia Baeza, José Miguel Soto, and Felipe Sánchez	April 29, 2012	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
0	April 2003	First draft	GB,CH
1	April 2008	First release	SB,JMS
2	October 2009	Port to asciidoc	FS
3	June 2010	• Add several undocumented commands. • Correct several outdated command descriptions. • Add docinfo file with attributions. • Fix document generation scripts.	FS
4	Oct 2010	• ESC key now causes IRs to emergency stop • ESTOP and UNSTOP commands	FS
5	Dec 2010	• Fix direction of Z axis	FS
6	Dec 2010	• Add ports to coordinates diagram	FS

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
7	Apr 2012	• Increase presicion on M2 tilts reporting through EDS.	FS

Contents

1 Overview

The secondary mirror of Magellan telescopes is supported by a cage. This cage is in turn supported by four vanes, each of which terminates on the top end ring. Each “vane end”, where a vane connects to the top end ring, is composed of a pair of levers connected to each other, the vanes, and the top end ring by pivots. Rotation of one of the levers about its pivot results in a predominately axial motion; the other lever mostly controls radial motion, though there is some interplay. The levers are driven by stepper motors connected to screw drives. There are a total of eight motors, two at each vane end; one for axial motion and one for radial motion. Each lever is also connected to a linear encoder, so there are also eight encoders, four axial and four radial. Finally, there are eight load cells that measure the tension on each axial and radial lever. Together, these systems allow the secondary mirror to be positioned down to a micron. In addition, the Vane End program also controls all instrument rotators.

Controlling all of this hardware is the software described in this manual. The software is required to perform a number of operations:

- Accept high level commands (both locally at the keyboard and remotely from higher level computer systems) to position the mirror and instrument rotators, and convert them to low level commands for the various motor controllers
- Report the position and status of the secondary mirror (both on the local display and to higher level computer systems)
- Monitor the vane end tensions and encoder readings to insure accurate positioning and avoid any damage to the system
- Report the position and status of all instrument rotators, and command motion

1.1 Devices

There are a number of devices attached to the vane end control computer, which executes the VANE.EXE program:

- Two internal ISA cards, the Galil DMC motor controllers: one card controls all radial motion, the other controls all axial motion. Each of these cards has an I/O port assigned to it, which VANE communicates with. These cards are attached to amplifiers that actually drive the motors. They are also attached to the Mitutoyo linear encoders, and read the encoder positions. Finally, these cards have auxiliary inputs and outputs for communicating with the Emergency Stop Card.
- Mycom servo amplifiers: There is one of these for each of the eight motors. The Galil DMC motor controllers send these clockwise or counterclockwise pulses, which the amplifier translates into currents to move the motors appropriately.
- Mitutoyo linear encoders (gages): There is one of these for each of the eight levers. They are read by the Galil DMC motor controllers, and provide a readout good to one micron. Note that they are relative encoders, not absolute encoders, necessitating a pressure controlled retraction/extension system for homing the encoders.
- Tension Gages/DGH modules: There are eight tension gages (four axial, four radial), each of which is attached to a DGH module that provides an RS-485 interface for reading the tensions from the vane end computer. All of these modules are strung on an RS-485 line that is connected to COM 2 on the vane end computer. Each RS-485 device listens for a prompt character and an address character (see document “Proposed RS-485 Control System Communications Protocol” by D.M. Carr, doc #96CY0007). All DGH modules look for a **\$** as a prompt character. Here is a table of the DGH address characters in our application:

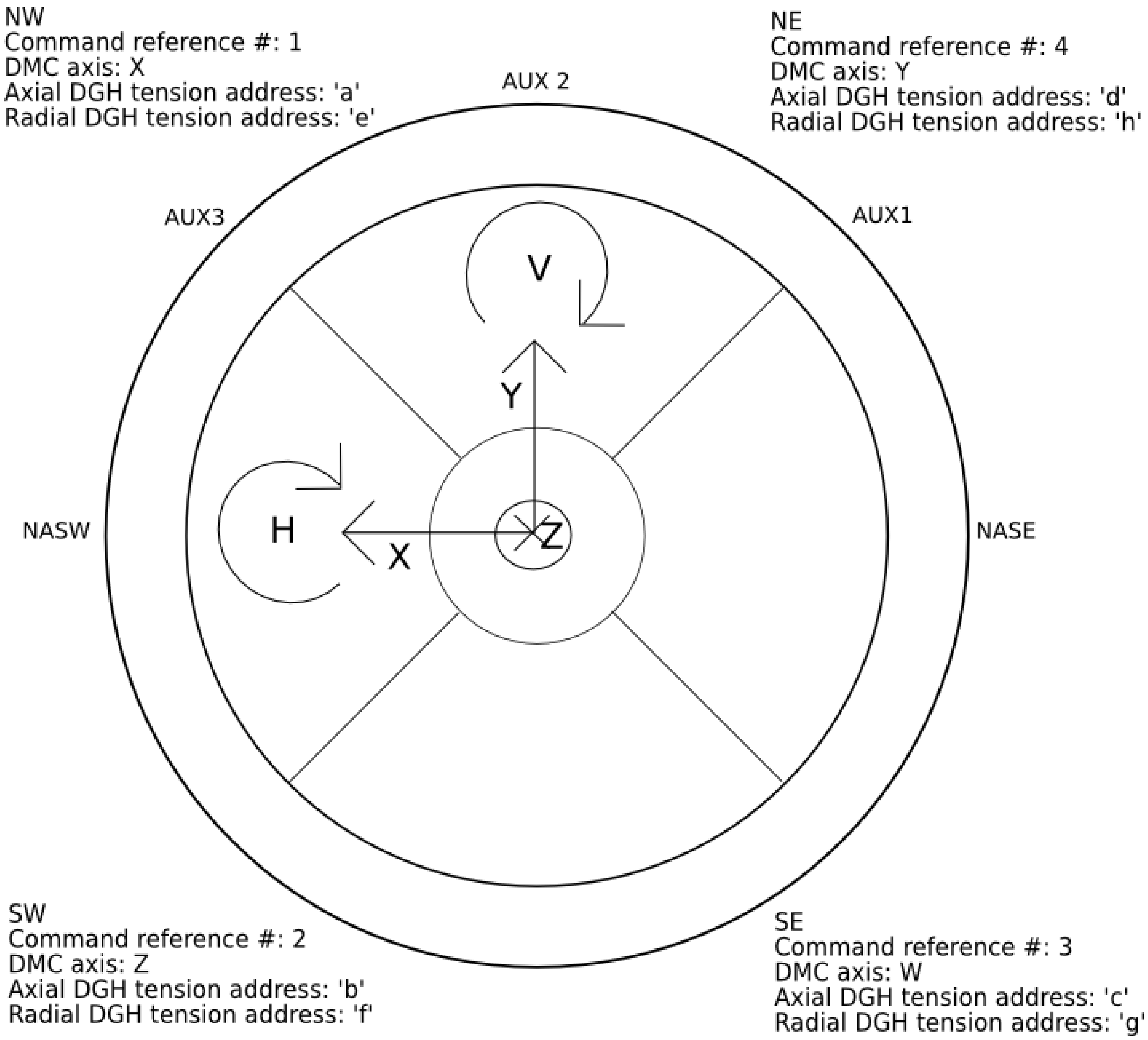
DGH Module	Axial	Radial
NW	a	e
SW	b	f
SE	c	g
NE	d	h

- Host computer: The vane end computer takes commands on where to position the secondary mirror in global coordinates, and reports on the mirror’s current location. The vane end computer is attached via an RS-485 connection on COM 1 to a higher level host computer (such as the TCS, the Telescope Control System). The vane end computer’s prompt character, address character, and response character on this communications line is set in VANE.INI.
- Instrument rotator Galil DMC Motor Controllers: Each instrument rotator is controlled by an associated Galil motor controller.

The motor controllers drive motor amplifiers to move the instrument rotator, and read each rotator's encoder, index marks, and limits. Each is connected to the vane end computer via serial lines.

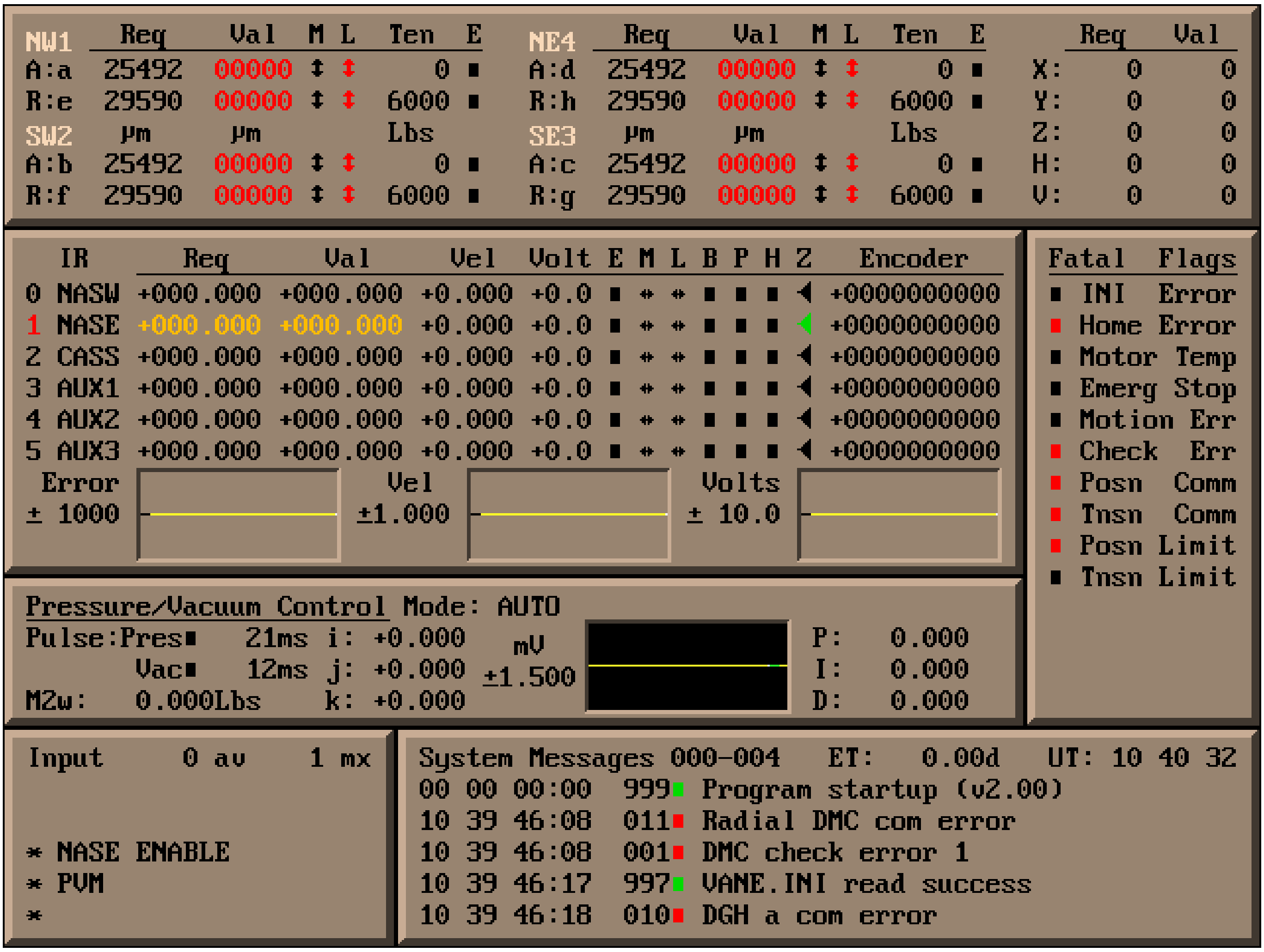
1.2 Coordinates

In the discussion of software commands following later, a few coordinate systems and labeling assumptions will be used. First, the four corners of the top end ring (NW,SW, etc) are referred to using a reference number, as defined in this sketch:



This sketch assumes the telescope is pointed at the horizon, and you're looking into the telescope towards the primary mirror. The four corners are physically labeled NW, SW, SE, and NE on the top end ring , along with the five instrument ports NASE, AUX1, AUX2, AUX3 and NASW. Each corner has an associated command reference number, used in the keyboard and serial commands. The DMC axis tells which axis that corner is attached to on the Galil DMC motor controllers. The axial and radial addresses give the RS-485 addresses of the attached DGH modules. Drawn inside the top end ring, are the global coordinate axes. These are used in commands for moving the entire secondary mirror instead of one vane end. X, Y, and Z (into the plane of the page) are given in microns. H and V (for rotation around horizontal and vertical axes) are given in arc-seconds.

2 User interface



2.1 Vane end readings

The top pane gives all of the readings from the vane end system. Each of the four vane ends has a labeled section giving target encoder readings, current encoder readings, and tensions. The target (**Req**) encoder readings are the encoder values the system is currently moving to. The current (**Val**) encoder readings display the present output from the axial and radial encoders, in microns. The tension readings give the current tensions each vane end is experiencing in pounds (this can be changed to mV for diagnostic purposes with the **P** command). All of these readings are color coded: a black reading means that the values are well within their soft limits; a yellow reading means that the values are near their limits, and a red reading means that the soft limits have been exceeded, a fatal error condition. These limits are set in the VANE.INI file, and exist to help prevent any actual hardware damage. To the right of the encoder readings is a pair of arrow indicators. The **M** column (motion) flashes a green arrow if the associated motor is moving. The **L** column (limit) is black during normal operation, and red with an arrow if the associated limit switch has been triggered. To the right of each tension reading is the **E** (error) column, which turns red if the associated DGH's internal alarm has been triggered. Finally, the far right side of the pane displays the target (**Req**) and current (**Val**) global coordinates. **X**, **Y**, and **Z** are in microns, while **H** and **V** are in arc-seconds. These axes are described above.

2.2 Instrument rotators

The middle left pane displays everything related to the instrument rotators. Each line shows an instrument rotator's address, a four letter name (such as NASW), the requested (**Req**) and current (**Val**) position in degrees, the current velocity in degrees per second, the current output voltage to the motors (torque) in volts, a number of flags, and the current encoder reading. The address of the instrument rotator is black if communication is disabled, red if there is an error, and green if it's OK. The name of the instrument rotator is green if it has been selected for display on the charts below, if not it's black. The target and current positions are yellow if the instrument rotator has not been homed.

The various flag columns have the following meanings:

E Error flag. This is black if OK and red if the Galil has flagged a motion error. This occurs if the servo error exceeds a certain limit. This usually means that the motor isn't getting power, or the brake isn't releasing, or something else is impeding commanded motion.

M Motion flag. A black double arrow means current to the motor is off. A green double arrow means current to the motor is on. A right green arrow indicates clockwise motion, and a left green arrow indicates counterclockwise motion.

L Limit flag. A black double arrow indicates that no limits have been tripped. A red right or left arrow indicates that the clockwise or counterclockwise limits are active, respectively.

B Brake flag. If black, the brake is off. If red, the brake is on.

P Lock pin flag. If black the lock pin is out, if red, the lock pin is in.

H Homed flag. If black a home operation has not been done (the program doesn't know where the instrument rotator is). If yellow, a home operation is in progress. If green, a home was completed successfully.

Z Zone flag. A right arrow indicates that the rotator is in its clockwise section of travel, and a left arrow indicates that it's in the counterclockwise zone. Each instrument rotator has about 180 degrees of travel on either side of this switch.

2.3 Fatal flags

The middle right pane is devoted to displaying the current fatal flags. Fatal flag LEDs are black if there is no error and red if an error occurred. Any fatal error that occurs halts any further movement of the secondary mirror until it is resolved. Note that the program generates a log file (if enabled in the INI file) at the first occurrence of a fatal flag for debugging use. It will have the name "mmddhhmm.log" (month/day/hour/minute.log). A description of each fatal error follows.

INI Error The program reads a text file called VANE.INI every time it is run. This file contains all of the major calibration data and soft limits, and is critical to proper operation. If the program fails to read VANE.INI (either VANE.INI was missing, or the various data fields were corrupted), this fatal flag is set. To fix the problem, you must exit the program, and either restore VANE.INI from a backup (recommended), or edit it by hand to correct any errors in it. The VANE.INI file format and contents are described in Configuration.

Home Error The first thing the program does after reading VANE.INI is home the encoders. This process involves retracting each of the encoders using the pressure system, setting the retracted position as zero, extending the encoders, re-retracting, re-extending, and verifying that the encoders are fully extending and retracting. If this process fails, this fatal flag is set. The first solution to this problem is to just re-home the encoders, using the HOME command. If the homing error re-occurs even after a few tries, it's necessary to clean the encoders (if the encoder rods get dirty, the encoders can "stick"), or check the pressure supplies and valves. Once done, re-home the encoders.

Motor Temp Each of the Galil DMC motor controllers is attached to an amplifier that drives a motor. If this fatal flag is set, one of these amplifiers is overheating. To fix it, ensure that the case enclosing the motor controllers is properly ventilated, and/or turn the motor controllers off and allow them to cool down. After they've cooled, use the **R** command to reset all fatal flags.

Emerg Stop The emergency stop card is wired to a number of hard limits, both for the encoders and for tensions. If any of these hard limits are triggered, or the watchdog isn't updated often enough, the emergency stop card will disable power to the motor amplifiers. To fix it, first try the **R** command which will reset the emergency stop card and the fatal flag. If this doesn't work, one of the hard limits was actually hit, and the emergency stop card itself must be inspected to figure out what's causing the error. The emergency stop card documentation describes what each of the emergency stop card LEDs indicates.

Motion Err This fatal flag is relatively serious. It means that the encoder values aren't changing as expected with motor movements. This can mean an error reading the encoders, or the encoders are sticking, or motors are losing steps, or the motors aren't responding to commands. Check these, and then try the **R** command to reset.

Check Err This fatal flag generally means an error occurred while communicating with the Galil cards. Check the generated log file to see what happened. Try to fix the communication problem, and then use the **R** command to reset the flag.

Posn Com An error occurred while querying the Galil cards for the encoder or limit switch readings. You can check the generated log file for details, or just use the **R** command to reset the flag.

Tnsn Com An error occurred while querying a DGH module for its tension reading. You can check the generated log file for details, or just use the **R** command to reset the flag. If the error persists, check the DGH wiring and the DGH's setup.

Posn Limit This fatal flag is set when the encoders have exceeded the soft limits. The motors can be moved back within their limits by temporarily editing the VANE.INI file using the **EDIT** command, changing the limits, moving the motors back to a safe location, and restoring VANE.INI.

Tnsn Limit This fatal flag occurs when the tensions exceed the soft limits. Try to fix this using the **R** command; if that fails, check to see if the limits really have been exceeded, or if there are communication problems with the DGH module (using a terminal). You can check DGH calibrations, or move the system back to the zeroed location (with G 0 0 0 0 0) and readjust the tensions.

2.4 Pressure vacuum control

Under the Instrument Rotator pane the status of the secondary support system is displayed. The secondary mirror support control is achieved sensing the mirror weight through three DGH modules (addresses 'i', 'j' and 'k') and then acting on two valves, one for letting the mirror rise (pressure valve) and one for lowering it (vacuum valve) according to a PID algorithm. The control can be enabled/disabled typing "PVM". The graph shows the average weight of the mirror in raw units (milli-volts).

2.5 Log interface

The bottom left pane displays the system's running log. By default, the system log only shows high level messages. However, more detailed, low-level information can be displayed using the **VERB** command to change verbose levels. At the top of the log display is a readout showing what part of the log is being displayed, how long the program has been running, and what universal time on the computer is. You can use **scroll lock** to pause and resume adding log entries for debugging purposes. The system log is a large (1000 line) circular log. You can use the cursor up and down arrow keys to scroll through a log a line at a time, **page up** and **page down** to scroll quickly, and **home** or *end *to look at the beginning or end of a log. Whenever you're looking at the end of a log, the log will automatically scroll down as new log entries are added.

The system log reports important events that have completed successfully, general problems that have occurred, and occasionally instructions or information for the user. A message time, message number, green or red indicator (good or bad), and the message text is displayed on each line. For minor or detailed messages (such as those that appear in verbose level 2), no number or color is displayed. A list of all major system log messages and a more detailed description of what the message means should be in the folder with this manual ("MSGNUM.TXT" in the development directory). If a fatal error occurs all motor movement is suspended. If "DumpLogOnFatal = 1" is in the VANE.INI file a time stamped text file is generated, with the file name "MMDDhhmm.LOG" where MM is the month, DD is the day, hh is the hour, and mm is the minute. A debugging log file can also be generated by hitting **F2** at any time, in which case the log file is named "TEMP.LOG".

The log should be examined (preferably in verbose mode = 2 or 3) any time there is a fatal error to determine the exact cause of the error.

2.6 Input interface

The bottom right pane is where user input is displayed (next to the *). At the top of this pane is another status line, which displays the average and maximum number of milliseconds it's taking for the main loop of the program to execute. Recent commands and short prompts are also displayed in this pane.

3 Commands summary

Hitting **F1** at any time will display a help screen that summarizes the available commands.

Commands that can be used locally from the keyboard are labeled **Local**. Commands that can be given remotely over the serial link are labeled **Remote**.

During normal operation, the host computer communicates with the Vane End computer via COM 1: (the serial port). The host computer will typically use the EDS (Engineering Data Stream) log entries to check for errors and get the system's current status. It will then use the move commands to update the mirror position. The **R** and **HOME** commands can be used to attempt to correct fatal errors and resume normal operation.

The vane end computer adheres to the serial communication protocol defined in D.M. Carr's "Proposed RS-485 Control System Communications Protocol", DOC#96CY0007. The vane end computer uses a prompt character of ':', an address of 'H', and a response character of '~'. These are defined in the VANE.INI file. Refer to the document "Control System Serial Communication Standards" for a description of Engineering Data Stream (EDS) communications and remote command protocols.

3.1 Vanes commands

3.1.1 X|Y|Z|H|V *n*

Move mirror to global coordinate X,Y,Z,H, or V to *n*, in absolute motion.

For example, **X 124.5**, a move is begun to the specified global coordinate (in this case, X=124.5 microns). All other coordinates are held fixed. This coordinate is relative to the mirror's absolute zero location.

3.1.2 G *x y z h v*

Move mirror to global coordinates X,Y,Z,H, and V to values *x*, *y*, *z*, *h* and *v*.

For example, **G 124.5 34.7 12.4 -19.2 0.67**, a move is begun to the specified global coordinates (in this case, X=124.5 microns, Y=34.7 microns, Z=12.4 microns, H=-19.2 arc-seconds, V=0.67 arc-seconds). These coordinates are relative to the mirror's absolute zero location.

3.1.3 DX|DY|DZ|DH|DV *n*

Move mirror on axis X,Y,Z,H, or V relative to the current position.

For example, **DX 124.5**, a move is begun in the specified coordinate relative to its current location (in this case, the mirror's X position will be increased by 124.5 microns). All other coordinates are held fixed.

3.1.4 DG *x y z h v*

Move mirror on all axes relative to the current position.

For example, **DG 124.5 34.7 12.4 -19.2 0.67**, a move is begun on all axes relative to its current location (in this case, the mirror's X position is increased by 124.5 microns, Y is increased by 34.7 microns, Z is decreased by 12.4 microns, H is decreased by 19.2 arc-seconds, and V is increased by 0.67 arc-seconds).

3.1.5 !

Stops any motion command in progress. This command gently halts any further motion. The motors will not move until another movement command is given.

3.1.6 ESC

Hitting the ESC key will inmediatelly stop any M2 and instrument rotators motion. In addition, it will raise the emergency stop condition to the vanes and the instrument rotators. You'll have to use the R command to reset the vane fatal flags and the UNSTOP command to allow the instrument rotator to move again.

3.1.7 EDIT

Edits the file VANE.INI from within the VANE program. This command enters the on-the-fly editor of VANE.INI. **ESC** exits the edit mode and returns to the VANE program.

3.1.8 IR/IA *n m*

Moves individual vane end motor #*n m* microns, either radial (IR) or axial (IA).

For example, **IR 2 -223** moves the #2 radial actuator -223 microns at its encoder. The first argument is the motor number to move, and the second number is the distance in microns at the encoder that you want to move. See the Coordinates section for the numbers assigned to each vane end. The routine will warn you if the encoder is close to its zero point, which can damage it (though the hard and soft limits should prevent this). This command should be used with caution, and should only be used for establishing the mirror's absolute zero position and for tensioning. Note that a move is limited to 100 microns (or whatever MaxManualMove is set to in VANE.INI) at a time for safety reasons.

3.1.9 A *n*

Tensions the vanes by moving all radial motors either in or out.

For example, **A 90**. This command reads VANE.INI, changes the TensionAmount entry, and writes the modified file to VANEINI.NEW. After changing this entry, a move is begun to establish this new tension value. The command **A 0** sets the tension amount to zero, which should be done when the vanes and cage are first installed. The argument to this command is the number of microns to move all radial motors in from their zeroed location. **A 2000** should establish a tension of 7000-10000 pounds at each vane end, although during installation this should be increased only slowly. For example, you should give the command **A 100** after installing and zeroing the cage, check the tensions, use the command **A 200**, check the tensions again, etc. Use this command with caution! See the section on configuration for more information. The commanded tension value can be made permanent by copying the generated VANEINI.NEW file to VANE.INI.

3.1.10 HOME

This command retracts all of the encoders, and sets the retracted position as the absolute zero position of the encoder. Several tests are performed to ensure that this zero location is accurate: the encoders are extended, retracted, and re-extended to verify that the encoders motions are repeatable and therefore accurate. After this is finished, the DMCs are initialized, and the emergency stop card is reset. The **Home Err** fatal flag is cleared, and motor motion is possible. It is vital that the encoders are properly homed before even thinking about moving the mirror.

3.1.11 CLEARFFHOME

Clears the HOME fatal flag without homing the M2 cage, in case the need arises, during engineering, of moving the cage when it could not be homed.

Caution
Use **extreme** caution in using this command. You can actually break things moving the M2 cage around without homing first.

3.1.12 R

All status flags are reset, and all fatal flags except for **INI Error** and **Home Err** are reset. The DMC's are reset and re-sent their initialization commands, and the emergency stop card is reset. This command should be used to recover from a fatal error, after the situation causing the error has been resolved. Note that the only way to fix a homing error is to redo the homing operation (**HOME** command), and the only way to recover from an error in VANE.INI is to exit the program and fix VANE.INI.

3.1.13 TE *n*

Toggles tension module *n* (a DGH) between enabled and disabled.

For example, **TE 7**, toggles a tension module between enabled and disabled. Use this command to disable communication with a DGH module/load cell pair that's malfunctioning. The tension readout will be replaced with **#** symbols, indicating that the module has been disabled. This command's argument is from 1-8. 1 = NW Axial, 2 = SW Axial. . . 4 = NE Axial, 5 = NW Radial. . . 8 = NE Radial. The example command is toggling the SE Radial tension.

3.1.14 TV/TP *n m*

Turns the pressure or vacuum valve at the specified corner on or off.

For example, **TP 3 0**, this command will turn the retraction or extension pressure on or off at a corner. Note that we no longer use vacuum, instead we use **retraction pressure**. The first number is the corner number (1 – 4), and the second number is 0 for Off and 1 for On. The example command turns extension pressure at corner 3 (SE) off. This command can be used to home the corners by hand, or to test the pressure valves. Extension and retraction pressure can not be on simultaneously. In normal operation (after the encoders have been homed), retraction pressure is off and extension pressure is on.

3.1.15 P

Toggle vane tension readings between pounds and mV.

3.2 Instrument rotators commands

In the following commands, replace *IR* by an instrument rotator denomination, i.e. **NASW**, **NASE**, **AUX1**, **AUX2**, **AUX3**.

3.2.1 *IR* ENABLE

Enables communication with the given instrument rotator.

For example, **NASW ENABLE**, will enable communication with the given instrument rotator. The number next to the instrument rotator designation onscreen should light green. If the number turns red, there are communication errors with the rotator's Galil motion controller. Communication must be enabled with an instrument rotator before further commands can be given. You should only enable communication with instrument rotators that are in use, to improve response times.

3.2.2 *IR* DISABLE

Disables communication with the given instrument rotator.

For example, **NASW DISABLE**, will disable communication with the given instrument rotator. The number next to the instrument rotator designation onscreen will turn black, indicating that communication with that instrument rotator is no longer taking place.

3.2.3 *IR* POWER

Toggle motor power on or off for the given instrument rotator.

For example, **NASW POWER**, toggles motor power on or off for the given instrument rotator. If power is off, the indicator in the instrument rotator's **M** column should be a black double arrow. If power is on, the indicator should be a green double arrow, or a green arrow going right or left if motion is in progress and power is on. This command is equivalent to sending the instrument rotator's Galil an **SH** (servo here) or **MO** (motor off) command. Note that turning motor power off and back on with this command can clear a servo error (indicated in the **E** column).

3.2.4 *IR* MO

Motor off. Turn off rotator's position servo control.

3.2.5 *IR* SH

Servo here. Turn on rotator's position servo control at the current position.

3.2.6 CASS BRAKE

Engage/release Clay's cassegrain rotator clamps.

Only Clay's cassegrain rotator has a clamping system to keep position when the M3 turret is mounted.

> **Caution**
> This command doesn't make any checks before engaging or releasing the clamp, and therefore, could potentially damage the clamp or anything that is mounted if executed at the wrong circumstances. Use it only for engeneering tasks if necessary. For normal operations use the more user friendly "CLAMP" or "RELEASE".

3.2.7 CASS CLAMP

Engage Clay's cassegrain rotator clamp.

This command should be safe to use as it checks for dangerous situations before clamping and aborts the operation if one such situation is detected.

3.2.8 CASS RELEASE

Release Clay's cassegrain rotator clamp.

This command should be safe to use as it checks for dangerous situations before clamping and aborts the operation if one such situation is detected.

3.2.9 *IR* STOP

Command given instrument rotator to stop motion.

For example, **NASW STOP**, it sends an **ST** command to the Galil for the given instrument rotator, smoothly decelerating any move in progress to a stop.

3.2.10 *IR* ESTOP

Instrument rotator emergency stop. This command will stop the rotator and will raise an emergency stop condition which will prevent any further movement command from being acccepted. To move the rotator again you'll need to use de UNSTOP command to reset the emergency stop condition.

3.2.11 *IR* UNSTOP

Resets the instrument rotator emergency stop condition, allowing it to comply with movement commands.

Caution
Before using the UNSTOP command you should fix the problem that caused the emergency stop to trigger in the first place, otherwise the instrument rotator could keep moving into a dangerous zone.

3.2.12 *IR* MOVE *n*

Move given instrument rotator the given number of steps.

A positive number is clockwise motion, and negative is counterclockwise. If a move is already in progress, the number of steps requested will be added to the current target location (the **IP** Galil command is used).

3.2.13 *IR* MOVEA *n*

Move given number of degrees.

A positive number is clockwise motion, and negative is counterclockwise. If a move is already in progress, the number of degrees requested will be added to the current target location.

3.2.14 *IR* GOA *n*

Have instrument rotator go to the given angle relative to absolute zero.

For example, **NASW GOA 123.1**, the given instrument rotator will begin a move to the specified angle. 0.0 degrees is about the middle of travel, and each instrument rotator can move +/- 180 degrees from there. Note that the displayed current and target positions will be yellow if the instrument rotator has not been homed, and black if it has. The **GOA** command will move the rotator without being homed, but the current and target positions will likely be wrong.

3.2.15 *IR* GOR *n*

Similar to GOA but the given angle is relative to the current position.

3.2.16 *IR* GOS *p v t*

Move instrument rotator to position *p* at velocity *v*, the time *t* at which the command was issued must be provided for jitter correction. Format for *t* is "HHMMSShh" (hh = hundredths of a second).

This command is intended to be issued by the TCS when tracking, for rotation control.

3.2.17 *IR* JOG *n*

Begin a jog motion of the instrument rotator at the given degrees per second.

For example, **NASW JOG 2.0**, the specified instrument rotator will begin a jog motion at the given number of degrees per second. A positive number will result in clockwise motion, negative will result in counterclockwise motion. The rotator will smoothly accelerate from it’s current position/speed to the given speed.

3.2.18 *IR* HOME

Home the given instrument rotator.

For example, **NASW HOME** commands the instrument rotator to home. Depending on whether the rotator begins in its clockwise or counterclockwise zone, it will begin moving clockwise or counterclockwise until it hits a limit or an index. If it hits a limit, it will move back to an index. Once it has done so, the program “knows” where it is. The home indicator will go green (it will be yellow while the home is in progress), and the current and target positions will go from yellow to black.

3.2.19 *IR* VEL *n*

Set the maximum velocity for moves in degrees per second.

For example, **NASW VEL 1.5**, sets the instrument rotator top speed (velocity) in degrees per second. If a move is in progress, the rotator will begin using this new speed immediately by accelerating or decelerating smoothly to the new value.

3.2.20 *IR* DISPLAY

Display given instrument rotator on the error/speed/torque charts.

For example, **NASW DISPLAY**, displays given instrument rotator on the three on-screen charts, showing error, speed, and torque. Note that communication with the rotator you want to see must be enabled in addition to using the **DISPLAY** command for the charts to update.

3.2.21 *IR* RESET

Reset communication with the given instrument rotator.

For example, **NASW RESET**, resets communication with the given instrument rotator. This includes resending all communication setup, motion profile and PID parameters to the rotator.

3.2.22 *IR* ACC|DEC|SPD *n*

Set instrument rotator motion profile variables.

For example, **NASE DEC 100000**, sets the given motion profile parameter in the instrument rotator to the number entered. The **ACC** and **DEC** commands set the acceleration and deceleration, respectively. In this case the parameter is in encoder counts/sec/sec. The **SPD** command sets the maximum speed in encoder counts/sec. To set the maximum speed in terms of degrees/sec, use the **VEL** command. The **SPD** command can be given while a move is in progress.

3.2.23 *IR* P|I|D *n*

Set P|I|D servo parameter for the given instrument rotator.

For example, **NASW D 6.0**, sets one of the instrument rotator’s PID servo control variables to the given number. All three of the variables should be set to a number between 0.1 and 10. They can be set to new values while a move is in progress, which is useful during tuning.

3.2.24 *IR* SPDP *n*

Set speed servo PID proportional constant.

3.2.25 *IR* SPDI *n*

Set speed servo PID integrative constant.

3.2.26 *IR* SPDD *n*

Set speed servo PID derivative constant.

3.2.27 *IR* SPDIL *n*

Set speed servo PID integrative limit.

3.2.28 *IR* PROG

Command to upload PID program into the controller to make it able to work with the ‘servo mode’ used for rotation tracking.

If you want to be sure that program was recorded into controller’s internal memory, do the following steps:

- Talk comport 19200, via IR terminal, where comport is from 2 to 6.
- %0LS, to see if program was recorded
- If you want to begin execution of program, do the following:
  – %0RS

3.2.29 *IR* SPDERR

Toggle display of speed servo error on leftmost chart.

3.2.30 *IR* MCURR

Toggle display of motor current on leftmost chart.

3.2.31 *IR* MESTAT

Toggle display of master encoder status on leftmost chart.

3.2.32 *IR* SESTAT

Toggle display of slave encoder status on leftmost chart.

3.3 Charts commands

3.3.1 ERROR *n*

Set the instrument rotator error chart scale.

For example, **ERROR 10000**, sets the on-screen instrument rotator error chart’s scale to the given value (in encoder counts). All old data is immediately plotted with the new scale, which ranges from –(given #) to +(given #).

3.3.2 VEL *n*

Set the instrument rotator velocity chart scale.

For example, **VEL 2.5**, sets the on-screen instrument rotator velocity chart’s scale to the given value (in degrees/second). All old data is immediately plotted with the new scale, which ranges from –(given n) to +(given n).

3.3.3 VOLT *n*

Set the instrument rotator voltage (torque) scale.

For example, **VOLT 10.0**, sets the on-screen instrument rotator voltage chart's scale to the given value (in volts). The voltage is representative of the torque the motors are supplying. Note that the Galil motor controller voltage signal is limited to +/- 10.0 Volts. All old data is immediately plotted with the new scale, which ranges from –(given #) to +(given #).

3.4 M2 support commands

3.4.1 PRES

Toggle pressure.

3.4.2 VAC

Toggle vacuum.

3.4.3 PVM

Toggle the pressure-vacuum control of the secondary mirror on or off. Note that there must be air and vacuum supplied for the control to work properly.

3.4.4 PVS *n*

Change the scale of the M2 weight graph for the pressure-vacuum control.

3.4.5 PVKP *n*

Set pressure/vacuum to Kp gain.

3.4.6 PVKI *n*

Set pressure/vacuum to Ki gain.

3.4.7 PVKD *n*

Set pressure/vacuum to Kd gain.

3.5 Message display commands

3.5.1 VERB *n*

Set system log verbose level to *n* (*n* = 0..2).

3.5.2 <Up>

Scroll system message display back one line.

3.5.3 <PgUp>

Scroll system message display back one page.

3.5.4 <Down>

Scroll system message display forward one line.

3.5.5 <PgDn>

Scroll system message display forward one page.

3.5.6 <Home>

Set system message display to show the first message.

3.5.7 <End>

Set system message display to show the current message.

3.5.8 <Scroll Lock>

Pause/Resume system log messages.

3.6 Engineering commands

3.6.1 TR|TA|TD|T3|TIR

Terminal mode with DMCs or DGHs.

Switches to a simple terminal mode with the radial DMC (TR), the axial DMC (TA), the DGHs at 19200 baud (TD) or 300 baud (T3), or the Instrument Rotator DMCs (TIR). Use **<ESC>** to leave terminal mode.

3.6.2 W

Write current location as zero to VANE.INI.

This command reads the VANE.INI file, modifies the zeroed-encoder entries to match the current encoder readings, and writes the modified file to VANEINI.NEW. All subsequent moves will be in reference to this zero position (the encoder readings that correspond to global coordinates 0,0,0,0,0). This command should be given right after the cage and vanes are installed at zero tension, in their zeroed mechanical position (i.e. axial and radial levers are at the middle of their arcs). To use the zero position permanently, copy VANEINI.NEW to VANE.INI.

3.7 Other commands

3.7.1 Q|EXIT

Exit the program.

4 Configuration

VANE.INI is the configuration file for the vane end program. It includes all of the parameters that might need to be modified. It can be edited in a simple text editor outside of the program, or it can be edited on-the-fly using the **E** (edit vane.ini), **W** (write current location as zero), and **A** (adjust tension) local keyboard commands. Whenever VANE.INI is altered by the program using these commands, the alterations are written to the file VANEINI.NEW, and immediately used in the program. This allows the changes to be tested before adopting them permanently. To use the changes you make on-the-fly permanently, you have to exit the program and copy VANEINI.NEW to VANE.INI.

VANE.INI follows a simple format: Any line beginning with a semicolon is ignored (it's a comment). All other lines are significant. On each of these lines there is a title (like Address), an equal sign, and data. Everything before the equal sign is ignored. There are three types of data: a single character (which must be enclosed in double quotes), integers, and floating point numbers. The VANE.INI file is mostly self-documented, but I'll run through it with a few more detailed descriptions and comments.

4.1 General settings

Note
In the following examples, an **i** represents an integer value, an **r** a real value, and a **c** a single character.

Warning
The order of the lines of the VANE.INI file must be preserved, as the vane end program identifies them by their position on the file and not by their title, to the left of the equal sign.

4.1.1 Telescope identification

The first line identifies the telescope that the programm is controling. **1** is Baade, **2** is Clay.

```
Telescope = i
```

4.1.2 Serial communication constants

Characters that vane end program uses to identify itself and comunicate via serial line with the upstream computer, usually, the TCS.

```
Address = "H"
Prompt = ":"
Response = "~"
```

4.1.3 Clock correction factor

The ClockPPM entry is used to speed up or slow down the system clock by the given number of parts per million per second. This is used to match the GPS-given Universal Time as closely as possible (typically to within 1/100 of a second per hour). To adjust this, allow the system to run for several hours. The TCS should send a UT update once an hour (visible on the system log). Subtract the time given in the UT update message from the vane end system's time stamp for the message. Divide the difference (in seconds) by the number of seconds since the last UT update message (typically 3600 seconds = 1 hour). Multiply this number by 1 million, and add to the value of ClockPPM already in the INI file. Example: If the system log time stamp is greater than the UT message's time by 5/100ths of a second, you should increase the value of ClockPPM by 1000000 * (0.05/3600.0), which is about 14. Average the corrections over many hours for better accuracy.

```
ClockPPM = i
```

4.1.4 Log Dump on Fatal Error

This entry controls whether or not the system log is dumped to a time-stamped file on an error. This is typically left at 0 (Off), so that the disk doesn't fill up with logs. Turn it on to try catching a rare error event in the act, particularly with the system log verbose level set to 2.

```
DumpLogOnFatal = i
```

4.2 Vanes settings

4.2.1 Zeroed encoder values

The next eight lines are the radial and axial encoder values for all four corners that correspond to the zero location of the vane ends. It's critical to set these correctly for proper motion of the cage. Setting these is discussed later in this section. Instead of altering these values by hand you can use the W command to write the current encoder values as the zero point.

```
ZS1[0] (NW Axial)  = i
ZS1[1] (SW Axial)  = i
ZS1[2] (SE Axial)  = i
ZS1[3] (NE Axial)  = i
ZS2[0] (NW Radial) = i
ZS2[1] (SW Radial) = i
ZS2[2] (SE Radial) = i
ZS2[3] (NE Radial) = i
```

4.2.2 Minimum and maximum encoder values

The next sixteen lines are soft limits for the encoder readings. The program will not allow moves if the encoders are outside these limits, and will not begin moves whose destinations are outside of these limits. Note that if the encoders are ever forced below a reading of zero (after being properly homed), the encoders will be damaged! Encoder ranges are also limited by limit switches, which should be set to be a little beyond the soft limits. If a limit switch is triggered, the DMC will only allow motion away from the limit switch.

```
MinS1[0] (NW Axial)  = i
MinS1[1] (SW Axial)  = i
MinS1[2] (SE Axial)  = i
MinS1[3] (NE Axial)  = i
MinS2[0] (NW Radial) = i
MinS2[1] (SW Radial) = i
MinS2[2] (SE Radial) = i
MinS2[3] (NE Radial) = i
MaxS1[0] (NW Axial)  = i
MaxS1[1] (SW Axial)  = i
MaxS1[2] (SE Axial)  = i
MaxS1[3] (NE Axial)  = i
MaxS2[0] (NW Radial) = i
MaxS2[1] (SW Radial) = i
MaxS2[2] (SE Radial) = i
MaxS2[3] (NE Radial) = i
```

4.2.3 Load cell calibration constants

Each of the eight load cells were calibrated by applying a range of loads, recording their outputs in mV, and fitting a second order function to convert mV to pounds. Each load cell has a k_0, k_1, and k_2 constant such that the load cell reading is $pounds = k_0 + k_1 * mV + k_2 * mV^2$.

```
T1k0[0] (NW Axial)  = r
T1k1[0] (NW Axial)  = r
T1k2[0] (NW Axial)  = r
T1k0[1] (SW Axial)  = r
T1k1[1] (SW Axial)  = r
T1k2[1] (SW Axial)  = r
T1k0[2] (SE Axial)  = r
T1k1[2] (SE Axial)  = r
T1k2[2] (SE Axial)  = r
T1k0[3] (NE Axial)  = r
T1k1[3] (NE Axial)  = r
T1k2[3] (NE Axial)  = r
T2k0[0] (NW Radial) = r
T2k1[0] (NW Radial) = r
T2k2[0] (NW Radial) = r
T2k0[1] (SW Radial) = r
T2k1[1] (SW Radial) = r
T2k2[1] (SW Radial) = r
T2k0[2] (SE Radial) = r
T2k1[2] (SE Radial) = r
T2k2[2] (SE Radial) = r
T2k0[3] (NE Radial) = r
T2k1[3] (NE Radial) = r
T2k2[3] (NE Radial) = r
```

4.2.4 Minimum and maximum tension values

The next sixteen lines are soft limits for the tension readings in pounds. The program will not allow movement commanded remotely if the DGH tension readings exceed these amounts. However, local movement will be allowed. The DGHs also have internal alarms that are triggered when their readings exceed a given value. These values are set by the vane end program. Each radial DGH is set so that its internal alarm will trigger when the tension is 10% higher than the radial maximum soft tension limit. Each axial DGH is set so that its internal alarm will trigger when the tension is 10% less than the axial minimum soft tension limit. When these alarms trigger, the emergency stop card disables power to the motors.

```
MinT1[0] (NW Axial)  = r
MinT1[1] (SW Axial)  = r
MinT1[2] (SE Axial)  = r
MinT1[3] (NE Axial)  = r
MinT2[0] (NW Radial) = r
MinT2[1] (SW Radial) = r
MinT2[2] (SE Radial) = r
MinT2[3] (NE Radial) = r
MaxT1[0] (NW Axial)  = r
MaxT1[1] (SW Axial)  = r
MaxT1[2] (SE Axial)  = r
MaxT1[3] (NE Axial)  = r
MaxT2[0] (NW Radial) = r
MaxT2[1] (SW Radial) = r
MaxT2[2] (SE Radial) = r
MaxT2[3] (NE Radial) = r
```

4.2.5 Motion correction factors

In its native configuration, the vane end software does not move the expected distances in the X Y and Z axes. It is off by a linear factor. Actual cage motion was measured, and FixXY, FixZ, FixH and FixV were set to correct axial, radial and rotation motion.

```
FixXY = r
FixZ = r
```

```
FixH = r
FixV = r
```

4.2.6 Tension amount

This value is the number of microns each vane end should be pulled away from the cage to establish tension in the vanes. A value of 2000 microns corresponds to a little less than 10000 pounds of tension in each. This value should be adjusted to zero (using the A command) before the cage is installed, and then slowly increased until the required tension is achieved.

```
TensionAmount = i
```

4.2.7 Axial and Radial error amounts

The next two lines are the maximum error the vane end system should allow at the axial and radial encoders. Typically this is set to 1, as 1 micron is the smallest measurable unit from the encoders. This value also determines how close the encoders must be to their original values during the verification step of the homing operation.

```
AxialError = i
RadialError = i
```

4.2.8 Maximum manual move

The next value determines how large of a movement will be permitted by the program for **IA** and **IR** moves. This just encourages smaller, cautious movements.

```
MaxManualMove = i
```

4.2.9 Backlash movements constants

These constants control the motion during a backlash removal movement step. Before making a macro move, the program first performs a few backlash moves, where a small movement is given and the encoders checked to see if movement occurred. BacklashMove gives the number of microns at the encoders that the move tries to accomplish. BacklashTime is the number of milliseconds the move should take (which controls the speed). BacklashTimeout is the number of milliseconds to wait for the move to finish before flagging a movement error. BacklashDelay is the number of milliseconds to wait after the move is complete (to allow the DMCs to return to normal operation).

```
BacklashMove = i
BacklashTime = i
BacklashTimeout = i
BacklashDelay = i
```

4.2.10 Macro movement constants

These constants control the motion during a macro move (large movements towards a destination). MacroMove is the maximum number of microns at the encoders to attempt to move. MacroTime is the time in milliseconds the move should take (which controls the speed). MacroTimeout is the maximum amount of time in milliseconds to wait for the move to complete before flagging an error. MacroDelay is the time in milliseconds to wait after a move is complete to allow the DMCs to return to normal operation.

```
MacroMove = i
MacroTime = i
MacroTimeout = i
MacroDelay = i
```

4.2.11 Micro movement constants

Once the encoders are close to their destination values, the vane end program executes micro moves to nudge the encoders to their final position. Each micro move moves one micron at the encoder. MicroTime is the time in milliseconds the move should take, MicroTimeout is the time in milliseconds to wait for the move to complete before flagging an error, and MicroDelay is the time in milliseconds to wait after the move for the DMCs to return to normal operation.

```
MicroTime = i
MicroTimeout = i
MicroDelay = i
```

4.2.12 Movement thresholds

These are the thresholds the program uses to decide which type of movement to use. If an encoder has to move more than MacroThreshold microns to reach the destination, a macro move is performed. If an encoder has to move more than BacklashThreshold microns, backlash removals are performed. BacklashTarget sets the location in microns before the actual final destination that the backlash move attempts to move to. MacroError is the percentage of the commanded move that the encoders must show without flagging an error (for example, 0.2 means that the encoders must move at least 20% of the commanded distance).

```
BacklashThreshold = i
MacroThreshold = i
BacklashTarget = i
MacroError = r
```

4.3 Instrument rotators settings

There are six instrument rotators sections, one for each instrument station. All of them with the same configurable settings, with different values for each of them of course.

4.3.1 PID control constants

PID control constants. These should typically be in the range of 0.0 – 10.0. To tune them, start with P = 1.0, I = 0.0, D = 0.0. Set the instrument rotator being tuned to be displayed on the on-screen charts with the *"DISPLAY"* command. Perform small moves, and increase P until it gets close to the target position pretty quickly. Decrease it if the system starts to oscillate. Increase I until residual errors decrease quickly. Try increasing D to minimize ringing after moves or during jogs.

```
P = r
I = r
D = r
```

4.3.2 Acceleration|Deceleration|Speed Settings

Sets the rate at which the instrument rotator accelerates and decelerates in encoder counts/second/second. The speed is the maximum speed of motion in encoder counts/second.

```
Acc = i
Dec = i
Spd = i
```

4.3.3 Error Limit

Maximum error in encoder counts before the Galil flags an error, which triggers the brake and halts motion. This type of an error occurs if someone tries to move the rotator by hand, or the motor isn't moving the rotator when commanded to, or the system is oscillating due to bad PID settings. This must be in the range 0-10000. It's usually set to 10000, because 10000 encoder counts is a small amount of motion. If an error of this type occurs, the *"E"* indicator will turn red on the display.

```
ErrorLimit = i
```

4.3.4 Velocity profile filter

This value is used to filter the acceleration and deceleration functions in independent moves, like **GOA**, **GOR** or **JOG**, to produce a smooth velocity profile. The resulting profile, known as S-curve, has continuous accelerartion, reducing mechanical vibrations. The value sets the bandwidth of the filter, with a maximum value os **1.0** meaning no filter and a minimum value of **0.004** meaning maximum filtering. Note that filtering results in longer motion times.

```
VelProfFilter = r
```

4.3.5 Torque limit

Set the limit on the motor command output. Maximum is * 9.998*

```
TorqueLimit = r
```

4.3.6 Revolution

Set to the number of encoder counts equal to exactly one revolution of the rotator (360.0 degrees). These should be measured accurately, as this number will control how close to a requested position the system can come.

```
Revolution = i
```

4.3.7 Home jog speed

Sets the speed at which the rotator will move towards the index mark, once the zone switch has been found during a homing procedure, in counts/sec.

```
HomeJogSpeed = i
```

4.3.8 Home

During a homing procedure, value to set the encoder to when when the index is reached.

```
Home = i
```

4.3.9 Minimum home distance

The number of encoder counts the rotator should be moved to insure an index or a limit is hit. This is used during the homing operation. An error will be generated if an index or limit isn't hit in this distance.

```
MinHomeDistance = i
```

4.3.10 Minimum back off limit distance

The number of encoder counts the rotator should be “backed-up” by after hitting a limit, and moving forwards again to hit the index.

```
MinBackOffLimitDistance = i
```

4.3.11 Servo PID control constants

PID constants, plus integral limit, for the servo program that runs inside the rotator controler and that controls the rotator movement during servo mode.

```
SpdP = r
SpdI = r
SpdD = r
SpdIL = r
```

4.3.12 Hard limits

Where, in degrees, are the clockwise and counterclockwise hard limits located

```
CWLimit = r
CCWLimit = r
```

4.3.13 Soft limit distance

How far from the hard limits should the rotator stop.

```
SLDist = r
```

4.3.14 Enabled flag

Sets whether this instrument rotator has communication enabled at program startup. 0 = disabled, 1 = enabled

```
Enabled = 0
```

4.4 Secondary mirror support settings

At the moment, it is possible to install two secondary mirrors, an F11 mirror and an F5 one. The F11 have only one pressure vacuum circuit, while the F5 have four of them. The INI file have five sections, with the same configurable parameters, for each of this five pressure vacuum circuits.

4.4.1 PID constants

PID constants for pressure vacuum control loop.

```
Kp = r
Ki = r
Kd = r
```

4.4.2 Integral limits

Integral limits for the integral part of the PID loop, to avoid wind up.

```
VIl = r
PIl = r
```

4.4.3 Integral reset threshold

Lower limit for the integral part of the PID, above which integration occurs.

```
IntegralThreshold = 15.000
```

4.4.4 Valves minimum open times

Minimum pulse time, in milliseconds, for a valve to actually open. At smaller pulses the valve frequently doesn't open at all.

```
VacuumMinPulseWidth = i
PressureMinPulseWidth = i
```

4.4.5 Mirror weight conversion factor

Convertion factor, pounds per millivolt.

```
PMfact = r
```

4.4.6 Pulse cycle

The width of PID control cycle, in milliseconds. This value is common to all five PID loops and is found only once, at the end of the five PID control loop sections.

```
PulseCycle = 200
```

4.5 Clay cassegrain positions

Cassegrain observing ports positions for pointing M3. CP values are in degrees and CE ones in encoder counts. At the moment only CP values are used.

```
CPNASW = r
CENASW = i
CPNASE = r
CENASE = i
CPAUX1 = r
CEAUX1 = i
CPAUX2 = r
CEAUX2 = i
CPAUX3 = r
CEAUX3 = i
```

5 Initial setup

Charlie Hull has written detailed documents about the actual hardware setup for the vane end system. The general steps are:

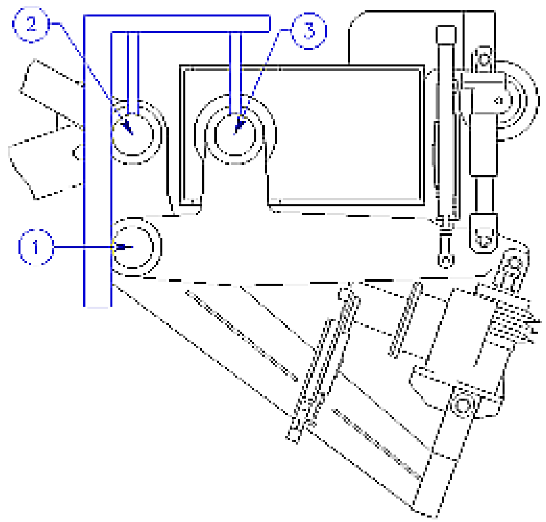
- With the cage and vanes off, adjust each vane end to its zeroed position (i.e., the center of their arcs) using Charlie Hull's reference marks. Also, adjust the position of each encoder so that it's in the middle of its range (1" or 25400 encoder counts).
- Set the tension amount entry in VANE.INI (TensionAmount =) to zero.
- Run VANE.EXE. Allow the encoders to home. Then, use the "W" local keyboard command to write this position as the zeroed encoder position. Exit, and copy VANEINI.NEW to VANE.INI to preserve this zero location.
- Calibrate tensions using cables, pulleys, and force measurements. Take enough data to generate the tension calibration values, and put them in VANE.INI.
- Attach the vanes and cage
- Run VANE.EXE. After homing the encoders, use the "A" command to slowly increase the tensions until you reach about 10000 pounds. For example, execute "A 100", then "A 200", etc. Around "A 2000" the tension should be near 10000 pounds. Exit the program and copy VANEINI.NEW to VANE.INI to preserve this tension setting.
- Run the system through its full range of motion, and set the limit switches and soft encoder and tension limits to prevent any damage. Encoder limits should have a minimum of around 5000 and a maximum of around 45000. Axial tensions should range from about –2000 to 2000 pounds. Radial tensions should be between 6000 and 12000 pounds.
- Tune the PID and speed/acceleration/deceleration parameters for any instrument rotators. Also find the number of encoder counts per 360.0, rotator zero point, and two home positions.

Charlie Hull's document is next.

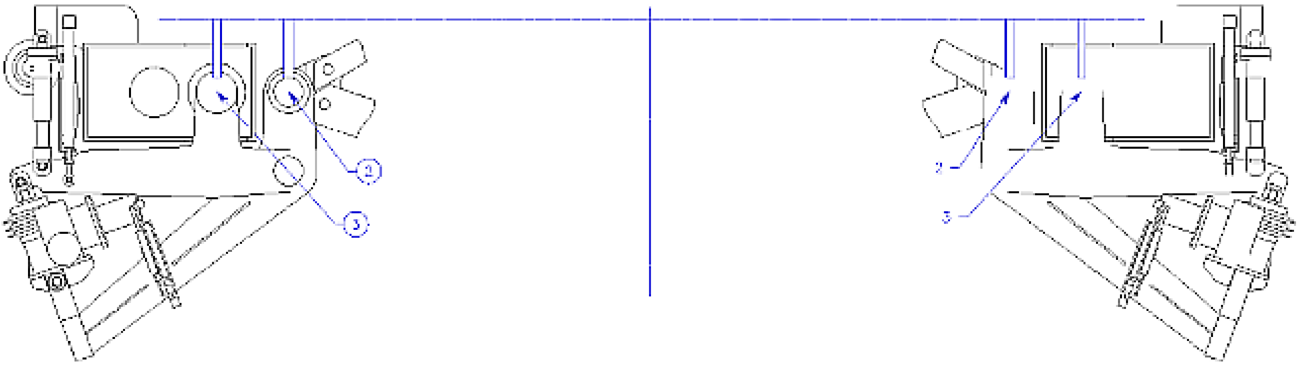
5.1 Setup of Vane End Actuators for the Magellan 6.5m telescope

For proper function of the vane end actuators the zero position needs to be known.

1. Adjust the relationship between the radial and axial actuator. This was done by placing a square on the sides of the shafts of the radial pivot (1) and the vane end pivot (2) and fixed length scales between the square and the tops of the shafts of the vane end pivot (2) and the axial pivot (3). The radial actuator was then adjusted until all the pivots lined up properly.



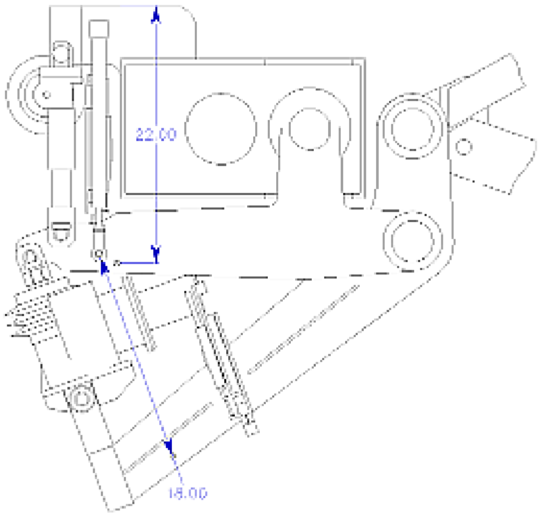
2. Adjust the relationship between opposed axial pivots. This was done using a surveying level and sighting on the axial pivots (3) and establishing the line between the two axial pivots (3). Once this line was established the position of each of the vane end pivots (2) was adjusted such that they were also on this line. The position of the vane end pivot was adjusted in the axial direction using the axial actuator. (Note that the separation between the two vane ends is not to scale)



3. Marks were made, using a center punch, to indicate these zero positions.

   One punch mark is on the axial arm near the attachment point for the preload spring. It is placed 22 inches from the top of the axial actuator support structure

   The other mark is on the side of the radial arm and is 18 inches from the cylindrical surface of the bolt holding the preload spring to the axial arm.



4. The linear actuators were then set to the middle of their range using the control system to read out the encoder positions. At zero, the encoders should read about 25000.

6 Running the program

The vane end program requieres the following files installed in a directory in order to run:

VANE.EXE The main program binary.

VIDEO16.FNT, VIDEO8.FNT, VIDEO32.FNT Fonts used for the display.

VANE.INI An user-editable text file with all configurable options.

Usually, the program is run automatically after a boot by the autoexec.bat script, but if the necessity arises (maintenance, debugging, etc.) of launching it manually, it suffices to CD to the installation directory and run “vane”.

7 Log message system

Note that for error messages with an error code, the error code is just used to find where in the program’s source code the error occurred, and is used to find bugs. The system in log in verbose mode (and with log dumps enabled) should be used in most cases to diagnose strange problems. Then, just recreate the problem/error message and hit the log dump function key.

7.1 Errors (0-799)

7.1.1 001: *DMC check error x*

On startup, the program checks the Galil DMCs' power-up outputs and corrects them, if necessary. An error means that communication with the Galils is impaired in some way. Use a Galil terminal in the program (**TA** or **TR**) to try communicating, and check that the cards are seated in the computer properly.

7.1.2 002: *VANE.INI read error x*

There was an error reading VANE.INI during startup. This could mean the file was missing, or that the program had trouble parsing it. Check to make sure it exists and that the lines are all formatted correctly and in the right order (compare to a backup known working copy).

7.1.3 003: *Encoder homing error x* or *xx corner homing error y*

There was an error homing the Mitutoyo encoders. Check encoder retraction, extension, and readings. You can retry the home operation with the **M** command. Note that the encoders MUST be homed before any vane-end motion can occur.

or

There was an error homing the encoder on the given corner. The local display will ask if you want to retry the corner, and then timeout after a short time.

7.1.4 004: *DGH init error x*

There was an error initializing one of the DGHs (COM errors will tell you which DGH). Check DGH communication and DGH setup strings.

7.1.5 005: *Destination exceeds soft limits*

The target of the move is beyond the allowable movement range set up in VANE.INI (min and max encoder values). Try a different move or extend the allowable range in VANE.INI. Be cautious of running into limits or doing damage, though.

7.1.6 006: *Move aborted: A fatal flag is set* or *Move aborted: Move error x*

Motion was aborted bacuse a fatal flag was set. Resolve the fatal error, and then try again.

or

Motion was aborted because an error occurred during the move. Recreate the error with full logging on and check the log to see what exactly went wrong.

7.1.7 007: *Encoder read error x*

There was an error reading the encoder values from a Galil. Check Galil communication.

7.1.8 008: *DMC config error x*

There was an error during the configuration of the Galil DMCs. Check Galil communication or check verbose log entries.

7.1.9 009: ***Emerg stop reset error x***

There was an error resetting the emergency stop card.

Check the emergency stop card to see which LEDs are lit red, and fix the cause (possibly DGH setups). Also check Galil DMC communication.

7.1.10 010: ***DGH x com error***

A communication error occurred with the given DGH module. If this happens only rarely it's probably just noise in the serial line.

If a bunch occur, check DGH wiring, power, setup, and check communication with a terminal.

7.1.11 011: ***Axial/Radial DMC com error***

A communication error occurred with the Axial or Radial Galil DMC. Check communication with it manually, and check the logs.

7.1.12 012: ***Axial/Radial motor timeout***

The program waited for the Galil DMC to report that a move was finished, but the Galil didn't report this in the given timeout period. Try to recreate the error with full logging on, and see if the Galil was given a bad move to do.

7.1.13 013: ***Individual move error x***

An error occurred during an individual motor move. Check the log in verbose mode to find out exactly what went wrong.

7.1.14 014: ***DGH x error messages suspended***

Error messages with the given DGH will no longer be displayed because too many occurred.

7.1.15 015: ***Axial/Radial DMC error messages suspended***

Too many consecutive com errors with the Axial or Radial Galil DMC have occurred. Further errors will not be reported until there is a successful communication.

7.1.16 016: ***Check fatal error x***

There was an error checking to see if any fatal errors exist. This is always a communication error with one of the Galils. Try to recreate the error with full logging on to see which Galil is having trouble with what.

7.1.17 017: ***Encoder read errors suspended***

There were too many encoder read errors. Further error messages will not be displayed until a successful communication occurs.

7.1.18 018: ***Move ignored, fatal errors exist***

A movement was commanded while there were fatal errors that prevent motion. Fix the fatals, and then try again.

7.1.19 019: ***Incorrectly formatted UT message from TCS***

The command to set Universal Time from the TCS was garbled. Check communications.

7.1.20 020: ***Movement aborted! (via ESC key)***

All motion has been aborted because the user hit the ESC key.

7.1.21 021: ***Host com error x***

There was a host (TCS) communications error. Put the log into verbose mode to see what is being said between the computer and the TCS. The error number given helps pinpoint where in the source code the error occurred.

7.1.22 022: ***Host error messages suspended***

The host (TCS) had too many consecutive communication errors. Further error messages will be suppressed.

7.2 Instrument rotator specific errors

7.2.1 100: ***Brake not toggled: IR not enabled/com error*** or ***Brake not toggled: Move in progress (use STOP).***

A command to toggle an instrument rotator's brake was ignored for the given reason. To correct, enable IR communication, fix any communication problems, or stop current motion.

7.2.2 101: ***Free command ignored: IR not enabled/error***

The command to free an instrument rotator for motion by hand wasignored because the given instrument rotator doesn't have communicationenabled or is having communication errors.

7.2.3 102: ***Stop ignored: IR not enabled***

The command to stop an instrument rotator was ignored because the given instrument rotator doesn't have communication enabled.

7.2.4 103: ***Power not toggled: IR not enabled/com error*** or ***Power not toggled: Move in progress.***

The command to toggle an instrument rotator's power was ignored for the given reason. To correct, enable IR communication, fix any communication problems, or stop current motion.

7.2.5 104: ***Move ignored: IR not enabled/error*** or ***Move ignored: Brake on*** or ***Move ignored: Another move is being processed***

The command to move the instrument rotator was ignored for the given reason. To correct, enable IR communication, fix any communication problems, release the brake, or wait for the current move command to finish getting sent to the IR and try again.

7.2.6 105: ***Home ignored: IR not enabled/error*** or ***Home ignored: Brake on*** or ***Home ignored: Another move is being processed.***

The command to home the instrument rotator was ignored for the given reason. To correct, enable IR communication, fix any communication problems, release the brake, or wait for the current move command to finish getting sent to the IR and try again.

7.2.7 106: ***xxxx IR com error***

The specified instrument rotator had a communications error. Turn on full verbose mode to see commands and responses.

7.2.8 107: ***xxxx IR com errors suspended***

The specified instrument rotator had too many com errors. Further error messages will be suppressed until communication resumes.

7.2.9 500: ***Warning! Instrument Rotator not homed***

Instrument rotator has not been homed. Home it with ***IR*** **HOME** command.

7.3 EDS log message (800-899)

7.3.1 800: Current encoder reading and hexadecimal motion values

Format: 800aaaaaabbbbbbccccccddddddeeff

aaaaaa: NW Axial reading (decimal)

bbbbbb: SW Axial reading (decimal)

cccccc: SE Axial reading (decimal)

dddddd: NE Axial reading (decimal)

ee: Encoder reading status (hex)

0..1: NW Axial reading status (0=OK, 1=Near Limit, 2=Error)

2..3: SW Axial reading status (0=OK, 1=Near Limit, 2=Error)

4..5: SE Axial reading status (0=OK, 1=Near Limit, 2=Error)

6..7: NE Axial reading status (0=OK, 1=Near Limit, 2=Error)

ff: Encoder motion status (hex)

0..1: NW Axial motion status (0=None, 1=Up, 2=Down)

2..3: SW Axial motion status (0=None, 1=Up, 2=Down)

4..5: SE Axial motion status (0=None, 1=Up, 2=Down)

6..7: NE Axial motion status (0=None, 1=Up, 2=Down)

7.3.2 801: Current encoder reading and hexadecimal values

Format: 801aaaaaabbbbbbccccccddddddeeff

aaaaaa: NW Radial reading (decimal)

bbbbbb: SW Radial reading (decimal)

cccccc: SE Radial reading (decimal)

dddddd: NE Radial reading (decimal)

ee: Encoder reading status (hex)

0..1: NW Radial reading status (0=OK, 1=Near Limit, 2=Error)

2..3: SW Radial reading status (0=OK, 1=Near Limit, 2=Error)

4..5: SE Radial reading status (0=OK, 1=Near Limit, 2=Error)

6..7: NE Radial reading status (0=OK, 1=Near Limit, 2=Error)

ff: Encoder motion status (hex)

0..1: NW Radial motion status (0=None, 1=Up, 2=Down)

2..3: SW Radial motion status (0=None, 1=Up, 2=Down)

4..5: SE Radial motion status (0=None, 1=Up, 2=Down)

6..7: NE Radial motion status (0=None, 1=Up, 2=Down)

7.3.3 802: Target encoder reading

Format: 802aaaaaabbbbbbccccccddddddee

aaaaaa: NW Axial target reading (decimal)

bbbbbb: SW Axial target reading (decimal)

cccccc: SE Axial target reading (decimal)

dddddd: NE Axial target reading (decimal)

ee: Encoder status (hex)

0..1: NW Axial target reading status (0=OK, 1=Near Limit, 2=Error)

2..3: SW Axial target reading status (0=OK, 1=Near Limit, 2=Error)

4..5: SE Axial target reading status (0=OK, 1=Near Limit, 2=Error)

6..7: NE Axial target reading status (0=OK, 1=Near Limit, 2=Error)

7.3.4 803: Target encoder reading

Format: 803aaaaaabbbbbbccccccddddddee

aaaaaa: NW Radial target reading (decimal)

bbbbbb: SW Radial target reading (decimal)

cccccc: SE Radial target reading (decimal)

dddddd: NE Radial target reading (decimal)

ee: Encoder status (hex)

0..1: NW Radial target reading status (0=OK, 1=Near Limit, 2=Error)

2..3: SW Radial target reading status (0=OK, 1=Near Limit, 2=Error)

4..5: SE Radial target reading status (0=OK, 1=Near Limit, 2=Error)

6..7: NE Radial target reading status (0=OK, 1=Near Limit,2=Error)

7.3.5 804: Current Galil switches

Format: 804aabbccddeeffgghh

aa: NW Axial status (hex)

bb: NW Radial status (hex)

cc: SW Axial status (hex)

dd: SW Radial status (hex)

ee: SE Axial status (hex)

ff: SE Radial status (hex)

gg: NE Axial status (hex)

hh: NE Radial status (hex)

For a-h,

2: 0 = At Up limit, 1 = Not at limit

3: 0 = At Down limit, 1 = Not at limit

7.3.6 805: Current axial DGH Load cell reading in mV and hexadecimal encoded alarm

Format: 805+aaaaa.aa+bbbbb.bb+ccccc.cc+ddddd.dde

+aaaaa.aa: NW Axial Load cell reading in mV

+bbbbb.bb: SW Axial Load cell reading in mV

+ccccc.cc: SE Axial Load cell reading in mV

+ddddd.dd: NE Axial Load cell reading in mV

e: Load cell alarm flags (beyond internal load cell alarm limit) (hex)

0: NW Axial alarm (0 = No alarm, 1 = Alarm)

1: SW Axial alarm (0 = No alarm, 1 = Alarm)

2: SE Axial alarm (0 = No alarm, 1 = Alarm)

3: NE Axial alarm (0 = No alarm, 1 = Alarm)

7.3.7 806: Current radial DGH Load cell reading in mV and hexadecimal encoded alarm

Format: 806+aaaaa.aa+bbbbb.bb+ccccc.cc+ddddd.dde

+aaaaa.aa: NW Radial Load cell reading in mV

+bbbbb.bb: SW Radial Load cell reading in mV

+ccccc.cc: SE Radial Load cell reading in mV

+ddddd.dd: NE Radial Load cell reading in mV

e: Load cell alarm flags (beyond internal load cell alarm limit) (hex)

0: NW Radial alarm (0 = No alarm, 1 = Alarm)

1: SW Radial alarm (0 = No alarm, 1 = Alarm)

2: SE Radial alarm (0 = No alarm, 1 = Alarm)

3: NE Radial alarm (0 = No alarm, 1 = Alarm)

7.3.8 807: Current axial tensions in pounds and status

Format: 807+aaaaa.aa+bbbbb.bb+ccccc.cc+ddddd.ddee

+aaaaa.aa: NW Axial Load cell reading in Pounds

+bbbbb.bb: SW Axial Load cell reading in Pounds

+ccccc.cc: SE Axial Load cell reading in Pounds

+ddddd.dd: NE Axial Load cell reading in Pounds ee: Tension Status (hex)

0..1: NW Axial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

2..3: SW Axial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

4..5: SE Axial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

6..7: NE Axial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

7.3.9 808: Current radial tensions in pounds and status

Format: 808+aaaaa.aa+bbbbb.bb+ccccc.cc+ddddd.ddee

+aaaaa.aa: NW Radial Load cell reading in Pounds

+bbbbb.bb: SW Radial Load cell reading in Pounds

+ccccc.cc: SE Radial Load cell reading in Pounds

+ddddd.dd: NE Radial Load cell reading in Pounds

ee: Tension Status (hex)

0..1: NW Radial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

2..3: SW Radial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

4..5: SE Radial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

6..7: NE Radial Tension status (0 = OK, 1 = Near Limit, 2 = Error)

7.3.10 809: Current Global coordinates

Format: 809aaaaaabbbbbbccccccddddddddeeeeeeeee

aaaaaa: current x coordinate in microns (decimal)

bbbbbb: current y coordinate in microns (decimal)

cccccc: current z coordinate in microns (decimal)

ddddddddd: current h coordinate in thousands of arc-seconds (decimal)

eeeeeeeee: current v coordinate in thousands of arc-seconds (decimal)

7.3.11 810: Target global coordinates

Format: 810aaaaaabbbbbbccccccddddddddeeeeeeeee

aaaaaa: target x coordinate in microns (decimal)

bbbbbb: target y coordinate in microns (decimal)

cccccc: target z coordinate in microns (decimal)

ddddddddd: target h coordinate in thousands of arc-seconds (decimal)

eeeeeeeee: target v coordinate in thousands of arc-seconds (decimal)

7.3.12 811: Fatal status and motion stage flags

Format: 811aaaabbcc

aaaa: Fatal flags (hex)

0: 1 = Error reading INI file

1: 1 = Homing error

2: 1 = Motors overheating

3: 1 = Emergency stop card triggered

4: 1 = Error reading tensions

5: 1 = No motor response

6: 1 = Error checking for fatal errors

7: 1 = Error reading encoders

8: 1 = Encoders beyond soft limits

9: 1 = Tensions beyond soft limits

bb: Status flags (hex)

0: 1 = Move in progress

1: 1 = Host communication OK

cc: Movement stage (hex) (0 = stopped, 1 = macro, 2 = micro, 3 = backlash)

7.3.13 82i: Instrument rotators status

Format: 82iabbbbbbbcccccccdddddeeeffff

i = The number of rotator: 0(NASW), 1(NASE), 2(CASS), 3(AUX1), 4(AUX2) and 5(AUX3).

a: 0(NASW), 1(NASE), 2(CASS), 3(AUX1), 4(AUX2) and 5(AUX3).

bbbbbbb: Requested position/velocity in deg * 1000 or deg/sec * 1000

ccccccc: Current position in deg * 1000

ddddd: Current velocity in deg/sec * 1000

eee: Current volatge in volts * 10

ffff: Status flags (hex)

0: 1 = Communication Enabled

1: 1 = Communication Error

2: 0 = Last command was a position command, 1 = velocity command

3: 1 = Servo error

4: 1 = Motor power enabled

5: 1 = CW motion

6: 1 = CCW motion

7: 1 = CW limit hit

8: 1 = CCW limit hit

9: 1 = Brake on

10: 1 = Lock pin in

11: 1 = Home in progress

12: 1 = Home completed/OK

13: 0 = CW Zone, 1 = CCW Zone

7.3.14 83i: Encoder values

Format: 83iabbbbbbbbbbbccccccccccdddddddddd

i = 0(NASW), 1(NASE), 2(CASS), 3(AUX1), 4(AUX2) and 5(AUX3).

a: 0(NASW), 1(NASE), 2(CASS), 3(AUX1), 4(AUX2) and 5(AUX3).

bbbbbbbbbbb: Encoder value

ccccccccccc: Servo error value

dddddddddd: Velocity in encoder counts/sec

7.3.15 84i: Servo flag for time and error tolerance

Format: 84iabc

i = 0(NASW), 1(NASE), 2(CASS), 3(AUX1), 4(AUX2) and 5(AUX3).

a: 0(NASW), 1(NASE), 2(CASS), 3(AUX1), 4(AUX2) and 5(AUX3).

b: Flag if internal servo has reached the desired tolerance (ErrorTolerance)

c: Flag that servo has been within certain tolerance (ErrorTolerance) for certain amount of time (TimeTolerance)

7.4 Successes (900-999)

7.4.1 983: *Host error messages resumed*

Communication with the host (TCS) was restored.

7.4.2 984: *xxxx IR com errors resumed*

Communication has resumed with the given instrument rotator.

7.4.3 985: *xxxx Instrument Rotator homed*

The given instrument rotator successfully completed a homing operation.

7.4.4 986: *UT set by TCS to xx xx xx.xx*

The vane end computer's clock has been set to the proper universal time by the TCS.

7.4.5 987: *Encoder read errors resumed*

Encoder read errors have been resumed because communication has been restored.

7.4.6 988: *Axial/Radial error messages resumed*

Either axial or radial Galil DMC error messages have been resumed, because communication with them has been restored.

7.4.7 989: *DGH x error messages resumed*

Error messages about DGH x have been resumed because communication with the DGH has been restored.

7.4.8 990: ***Screen dumped to SCREEN.BMP***

An picture of the screen was dumped to the file SCREEN.BMP in Windows BMP format.

7.4.9 993: ***Homing xx corner***

The given corner of the vane end system is being homed.

7.4.10 994: ***DGH init success***

All DGH modules were initialized successfully.

7.4.11 995: ***Main loop started***

All initialization completed successfully, and the main program loop is running.

7.4.12 996: ***Encoder homing success***

All 8 vane-end encoders were homed successfully.

7.4.13 997: ***VANE.INI read success***

The VANE.INI initialization file was read successfully.

7.4.14 998: ***DMC check success***

The Galil DMCs passed the program's startup check.

7.4.15 999: ***Program startup (vx.xx)***

First log entry after program startup (includes program version #)