CHARA Array Control Software Overview

1 Introduction

This document is intended as a guide for the control software for the CHARA Array. It is most certainly not complete and is probably not up to date either, but I have attempted to include a general description of how things work and, where I know about them, how things fail to work and what can be done to fix them. If you do not find what you need to know in here it is unlikely to be written up anywhere else and you will either have to ask someone who knows or work it out for yourself. If you find that you have come across some failure mode and recovery that is not in this document please feel free to add it, or at least tell me about it.

1.1 Real Time Modules, Servers and Clients

The CHARA control software is based on a three level approach. At the lowest level is the real-time (RT) code running in kernel space while the next two layers up are based on a "Client/Server" model and run under Linux proper. Each subsystem of the array is controlled either by an RT module, which is in turn controlled by a server program, or the subsystem is controlled directly by a server. The servers can then have multiple clients running on various machines that make requests of the hardware. If a client crashes for some reason you simply need to restart that client. If a server crashes things are more serious, but I will talk about that later.

1.1.1 Real Time Modules

Any subsystems that needs to do things at predictable times, or that needs to talk directly to hardware via the ISA or PCI bus, must do so using a real time module. This module, once compiled and installed, becomes part of the kernel itself and is almost always loaded when the machine boots up. If the module is not present the server that goes with it will complain about missing commands or FIFOs[1]. If this happens it's probably best to reboot the machine in question. Note that this reboot can have big effects as any other servers running on this machine will also be killed. If you are really worried about which modules are loaded and don't wish to reboot you can use the command `/sbin/lsmod` to get a list of the currently loaded modules. The file `/etc/rc.d/rc.local`, also sometimes accessible via `/etc/rc.local`, will contain the commands used to load the various modules required on any given machine. Note that you need to be logged in as super user to do change modules.

Sometimes these real-time modules can cause crashes all on their own. For example, it is known that the `irimage` computer can go legs up because of a conflict with the serial port. When this happens there is nothing you can do but force the machine to reboot, either by using the power GUI to cycle the power on that machine or walking over to the machine and hitting the reset button. In fact, in most cases it is actually better to turn the machine completely off and then back on again as this forces all attached hardware to reset. Before you do this you should turn off any attached hardware. For example you should turn off the tip/tilt high voltage and shutter power before you reboot the `ople` machine as otherwise all the shutters will open for a while during reboot. You should also turn off the CCD power before you reboot `tiptilt`. Look at the appropriate section on each computer to see what exactly is attached before you reboot anything.

---

[1] First-In-First-Out queues, the way servers and real-time modules communicate

In general, you should not have to worry too much about the real-time modules, you just need be aware that they exist, do most of the real control work, and can sometimes cause a serious crash, that is, one that will stop the entire machine, not just a single process. Would that all software was 100% stable.

1.1.2 Servers

Each device or subsystem is controlled by a single program called a server. A server is a program that accepts requests, performs a task and in many cases returns some data. There are two types of servers in the CHARA Array control system. One kind runs as a daemon and is normally connected directly to a piece of hardware via a serial port. The second kind has a text based engineering interface and normally connects to a real time module through a series of FIFOs.

While there can be multiple real-time modules loaded on any machine, and this is almost always the case, there will only ever be a single server connected via the FIFOs to these modules and from the outside world this will appear as a single entity. Similarly there can only be one server connected to any serial port.

When logged in as the user `observe` you should find a menu listing for `Servers` when you click on the background screen. This menu has a list of servers that need to be started manually. The other servers are generally started automatically at boot time from the file

> `/etc/rc.d/rc.local`

If you really get in a fix you can look in this file on any given machine and find the commands needed to start a server. This is to be avoided if possible. In truly desperate circumstances you may need to reboot the machine in question. The main advantage to starting a server directly from a command line is that you will be able to see any error messages it produces. This is sometimes handy if the socket manager (see Chapter 3 [Socket Manager], page 5) doesn't seem to be able to restart it.

Those servers with a text based engineering interface are the most complex and do the most work. There are many commands associated with engineering tests that are only available from the server command line and will not will be available from a GTK[2] client. While these servers can be run from anywhere they are most commonly run from `zoot`[3], `ctrscrut`[4] or `wazoo`[5]. These servers are complex and attempt to start in a state appropriate for most observing situations. However, there are some connections not made automatically on start up as they can be dangerous. Instead of doing everything at start-up, each of these servers has a list of commands called the `autolist` which you may need to run manually on some occasions. If you are not an experienced operator do not use the `autolist`, it can do very dangerous things, for example put the telescopes out of focus or even screw up the secondary alignment. In the discussions below for each server I will try and mention in the various sections on each server when you may need to run the `autolist`. At the time of the latest update of this document the `autolist` has been made largely obsolete. Only the GPS server has an autolist you might need to use.

---

[2] Gimp Tool Kit, a library of routines for developing X-windows interfaces

[3] From "Zoot Allures" a favorite Frank Zappa album of mine

[4] Another FZ reference, "Central Scrutinizer" from Joe's Garage. Note that this machine is also known as `michelson` from the outside world.

[5] Yes, yet another Zappa reference to the "Grand Wazoo".

It is also possible to connect to most servers with an engineering interface from any text terminal with the command

```
speak server_name
```

which will start a `telnet` like session with the server. Almost all commands will be available during a `speak` session, though some commands are reserved use only directly from the server console. Two useful commands are `commands` which will list all the known commands and `status` which will show you the current status of that server. In almost all cases while observing you will not have to use the command line interface.

Sometimes server displays get contaminated by text from other programs. You can fix this by typing `CTRL-L` in the server window.

The system is always changing, and we on occasion move a server from one machine to another, so this document will date relatively quickly. At the time of writing this addition, all serial port based servers that do not talk to hardware at the telescopes run on `ctrscrut`.

1.1.3 Clients

A client is a user interface that connects to one, or several, servers and allows the observer to make requests for hardware changes or data. There can be many clients running for a given server and clients are not restricted to talking to a single server. Most clients are GTK based, where GTK stands for Gimp Tool Kit, a software library for developing X windows programs, although some are text based, and the FLUOR system, which can be looked upon as one big client for the entire array, uses LabView.

There is a menu listing on the X windows screen called for all the GTK based clients. During normal observation it should not be necessary to use any clients not listed in this menu.

Under normal circumstances all clients for a given server will automatically terminate, or issue a warning, when a server stops running. It is possible, however, for servers to crash leaving clients still running. When this happens the client still thinks it has a valid connection while in fact it is talking to nothing at all. The most common symptom of this is that nothing happens when you click on buttons. All clients have a PING button which should make the server respond with a text message of the form

```
irimage on buckhead.chara-array.org is alive
```

On some servers this message may take a few seconds to appear so wait for a while before deciding things are broken. If you still don't see a return message it could mean the server has died, although it could also mean it is busy doing other things, but this is very rare. If the server really is dead, you will need to restart it. Once the server has been restarted you may have to restart the client also, although most clients do have `REOPEN` buttons that should have the same effect. So, if you ping a server and there is no response, first try the `REOPEN` button on the client if there is one and then see if it responds to a `PING`. If it doesn't you will have to go and see what can be done using the socket manager (see Chapter 3 [Socket Manager], page 5) program.

2 Logging In

Before you can do anything on any machine you need to be able to log into that machine and start X windows. The account to use is

     `observe`

and you can get the current password from someone on the mountain. Once logged in, start X windows with the command

     `startx`

which should, eventually, present you with a blue background screen and start a number of the most commonly used GTK clients.

We normally leave the observe account logged in at all the telescopes and on the machines `ctrscrut` and `zoot`, but sometimes, like after a machine has been re-booted, you may have to log in yourself. Never do anything with the CHARA system as any user other than observe. A lot of things will work, but some important things will not.

3 Socket Manager

There is one client/server combination that is different to all the others in the CHARA control system in that it does not control any hardware. Instead it keeps track of which servers are running, which machines they are running on, and which sockets they are listening to. The server is called `socket_manager` and it should always be running on the machine `ctrscrut` and it should always be listening to socket 2200. If the client fails to speak with the server it will automatically try and restart it. If the socket manager is not running nothing else will work correctly. If you have to, you can start the socket manager manually by typing, on `ctrscrut`, the command

        /usr/local/bin/socket_manager &

You will then get a number of messages from the server, or an error message if there is a problem. As pointed out in the last section it is very important that you do not start the socket manager under any account other than `observe`.

The GTK client for the socket manager is called `sockman` and should start automatically when you start X windows. It is also available from the `Observing` menu when you click on the blue background screen. Figure 1 shows a screen shot of the socket manager client.

Figure 1. This is what the socket manager client looks like.

There is also a text based client called `tsockman` for the people who like to type things at a command line. This is normally used in scripts and there is rarely any need to use it by hand. Still, for the command line lovers amongst us it is there to use should you wish to.

When a server starts running the first thing it will do is ask the socket manager if a server by this name already exists. If it does the server will refuse to run again. If no server of that name exists it will connect to various sockets and then register itself with the socket manager. The seven pieces of information registered with the socket manager are:

Data Item	*Description*
Name	The server name, which must be unique.
Machine	The name of the machine on which the server is running.
PID	The process identification number of the server.
Commands	The command socket of the server. A command socket is a text based socket interface rather like telnet for the server and can be reached using the command `speak`.
Data	The data socket of the server. A data socket is often used to pass information between two servers. For example the `irimage` server uses the data socket to pass fringe location information to the `ople` server when scanning for fringes.
Messages	The message socket of the server. This is the socket used by the clients to communicate with the server.
Restart	This field displays the text of the command you may need to restart the server.

Not all server entries will have data in all fields, though they should all have a name, machine and PID. The buttons/commands on the clients are:

Data Item	*Description*
`GET`	Retrieves the data for a server. Type the server name in the name entry area and either type the return key or click the `GET` button.
`SET`	Adds an entry to the data base. Type the various pieces of information in and click the `SET` button. Since servers automatically add their information to the data base this button is only ever used for debugging.
`LIST`	Produces a click-able list of know servers in the data base.
`REMOVE`	Removes an entry from the data base. You may have to do this sometimes if a server has died and not removed itself from the socket manager data base.
`STOP`	Clicking this button will send out a `kill -2` command to the server. This should stop the server gracefully, but it may not.
`KILL`	This issues a `kill -9` signal to the server and will stop it, though it will not stop gracefully and will not remove itself from the socket manager data base. Be very careful you are sure of what you are doing before you push this button!

PING	Sends a ping message to the server and waits for a response. The client will then tell you if it got a response or if the server appears to be dead.
RESTART	Try and issue the restart command if there is one. This button can be used to restart a server that has died.
KILL SERVER	Try and issue the restart command if there is one. This button can be used to restart a server that has died.
QUIT	Quit the socket manager client.

If you suspect that a server has died you should go to the socket manager client and see if the process is still listed in the data base. If it is not listed you will have to restart the server manually. Please refer to the relevant section below for that server, or at a pinch, go have a look at the command in the `rc.local` file for the computer in question. If it is still listed go ahead and `GET` its information and try and `PING` it. If it responds to the ping it means the server is alive, if not well. If the socket manager reports that it is dead you should try and `REMOVE` the entry from the data base and then `RESTART` it. Note that you should not hit the `GET` button again as you will loose all the information you will need for the restart. You should try and ping it again to check that all went well.

Sometimes this process will not work. One common cause is that the server was not started by the account observe and so it can not be restarted by that account. If you started the socket manager client from a command line you should see messages appear from the socket manager server when you try and restart another server. There will be error messages amongst them if there is a problem. You could try and issue the restart command manually on a command line but make sure you are logged into the correct machine before doing this.

Another common problem for servers that connect to a serial port is that it will complain that it could not open a serial port because it was locked. This is the result of a stale lock file on the machine left from the previous, assumed crashed, incarnation of the server. You need to manually delete the lock file. Lock file names are always of the form:

    /var/lock/LCK..ttyXX

where `XX` denotes the particular port. If a server complains that a serial port is locked, make sure there is no other version of the server running and then remove the lock file manually. You may have to become root to do this. Normally the restart command will remove any stale lock files for you.

For the servers that have a engineering interface any serial ports used will be opened either at startup or when you run the `autolist`. If there is a stale lock file it will prompt you to see if it should be removed.

If a server has crashed and you have been forced to resort to using the socket manager to do a restart it is very likely that there will be many clients out there that still think they are connected to the old server. Most clients have `REOPEN` buttons which should re-connect to the server. Those that do not have this button will need to be restarted. Furthermore, some servers are themselves clients of other servers. For example many servers try and talk to the shutter server so they can open and close shutters. These connections will also have to be re-opened before you continue observing. I will try and mention these interconnections when I talk about each server below.

Finally, it is possible for the socket manager to lock up. Actually, it is likely to happen to you at some stage during an observing run. At the time of the most recent update of this document this fault had been traced to times when the socket manager is trying to service too many requests at once. Unfortunately, the problem has not yet been resolved. The most obvious symptom of this is that when you ask `sockman` or `tsockman` for a list of servers it doesn't reposed or responds with an error. Try hitting the `KILL SERVER` button, wait for a while and then ask for a list. If this doesn't fix the problem you will need to connect to `ctrscrut` and do things manually. Use a `ps` command to find the process ID of the `socket_manager` program and manually kill it. For example you could use the command:

```
ps -aux --cols 200 | grep socket_manager
```

to get the PID and then

```
kill -9 ###
```

where `###` is the process ID for the program. Once you have done all this, make sure that the `sockman` program can now talk to the server.

Finally, sometimes the socket that the socket manager uses (2200) remains locked and it will refuse to start. The socket manager is the only program that must use the same socket number every time it runs, and if the socket is locked it will not start. There is little you can do about this except wait, having first made sure there are no versions of the program still running. Normally the socket will free itself up in a few minutes. If it still doesn't clear, you will have to reboot `ctrscrut`, but please don't do this lightly, it is the file server for the entire array.

4 GPS Computer

The time standard for the all the computers in the OPLE building is derived from a small GPS receiver attached to a card inside the GPS machine. The telescope servers no longer use this time base as they each have their own GPS receiver called the `Bunker Clock`. The antenna for the GPS in the lab is above the beam combining lab. A local oscillator provides 16 MHz, 1 kHz and 1 Hz signals that are distributed around the electronics rack. All of the computers in the system use these signals to run a local clock. The whole thing can be synchronized to the GPS clock using the command `gsync` on the server or clicking on the `GSYNC` button on the client. Note that this only gets the clock on the `gps` machine synchronized and this then needs to be distributed throughout the array. Individual machines can be synchronized using the `msync` command and button. Normally all this is done by the `grand_wazoo`[1] at the beginning of an observing session.

Figure 2. Screen shot of the GPS server

If for some reason you need to switch off the `gps` machine, or simply reboot it, it can take quite a while for the unit to find the three satellite signals it needs to properly run the clock. This can can 10 to 15 minutes, and there is nothing you can do but wait. This is why `gps` is a machine by itself. When you start up the GPS server if the clock is running there is no need to use the `autolist`. If you have recently re-booted the machine, or for some other reason the clock is not running use the `autolist` to get things going. Note that if the GPS unit has not found all the satellite signals yet this will not work. It's best to

---

[1] Right, FZ again.

wait until you see that the GPS clock field in the status window is showing a time and that time is changing.

There has long been an unexplained 2 or 3 second difference between the time that the GPS says it is and the UT as presented on several web pages, like the USNO master clock. We still do not understand this, but we suspect it has something to do with leap seconds. The default offset of 3 seconds is built into the `gsync` command. Normally the `gps` clock is set to the time on one of the bunker clocks and this issue with leap seconds is not important. What is important is that the one second ticks are happening at the right time. Always check that the time shown on the GPS server matches the USNO clock on a web page or one of the bunker clocks on one of the telescopes. The command `time` on any server with a clock will display the time of the local CHARA clock, as well as the difference in milliseconds between this clock and the local Linux clock. Most Linux clocks are synchronized using the `NTP` daemon. If there is a large difference you can force the Linux clock to agree with the CHARA clock using the `sync` command.

The `gps` server has a servo built in that forces the Linux clock on this machine to be synchronized to the GPS time. This means that this computer can be used as an `NTP` server for machines on the local network.

Note that the GPS and CHARA times will slowly drift with respect to one another. The CHARA clock uses a local oscillator that is not perfect and it can loose several seconds per day. The `gsync` command will fix this. While it will mean small problems in delay line positioning a drift like this is not dangerous. Having many clocks in the system disagree on the time is far more serious than a few, but constant, second offset.

A final thing of note concerning time and the array concerns the small corrections applied by astromod for polar motion of the earth. These are issued by the IERS service, and while they do not change very rapidly it is a good idea to to check on the current values and enter them into the file `/ctrscrut/chara/etc/system.chara`. The parameters of interest are `POLARX`, `POLARY`, and `DUT`. The latest values can be found on the web page `http://maia.usno.navy.mil`. If you wish, you can sign up and have them email the predictions directly to you.

Figure 3. Screen shot of the GPS GTK client

5 Telescope Computers

There are two computers in each telescope bunker, one that runs the telescope control software (TCS) and another running RT-Linux that controls almost everything else. The DOS based machines running TCS are not on the network, but the Linux machines are and are called S1, S2, E1, E2, W1 and W2. Each telescope computer runs four servers, although only three of these are controlled by the socket manager.

5.1 Telescope Control Computer

The TCS computer is a plain DOS machine[1] and it runs a commercial program that controls the telescope and the dome. Controlling the telescope directly from TCS is quite intuitive, but is beyond the scope of this document. Come on out to Mount Wilson and one of us will be happy to show you how.

Figure 4. Screen shot of the TCS program

Since the telescope encoders are relative, and not absolute, the TCS program is left running all the time in case someone moves a scope by hand[2]. If this does happen, at least the TCS program will know where the telescope is. If it does lose track of the telescope position someone has to manually find a star[3] and initialize TCS. If for some reason you need to stop TCS or reboot the TCS machine it is very important to first make sure that the telescope and dome are properly stowed and that the power for moving the dome slit and the telescope itself is switched off. This is because when TCS is first started it assumes the telescope and dome are in the stow positions and that you want the slit closed. If the slit is open and the power is on it will close as soon as you start TCS, which can be dangerous if the primary mirror cover is open. If you do this by mistake quickly tell TCS to open

---

[1] Yes DOS! We used a DOS window under windoze for a while but it was not stable. Plain old DOS can run for months without a reboot

[2] Or bumping into it.... it has happened.

[3] Yes we do have telrads.

the slit or cut the power to the slit. We'd hate to see things falling off the dome and onto the primary mirror! Note that the `STOW` command in the telescope GUI forces the dome to point north. This is not the stow position. Always stow the dome using the TCS software before rebooting the TCS computer.

5.1.1 Things that can go wrong with TCS

Here is an incomplete list of things that are known to go wrong with TCS and what you might try to do to fix them.

5.1.1.1 TCS hangs when rebooting

For reasons unknown sometimes the TCS computer will stop when rebooting. In most cases you can get around this by disconnecting the D25 plug that connects the paddle to the DOS machine, reboot and then reconnect the D25. If this fails you have a more serious problem.

5.1.1.2 The dome doesn't move with the telescope

Is `autodome` switched on? This can be done from the TCS program direct or from the `CONTROL` menu on the GTK client or from the telescope server using the command

```
autodome on
```

5.1.1.3 The dome moves to the wrong place

Sometimes TCS can get very confused about the dome position, which is also relative and not absolute. However in this case there is a fiducial and if you ask TCS to send the dome to home, either directly through TCS or via the Linux based telescope server using the command `HOMEDOME` it will rotate around until it finds the stow position. If you then turn on the `autodome` function all should be well. If the dome still goes to the wrong position it may mean TCS has lost track of which direction it should move the dome, this happens rarely but it does happen. The most common fault is for the dome to be exactly 180 degrees away from where it should be. In this case you need to restart the TCS software, being careful to stow the dome and telescope first and turn off the slit power.

5.1.1.4 The telescope doesn't move

Have you turned on the power to the telescope drives? See the section on the RPC controller below. Have the drives been disabled? If they have, something serious is going on. Drives are disabled by hitting the `F10` key on the TCS computer or via the telescope server with the commands `kill` or `disable`. You can see that it has been disabled as you will see text on the TCS screen flashing saying that this has happened. If you are sure everything is alright you can re-enable the drives by hitting the `F10` key again on the TCS computer or using the commands `unkill` or `enable` on the telescope server.

5.1.1.5 The telescope doesn't point at the star

Is the time and date on TCS set correctly? Often, if you have restarted TCS or re-booted that TCS machine the clock will be wrong. TCS uses the normal PC clock for its time base and loses a few seconds per day. Under normal observing conditions the `grand_wazoo` will synchronize all the clocks in the system. If you wish to synchronize the TCS clock with the telescope server this can be done in the `CONTROL` window of the telescope GTK client using

the buttons `SET TIME` and `SET DATE` or at the command prompt with the commands `tdate` and `tsync`. Of course, this assumes that the telescope server clock is correct.

If the time and date are correct and the telescope is still not pointing at the star it means that it has lost its position. You need to use the paddle to manually find the star and then re-initialize TCS. This can be done directly at the TCS computer, with the command `init` at the telescope server or using the `INIT` button in the `CONTROL` window of the GTK client. Of course you should never initialize a telescope if it is not pointing at a star.

If the telescope still doesn't want to find stars the pointing model is broken in some way. You need to build a new pointing model, but that is beyond the scope of this document.

5.1.1.6 When I re-initialize TCS the star moves

This problem is caused by a low level offset in the elevation and azimuth control of TCS. When tracking a star, the telescope server is continually checking to see if the tip/tilt mirror is safely within its range. If it gets too far away from the center of its range it commands the telescope to move a little in either azimuth or elevation to re-center the tip/tilt mirror. You can see this happening on the `wobbler` or `tiptilt` displays when the green dots that represent the tip/tilt mirror go close to the edge and suddenly jump back towards the center. These changes in elevation and azimuth are done at a very low level within the TCS software, and the higher levels of the program know nothing about it. So, if these offsets are not zero when you initialize the telescope things can go wrong. The solution is to make sure these offsets are zero, and the only way to do that is to use the command `xoffzero` on the telescope server. There is no equivalent GTK client command as under almost all normal conditions this zeroing is done automatically.

5.1.1.7 The paddle moves the star too slow/fast

The paddle can either be controlled manually or by the telescope server and the speed at which it moves can be changed. This speed can not be changed at the TCS console[4] but it can be changed from the telescope server using the command `guide XX` where `XX` is a number of arcseconds per second. It can also be changed when using the `PADDLE` window, under the `CONTROL` window on the GTK client. Note that under the GTK client the new speed will not take affect until you hit one of the paddle buttons.

5.2 The Enclosure Server

The old enclosure server has been replaced with the `Cylinder` server, and it is fully complient with the `socket_manager`. Normally you will only nee d to click on the `OBSERVE` or `CLOSE` buttons. An engieerning interfa ce also exists to allow more general commands. You can begin the client with the command

        cylindergtk &

and adding the switch `-e` to it will give you the engineering version. It is also available from the normal menus.

---

[4] No, I don't know why not.... ask the guy that wrote TCS

If you need to restart the cyclinder servers on a scope you will need to look into the `/etc/rc.local` file of that machine to see the command.

CLOSED	CLOSED	CLOSED	CLOSED	CLOSED	NO SERVER
STOP	STOP	STOP	STOP	STOP	STOP
CLOSE	CLOSE	CLOSE	CLOSE	CLOSE	CLOSE
OBSERVE	OBSERVE	OBSERVE	OBSERVE	OBSERVE	OBSERVE
OPEN	OPEN	OPEN	OPEN	OPEN	OPEN
S1	S2	E1	E2	W1	W2
OPEN	OPEN	OPEN	OPEN	OPEN	OPEN
OBSERVE	OBSERVE	OBSERVE	OBSERVE	OBSERVE	OBSERVE
CLOSE	CLOSE	CLOSE	CLOSE	CLOSE	CLOSE
STOP	STOP	STOP	STOP	STOP	STOP
CLOSED	CLOSED	NO SERVER	CLOSED	1.37	NO SERVER

ALL CLOSE | PING | REOPEN | QUIT

Figure 5. This is what the cylinder client looks like.

5.3 The Remote Power Controller (RPC)

In each telescope bunker there is a device called the remote power controller or RPC that allows us to turn the power to various things, like the telescope drives, on and off. This connects to serial port `/dev/ttyC1` on the telescope computer and is controlled by the program `rpc_server`. The manual command for starting this server as it appears in the `rc.local` file is

```
/bin/rm -f /var/lock/LCK..ttyC1
/usr/local/bin/rpc_server /dev/ttyC1 /ctrscrut/chara/etc/rpc.cfg >& /var/log/rpc.log &
```

although redirecting the output to `/var/log/rpc.log` is not really necessary and can only be done as root. It is extremely rare for this server to crash and you can almost always restart it using the socket manager. The file `/ctrscrut/chara/etc/rpc.cfg` is a text file and contains the configuration information for this device, basically a name which will be something like `RPC_S1` and a list of the devices connected to the RPC unit.

The client for the RPC unit is available from the `Observing` menu and has the command line form

```
rpcgtk RPC_S1
```

You cal also start the standard power GUI from the X windows menu which will let you control all of the RPC units in the entire system.
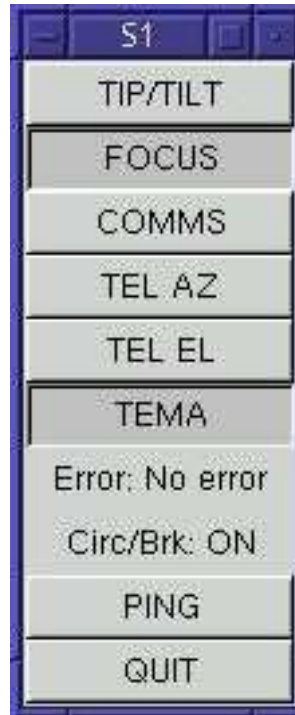
Figure 6. This is what the RPC client looks like.

Normally we always leave the FOCUS unit off. This is the PI844 box in the rack in the bunkers that controls the motorized micrometers on the telescope secondary. We rarely need to move these and we have found that on some units turning the power off and on can make the micrometers jump in random ways[5] and this moves the secondary out of alignment. This is a very bad thing. Unless you really know what you're doing don't turn these on.

All telescopes now have TEMA units[6]. We rarely turn these off as they monitor all sorts of handy things within the dome.

The other selections on the telescope RPC units are TIP/TILT, the high voltage amplifier for the secondary PZTs, COMMS, the power for the tip/tilt communications, and the power for the two telescope axes. These are only ever turned on when actively observing and it is not recommended to turn them on without having first seen the telescope and made sure it is in fact in the position the software thinks it is in.

5.4 Weather Server

Each telescope has its own weather monitoring system. You can start the weather server from the X windows menu or by typing the command xweather. Some of these servers are

---
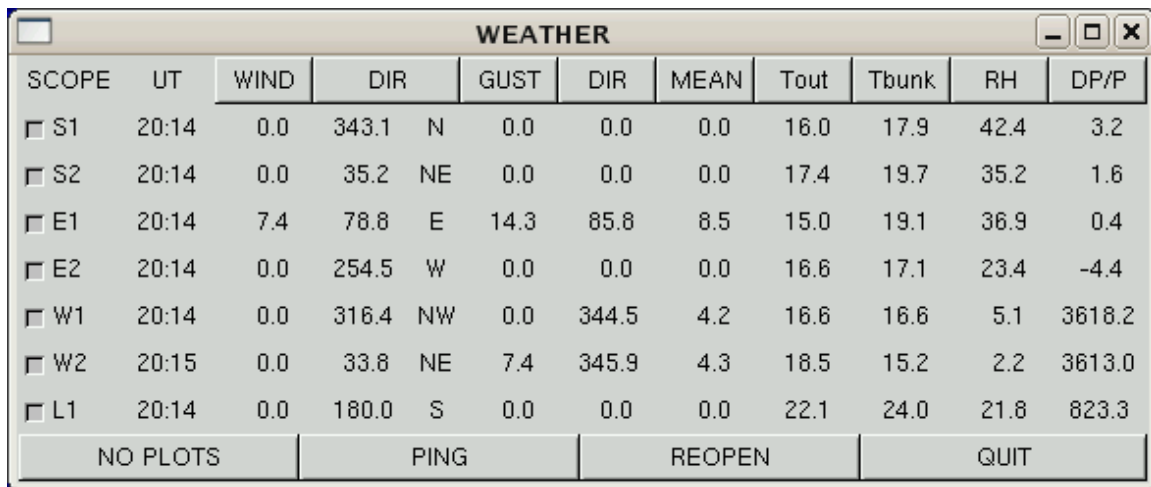
[5] Thank you very much PI!

[6] TElescope MAnagers, those little boxes attached to the telescopes that allow you to control things from within the domes

exposed to a great deal of interference from the TV transmitters on the mountain, especially those on the western arm. Consequently they give false humidity readings. We have found that the E1 station is the most reliable for relative humidity measurement.

The last line of the weather client labeled `L1` is a device setup inside the laboratory itself. The `Tout` represents the temperature inside the optics laboratory, while the `Tbunk` is the temperature in the electronics area. The `DP/P` entry for this unit is the atmospheric pressure, and not the dew point.

WEATHER										
SCOPE	UT	WIND	DIR		GUST	DIR	MEAN	Tout	Tbunk	RH	DP/P
S1	20:14	0.0	343.1	N	0.0	0.0	0.0	16.0	17.9	42.4	3.2
S2	20:14	0.0	35.2	NE	0.0	0.0	0.0	17.4	19.7	35.2	1.6
E1	20:14	7.4	78.8	E	14.3	85.8	8.5	15.0	19.1	36.9	0.4
E2	20:14	0.0	254.5	W	0.0	0.0	0.0	16.6	17.1	23.4	-4.4
W1	20:14	0.0	316.4	NW	0.0	344.5	4.2	16.6	16.6	5.1	3618.2
W2	20:15	0.0	33.8	NE	7.4	345.9	4.3	18.5	15.2	2.2	3613.0
L1	20:14	0.0	180.0	S	0.0	0.0	0.0	22.1	24.0	21.8	823.3
NO PLOTS			PING			REOPEN			QUIT		

Figure 7. This is what the weather client looks like.

5.5 Telescope Server

The fourth, and most complex, server running on the telescope computers is the one that controls the telescopes themselves. This server is a text based program but it also registers itself with the socket manager (see Chapter 3 [Socket Manager], page 5) and will listen for commands from both GTK based clients and the text based clients `speak` and `com`. This program has serial port connections to several devices: the TCS computer, the PI844 focus box, and the TEMA unit. It also receives the tip/tilt data from the `wobbler` (see Chapter 6 [Wobble Computer], page 24), and `tiptilt` (see Chapter 7 [Tiptilt Computer], page 29), sends the correct signals to the secondary PZTs and has a digitizer for video signals, such as the acquisition, finder and spy scope signals. All in all, this server is surprisingly stable considering the amount of things it needs to do.

A telescope server can be started from the command line with a command like `xe1` or it can be started from the `Servers` menu on the X windows screen. Note that only one copy of any telescope server can run at any time. Most often we run this, and all other text based servers of this kind, on `zoot`, but it is not uncommon to have it running on a local telescope Linux box in one of the bunkers when debugging. The socket manager can be

used to see if and where any given server is running, though if you try and start a telescope
server, and it is already running somewhere, it will soon let you know.

Figure 8. Screen shot of the telescope server for E1

It is not recommended that observers ever try and play with telescope focus. If you do
need to talk to the PI844 unit, and this could only apply to one of the more experienced
people on the mountain, you will need to turn on the power and open the serial port con-
nection to the PI844 unit using the command opi and wait for it to complete, which will
include a few long pauses while the electronics settle. When you have finished working on
focus, or telescope alignment, close the port using cpi. One last warning for the inexperi-
enced, never play with the focus of the telescope unless you are really sure of what you are
doing, you can cause serious damage and I will be forced to hurt you.

Once a telescope server is up and running, it is most commonly operated via its associated
GTK based interface, available from the Observing menu under X windows or by command
line using

        telescopegtk S1

It is also accessed from the `grand_wazoo` the main sequencer for the array.
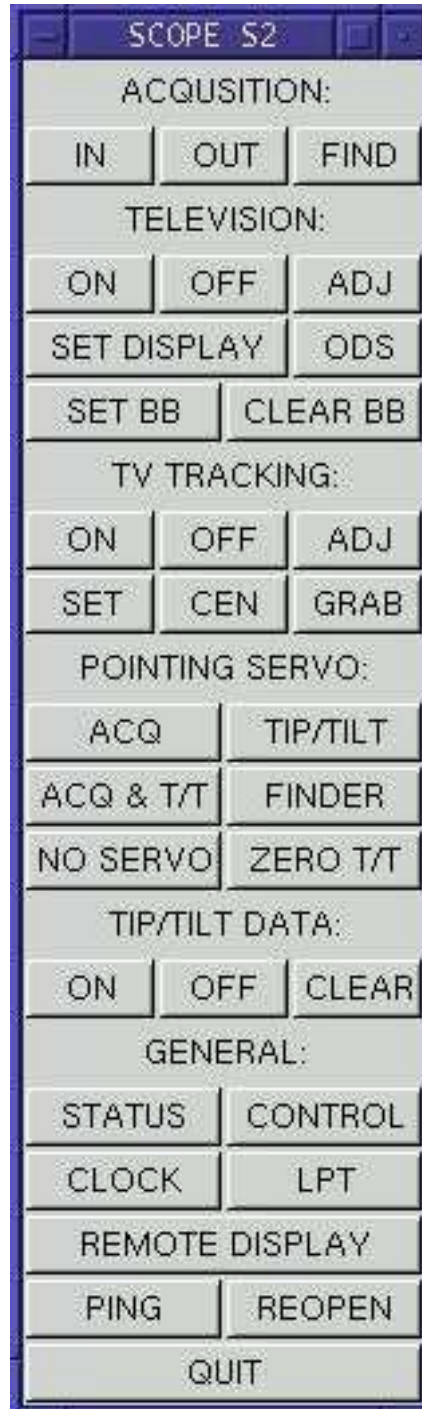
Figure 9. Screen shot of the telescope GTK client

5.5.1 TEMA

Each telescope has a small computer control box and paddle inside the dome called TEMA. This unit controls the dust covers, video signals and so on within the dome. They can also, via the telescope server, move the telescope, the dome and the appropriate M10 mirror for alignment. To use the TEMA from within the dome you need to put it into local mode and this is done by pressing the * key on the key pad, followed by the password and then the enter key. You leave local mode by hitting the * key again. You will need to ask someone on the mountain for the password.

Once you are in local mode there are codes for each of the functions that TEMA can perform. A list if these codes should be attached to the TEMA unit itself. Note that many of the functions of TEMA will not work unless the telescope server is running and listening to the TEMA commands. Once you are finished with the TEMA, be sure and go back into remote mode as if you leave it in local mode it will block any commands from the outside world. If you do leave it in local mode, you can force it back into remote mode from the telescope server by hitting the `ESC` key.

5.5.2 Things that can go wrong with telescope servers

Here is an incomplete list of things that are known to go wrong with telescope servers and what you might try to do to fix them.

5.5.2.1 Server complains about TCS communications

The link between the TCS computer and the telescope server is via a serial port, and if this is not open no communications can take place. You can try and reopen this connection by typing the command `otcs` on the command line of the server. If this doesn't work it could be that the TCS computer has closed its side of the connection. You will need to walk out to the telescope bunker and try to restart the serial port communication by typing `ALT-1` (the alt key and the 1 key at the same time). This toggles the communications and so you may have to do it a few times. Since we moved away from using `windoze` this is rarely necessary.

5.5.2.2 TEMA doesn't seem to talk to the server

One main symptom of this is that the M10 motion does not work, or the telescope does not move when instructed to by TEMA. Like TCS, TEMA uses a serial port connection to communicate with the server and this may have been closed. Try re-opening it with the command `otema` on the telescope server. If things still do not work try closing the port with `ctema` and then re-open it again. Next try turning the TEMA off and back on again using the RPC client for that scope. If this still fails, and though I have seen this happen and have no idea why, you need to restart the telescope server itself. This problem happens most often with W1 and almost never on S1 and S2.

5.5.2.3 TEMA still cannot move M10

The telescope server moves M10 by opening a socket connection with the relevant picomotor server. If this connection has failed it must be re-opened, which is automatically done each time the TEMA systems goes into local mode. So if you go into remote mode and back into local mode it should fix the problem. Failing that, the command `pico` will do the same

thing, but make sure TEMA is not in local mode when you do this. Finally, it could be that the picomotor server has crashed. This can be checked using the socket manager.

Note that since we installed the cameras that allow you to see the fiducial in the dome it is no longer necessary to move M10 from within the dome. Feel free to rest in the comfort of the control room and use one of the picomotor servers.

5.5.3 The TV picture does not update

The update of the TV images are controlled by the clock on the telescope server. Sometimes if you update that clock, or synchronize it to GPS the TV images will stop. Turn off the TV and then turn it back on again.

5.5.4 The TV picture does not show up when I click the button

Use the SET DISPLAY button on the GTK client to find out on which machine the system thinks it should display the TV images. If this is set correctly the TV may have been started from a different client somewhere. Turn off the the TV and turn it on again. As of early 2008, the displays themselves are generated by the GUI's and not the server. It should be possible to have displays running on multiple clients. If you do not get a display on your client, you might be running an outdated version of the client. If you really need to run the displays from the server, you need to type the command tvon in the server itself.

5.5.5 The acquisition/finder cross hairs are doing strange things.

The main symptoms of this are the cross hairs not moving in the way you expect, always being near one edge or another. This is a bug that still confuses me, but it is related to the system losing track of how many pixels are in a TV image. The only things you can do when this happens is to stop the telescope server and remove the acquisition position file with the command

        rm /ctrscrut/chara/etc/E1_acq_origin.dat

where you must replace E1 with the name of the telescope in question. Once this has been done restart the telescope server. When you next look at the acquisition TV the server will complain and force the origin to zero. You must then readjust the cross hairs to match the laser position.

A similar fault can occur with the finder scope positions. In this case you need to remove the file /ctrscrut/chara/etc/W1_find_origin.dat instead. Once this has been done and you have restarted the telescope server should center a star in acquisition, go to the finder image, turn on TV tracking and set the origin for the finder.

5.5.6 The Tip/Tilt mirror is not moving

This is most often because the tip/tilt communications between the wobbler or tiptilt machines and the telescopes has not been turned on. In the status window for the telescope the signals should be on and the receive rate should be 1000Hz, if you are using wobble to close the loop or the same as the frame rate of the CCD if you are using tiptilt to close the loop. Try clicking the WOBCOM button on the grand_wazoo and see if things improve. You should also make sure that the power for communications for these signals are turned on (These are the COMMS button on the RPC control for that telescope and on the lab power RPC unit). and that the wobble and tiptilt programs know which telescope is in which beam. If it is wrong you can fix that using the scopes command.

5.5.7 The acquisition laser position keeps changing

This is a bad thing as it means something is out of alignment. There are three common reasons for this to happen. The first is that the ball bearings that create the kinematic points for the acquisition mirror have got dirt or rust on them. To fix this you need to walk out to the telescope in question, open the acquisition box and clean the ball bearings. This should normally only be done by an experienced observer. The second reason for the laser spot to move around lot is that the Coudé alignment of the telescope is no good. There's not much you can do about this except report it and hope someone (usually Judit) fixes it in the next few nights. The third reason is that the M10 alignment has drifted and needs to be redone, and this is much more likely than a Coudé problem.

5.5.7.1 The telescope is out of focus

First thing to do if you think things are out of focus is to not touch anything for a while and convince yourself that it is indeed the telescope that is out of focus. Maybe the acquisition camera is out of focus, or the reference camera. Changing the telescope focus is not often necessary, that is once or twice a year, and it is not taken on lightly. That said, here's a brief outline of what to do.

During the internal alignment you should have seen the laser and/or white light spots on the reference camera. This is a good time to ensure that it is correctly focused on the laser. Of course, it is highly likely that if you have found a problem with focus on the sky you will have forgotten to check this. Still, we have not had to change the laser or the reference camera focus in many a year.

It is best to focus on a star of magnitude 4-5 in V if the seeing is not bad. If the seeing is bad, how sure are you that the problem is indeed focus? Wait till a good night before fooling with focus. You need to lock onto the star with tip/tilt and it is best to do this in the electronics area where you can see the image on the reference camera on a monitor, rather than in a digitized display. Make sure the reference camera shutter is open as well as the shutter for the beam of the telescope in question and, most important, make sure the reference camera is set for a focus at infinity. If you don't know how to do this, consult Judit's document on changing Pops. If you still aren't sure don't even try to change the focus. The shutter for the beam of the other telescope should be closed and the acquisition mirrors should not be in the beam. You should now see the star image on the reference camera and it is a good idea to adjust the intensity gain so you can easily see the structure of the spot. Make sure you know where the gain was before you change it so you can put it back to where you found it when you're done.

To change the telescope focus you need to have the telescope server on the terminal where you are sitting (`ctrscrut`) and have the `wobble` and `tiptilt` clients, or servers, there also. The focus motors are controlled by the PI844 and so the port to that needs to be opened. If it is not open already, open it with the command `opi` and wait for this function to complete. If it fails, make sure the power for the `FOCUS` is on for that telescope. The command to adjust focus using the telescope server is `foc` and it will ask you what step size you want. I have found 0.1mm, or 0.05mm work fine.

Once you are in the focus control routine you can now change the focus one step at a time using the `f` and `F` keys. Since the secondary is moved one motor at a time this will move the star image around a lot, so I have found it best to turn off the tip/tilt servo for

that scope using the `CONTROL` window on the `tiptilt` client, then move the secondary, wait for things to settle and then turn the tip/tilt servo back on again. Do not try to rush this process, things can easily get screwed up.

You can continue this with `tiptilt off`, adjust focus, `tiptilt on` routine until you are satisfied that the telescope is focused as well as possible. Sometimes the motion of the secondary, when doing these adjustments, is so large that you need to move the acquisition mirror back in and re-acquire the star. Make sure you remember to pull the acquisition mirror back out again before you continue to play with the focus.

Unless you are using a scope that needs this connection for controlling the acquisition mirror, close the PI844 connection with the command `cpi` when you are finished. It is also a good idea to ensure that the acquisition image is still close to in focus. Adjusting the acquisition focus is beyond the scope of this document. You will have to ask a someone on the mountain how this is done.

5.5.7.2 The telescope secondary is not where it should be

This is even more serious than telescope focus, and it may mean that the whole telescope optical alignment needs to be redone. Let's hope not. Again, make very sure that this is indeed the real problem. Don't do anything to the secondary unless you are very sure of what you are doing.

If you are convinced that this is the problem it is best to start by measuring where the secondary micromotors really are. Bring the scope down to 15 degrees in elevation and use a vernier caliper to measure the distance between the top of the red micrometer housing and the bottom of the PZT. Not all the PZTs look the same, and some have washers and other spacers. The measurement you want is from the top of the red mount to the first flat surface below the PZT along the shaft. Write these numbers down and make sure you know which is for X, which for Y and which for Z. You will be standing very close to the secondary itself, so try very hard not to touch it or bump it.

Below is a table for all of these numbers last time they where measured on each telescope:

Telescope	Date	X"	Y"	Z"
S1	Nov 11 2005	0.315	0.324	0.321
S2	Sep 12 2005	0.430	0.442	0.344
E1	Nov 2 2005	0.797	0.803	0.821
E2	Nov 2 2005	0.533	0.504	0.524
W1	June 5 2003	0.622	0.468	0.631
W2	Dec 04 2003	0.601	0.606	0.604

Your numbers should not be too different from these. However, these numbers do change and the numbers in the table in this document may not be right. Be vary careful about moving the secondary at all times.

Now you have these numbers you can make sure that the software agrees that this is where the motors are. Like so many of the encoders in our system these are relative and not absolute so it is possible for the software to get this wrong. You can force the software to agree on the position using the command

```
setpos X.XXX Y.YYY Z.ZZZ
```

where `X.XXX` is the position of the X axis and so on. This routine will tell you that this could be dangerous, but if you tell it you are sure it will put you into the secondary control

mode, the same as that discussed above for focus control. Note that none of this will work unless you have opened communications with the PI844 unit with the command `opi`. This routine will display the target position of each micrometer and where it really is. Note that the position is in mm not in inches. This is because it is the same as the reading on the micrometer itself. These displays will get out of synchronization regularly and this can be ignored. The position shown is with respect to the micrometer scale, not the position you just measured, so they may appear not to make any sense.

You can now, slowly, move the secondary around so that it is in a position close to where it should be by using the `x, y, z, X, Y`, and `Z` buttons. Make sure the step size is small, but you can adjust this on the fly. Below is a table of the last know good position for each telescope secondary:

Telescope	Date	Xmm	Ymm	Zmm
S1	Sep 12 2005	24.130	24.034	27.102
S2	June 5 2003	23.927	23.825	23.901
E1	May 28 2003	13.030	13.741	12.954
E2	Mar 16 2004	21.946	22.149	21.971
W1	Nov 3 2004	16.918	20.817	16.689
W2	Dec 4 2004	17.551	17.374	17.450

Once you have moved the secondary SLOWLY into position it should be close to well aligned. However you should check that the focus is good. If things still look bad, the telescope will need a re-alignment.

5.5.7.3 One of the covers will not move

All covers in the domes are controlled by TEMA, and there is a built in time out. If this time out has triggered things will not move. There is a CLEAR TIMEOUT button in the CONTROL panel of the telescope gui, try clicking that and re-issue the command to the cover. If you are in the dome, the magic number for the command on the TEMA panel is 121.

5.5.7.4 The telescope server time is ridiculous

There is a bug, either in the GPS unit itself or the on board PIC, in the GPS based time source of the telescopes. This sometimes results in an impossible time, like `28:34:12`. This can be fixed by reseting the bunker clock using the command `bcreset` on the telescope server. Another useful command is `bcshow` which will display all the current parameters of that clock, including how many satellites it sees and so on.

6 Wobble Computer

The computer called `wobble` had the most serial ports and was for that reason responsible for running many servers. All of these servers have been moved to `ctrscrut`. It is was also responsible for the tip/tilt detectors, for performing the tip/tilt servo calculations, and for sending the results to the appropriate telescope computer. The `tiptilt` computer has taken over most of this functionality although the `wobble` computer is still responsible for distributing the tiptilt data to the various telescopes. This machine is called `wobble` for historic reasons I will not go into[1].

6.1 Wobbler Server

The `wobbler` server is a text based server and is responsible for controlling the photomultiplier based tip/tilt detectors and closing the tip/tilt servo loop. It can be started from the command line using the command `xwobble` or from the `Servers` menu under X windows. This server is now obsolete.

Figure 11. Screen shot of the `wobbler` server

When the server first starts, it does not know which telescopes will be associated with the beam 5 and beam 6 positions, but apart from that it should set up everything it needs to operate. It should show no counts at all in either detector if the high voltage for the photo-multipliers is switched off, which should be the case unless you are actually observing.

---

[1] Try to think of it as Australian for tip/tilt, like Fosters is Australian for Budweiser.

If the high voltage is switched on you should see a few noise counts in the detectors. It is very dangerous to turn on high voltage if the shutters for the detectors are open and or any lights are on.

The `wobbler` client is available from the `Observing` menu under X windows or by using the command `xwobblegtk`.

The photomultiplier based tiptilt servo is now obsolete, in fact the photmultipliers are no longer even on the optical table. The CCD based tiptilt servo now does most of the work of the wobbler system. I have left the following section on possible problems with the wobbler system in for completeness only. It is also true that many of these problems are equally true for the newer tiptilt server.

Figure 12. Screen shot of the `wobbler` client

6.1.1 Things that can go wrong with the `wobble` server

Here is an incomplete list of things that are known to go wrong with the `wobble` server and what you might try to do to fix them.

6.1.1.1 The tip/tilt servo oscillates

Sometimes the tip/tilt servo oscillates, although this doesn't happen very often anymore. It is normally safe to use the `atune` command or the `ATUNE` button on the client and use a sample period of 5mS. If things do go unstable you need to manually tune the servo which you can do by clicking the `TUNE` button on the client or using the `tune` command on the server. Turning down the gain normally helps.

6.1.1.2 Wobble works very slowly

There is a known memory leak in the display of the tip/tilt system in the `wobbler` server. This display is turned on and off using the `BDISP` button on the client or using the command `bdisp` on the server. If things slow down on `wobble` try turning off this display, shutting down the server and restarting it. If things are really bad you may have to kill the server using the socket manager. To avoid this problem use the `REM T/T` button on the client to bring up an alternative display that looks the same but does not have the same memory leak. If the remote display fails to work try turning it off and back on again.

6.1.1.3 There are hardly any counts on `wobble`

You may need to re-align the acquisition target to the laser spot. This can be done using the `ALIGN ACQ` buttons on the `grand_wazoo`.

Alternatively, the star may be too faint. Try increasing the sample period by using the `SAMPLE PERIOD` button on the client or with the command `period` on the server. You can also change the default sample period on the `grand_wazoo`. If this still doesn't work it may be that the neutral density filters in front of the detectors are in place. You can move them out using the `NDB5` and `NDB6` buttons on the shutter client.

6.1.1.4 The system complains that a star is too bright

There is not much you can do about this. It should have already closed the neutral density filters. It may be that the shutter server has died and needs restarting and so while it thinks it has moved the neutral density filters nothing has actually happened. It is possible to turn down the voltage for the photo-multipliers, but only do this if you are sure you understand what you are doing.

6.1.1.5 Wobble can not talk to the shutter server

Perhaps the shutter server has died, use socket manager to check this. If it has and the connection has died, or if it has restarted and `wobble` didn't know about it, you can reopen the connection by typing the command `oss` on the `wobbler` server. This is true for any server that needs to talk to the shutters.

6.1.1.6 The Tip/Tilt servo doesn't work

There are many reasons why the tip/tilt servo won't work. I will try to cover some of them here. Naturally, you should ensure that the power is on for the tip/tilt amplifiers and that the servo itself is turned on. You should also check that `wobble` is talking to the right telescope by looking at the status window. Normally the `grand_wazoo` sets this up but it may be wrong.

Apart from that, the first thing to check is that the communications between the `wobbler` server and the telescope servers are up and running. Try using the `WOBCOM` button on the `grand_wazoo` to turn them on. You should also check with the telescope and lab power RPC units that the `COMMS` power is turned on. If the communications still fail, that is the `Rec Rate` is zero, something more serious is wrong. We have had two of the communication boxes fail since we first installed them some years ago and it is likely more will fail in time.

If you suspect that the communications is bad type a command of the form

```
sin S1 el
```

in the `wobble` server. You should see the star moving around with a one Hz signal. If it isn't the communication is not working. If it is moving, but in no regular manner, the communications may have lost a byte and got out of sync. Stop the signals on wobble, wait for the receive rates to go to zero on the scopes and start the signals again.

Another reason why the servo may be unstable is that the servo parameters are not right. Try running the autotune function by using the `ATUNE` button on the client or the `atune` command on the server. If that doesn't work you can try manually tuning the servo with the `TUNE` button or, you guessed it, the `tune` command. Generally if the servo is oscillating it means the gain is too high, so first up try turning down the proportional term of the servo.

The servo can also fail if the rotation matrix loaded in the `wobble` system is not correct. The `wobbler` computer needs to rotate the tip/tilt signal into the frame of reference of the M7 mirror, and the telescopes then rotate that into their own frame of reference. The former is a constant angle and can be set on the server using the command `rot`. The angle should be set to 39.852 degrees. The latter changes with the pointing direction of the telescope. The telescope rotation matrix is computed every ten seconds and downloaded to the real-time kernel. Check the telescope status window and look at the `rotation` field. It should not say `Turned off`. If it does you can start it again using the command `starttcs` on the server.

If the tip/tilt servo is on and you have checked everything else and things still don't seem to be working it may be that the telescope is not listening to the tip/tilt signals, even though they are reaching it. In the telescope display check that the `Servo` field says `Wobb/Xoff`. If it doesn't click the `TIP/TILT` button on the client in the `POINTING SERVO` area of the telescope client, or type the command `servo wob` on the telescope server.

If all of this fails to make it work you can try to go through the entire acquisition sequence manually and see where things fail. Assuming that you have the star in the acquisition field and that it is locked by the acquisition this is what you need to do.

1. Check that you have counts on the detector. Make sure the shutter and neutral density filters for that detector are open. If you don't have counts, you need to re-align the acquisition origin with the laser, try using the `ALIGN ACQ` buttons on `grand_wazoo`.

2. Once you see counts, tell the telescope to start listening to the tip/tilt signals from `wobble` by clicking the `TIP/TILT` button in the `POINTING SERVO` area of the telescope client or by typing the command `servo wob` on the telescope server.

3. Turn the `wobbler` servo on for that telescope by using the `CONTROL` window of the client or by typing `on E1` on the server, where naturally you replace `E1` with the name of the telescope in question.

If it still fails there is one more thing to check, and that is whether the tip/tilt detector itself is still fairly close to aligned on the optical axis of the system. To do this you need to remove the photo-multiplier and replace it with a small paper target. With this done you look at the laser spot on the target. This is best done by Judit or some other experienced person on the mountain. It is easy to damage things doing this, please be very careful. Most important of all, ensure the high voltage power is off before you touch anything near the detectors.

Finally, I should mention that if the telescope compumotor servo tuning is not good it can cause problems in the tip/tilt system. Again, this needs to be fixed by someone who

has done it before. If the servo locks but the tip/tilt mirror seems to move around a lot or oscillate this is also a sign that the telescope compumotor servo is not tuned well.

7 Tiptilt Computer

The CCD based tiptilt server has replaced the old photomultiplier base wobble server. However, much of what has been said about the wobbler server is also true of the tiptilt server so it's a good idea to check the previous chapter for problems.

7.1 Tiptilt Server

The tiptilt server controls the new CCD based tiptilt detection system.

Figure 13. Screen shot of the `tiptilt` server

Before you start the `tiptilt` server, you must ensure that the power to the cooling system and the CCD itself is on. It is extremely important that the cooler be running before you turn on the CCD and is only turned off if you are sure the CCD is **NOT** running. You can start the server with from the X windows menu or with the command `xtiptilt`.

Note that unlike the `wobble` system there are background counts and read noise to deal with. Whenever you change the frame rate, please ensure that the bias frame is OK. The server will attempt to load an old bias frame that should work, but if things are not working try making a new bias frame by ensuring the detector is in the dark and hitting the `MKBIAS` button in the `CONTROL` menu of the GUI. You can also type `mkbias` in the server window. If no old bias frame exists the server will print an error message to this effect, but it is mostly harmless. Just make a new bias.

The `tiptilt` client is very similar to the `wobble` client and has many of the same functions. The `grand_wazoo` knows most everything it needs to know about the `tiptilt` system acquisition is, on the whole, an automated process.

Figure 14. Screen shot of the `tiptilt` client

7.1.1 Things that can go wrong with the `tiptilt` server

Here is an incomplete list of things that are known to go wrong with the `tiptilt` server and what you might try to do to fix them. Note that many of the problems that occur using the `wobble` system can also happen to the `tiptilt` system, including servo oscillation, loss of communications and so on. WHile these have been repeated here, please check that list as well.

7.1.1.1 The tip/tilt servo oscillates

Sometimes the tip/tilt servo oscillates, although this doesn't happen very often anymore. If things do become unstable you need to manually tune the servo which you can do by using the `tune` command on the server. Turning down the gain normally helps.

7.1.1.2 The tip/tilt servo complains about the CCD

Is the CCD turned on? When the `tiptilt` server starts up it tries no more than five times to communicate with the CCD. If they all fail it will give up. If this happens, try cycling the power to the CCD and try again. If this still fails, connect to the `tiptilt` machine and type the command `rtccdAPIDemo`, which should return with no errors. Try this command a few times, but if it still fails there is a more serious problem. Turn off the CCD and reboot `tiptilt`. If it still fails, I am afraid you are in serious trouble.

7.1.1.3 The tip/tilt system doesn't seem to be talking to the telescopes.

First, like the `wobble` system, the `tiptilt` server can send sin waves to the telescopes, see if anything happens. Also like the `wobble` system there are buttons to turn on and off sending signals, and the telescopes must be told to receive them. You should also check that the power is turned on for the fiber-to-serial converters.

The `tiptilt` server sends its signals to the `wobble` server which then distributes them to the telescopes. If the `wobble` server is already talking to a particular telescope it will ignore the signal from the `tiptilt` server. The upshot of this is if you inadvertently have the `wobble` system talking to a scope it will override the signals coming from the `tiptilt` system. The quickest fix is to either type the command `SIGOFF` in the `wobble` server, or hit the same button in the wobbler GUI. Restarting the `wobble` server will also work. The `tiptilt` server has a field in its status window that will show you the current frame rate, and the `wobble` and `telescope` servers have fields to show you the receive rates. All these numbers should agree. If they don't you have a problem.

7.1.1.4 The frame rate is always zero even though the camera is working

This is a bug, that I hope to fix soon, in the real time part of the CCD code. It is caused by the clock in the `tiptilt` system either not running at all, or having been set to a time very different from the last time the CCD ran. For the time being the only solution is to reboot `tiptilt`, but do so from the lab and go into the BIOS. Make sure that interrupt 11 has been set to ISA legacy, save the BIOS and reboot.

Sometimes syncing the clock can also cause this problem, but that should be fixed soon. If it does, exit the `tiptilt` server, log in as root and reload the `tiptilt` model using the following commands:

```
/sbin/rmmod tiptilt_rt
/sgin/insmod /usr/local/modules/tiptilt_rt.o
```

7.1.1.5 The dam thing keeps crashing

Stuff happens. While I think this has been fixed, it does still happen on occasion, especially when you change frame rates or display images. Work continues in this area. For the time being you have no choice other than to reboot the machine.

7.1.1.6 The servo tuning is no good

We have yet to find good tunings for all conditions. Right now the default tuning is for the fastest frame rate and has not been tried on many stars. You can play with the tuning of the servo using the command `tune`. Be careful, you can easily cause servo oscillations and loose the star.

7.1.1.7 The counts are way too low

Try using a slower frame rate. Also ensure that the acquisition is properly aligned with the laser. See the `wobble` section above.

7.1.1.8 The counts are negative

The bias frame is bad, get a new one or turn it off.

7.1.1.9 I changed the exposure time but the frame rate didn't change

Sometimes the exposure time command is ignored by the camera, we don't know why. Send the command again.

7.1.1.10 I like the pretty pictures, can I run them all the time?

It used to be that displaying images on the tiptilt system was a sure way to crash the computer. This was related to an interrupt conflict between the floating point processor and the CCD and has now been resolved. Still, the image displays are for debugging the system so try and resist the temptation to use them unless you have to. If you do need to, you can turn on the ability of using the displays with the command `tad`.

7.1.1.11 The tiptilt clock or frame rate is wrong

This is due top a conflict between the tiptilt clock and e CCD driver. While we still do not understand this problem it is harmless and you should ignore it.

7.1.1.12 One of the beams keeps throwing the star away.

This is most common with beam 5. There is a fault in the CCD hardware that can change the bias. Make a new bias and see how things go. IF that doesn't help try recycling the power on the CCD. It might also be that another beam combiner is observing a very bright star and light from this star is spilling over into the pixels of your beam combiner. If you are the primary observer you can ask them to wait, or else you can create a beam specific bias for your beams using the `DBIAS` button on the `tiptilt` GUI. Keep in mind that this bias is only valid while the bright star is being observed and can cause lock failure should that bright star go away. These beam specific bias frames can be removed using the `ZBIAS` button on the `tiptilt` GUI.

It is also possible that the communications have gotten out of synch. Stop sending data from the tiptilt to that scope by hitting the "SEND" button on teh servo control for that scope, type `sflush` in the telescope server, then restart the communications with teh `SEND` button again.

8 OPLE Computer

The `ople` computer is responsible for the shutter server and the server that controls the delay lines and the longitudinal dispersion correctors.

8.1 Shutter Server

The shutter server, oddly enough, controls the shutters in the laboratory. When the module that actually does the work first loads, the default is for all shutters to be open and this is often not what you want. For this reason it is very important to turn off the power to the shutters before re-booting `wobble` or reloading the modules. Like the picomotors the shutter server is quite stable and when it does fail can normally be restarted using the socket manager. The manual command to start it is

```
/usr/local/bin/tsockman remove shutters
/usr/local/bin/shutter_server /ctrscrut/chara/etc/shutter.cfg &
```

and the client, like all GTK clients, is available via the `Observing` menu under X windows or can be started with the command `xshutgtk`. On this client the buttons that are pressed down mean that that shutter is open.

Figure 10. Screen shot of the shutter client

A final thing to know about the shutters is that several of the other servers try to open a connection to the shutter server when they start. Right now these are the `classic` server, the `wobble` server, the telescope servers, and the `tiptilt` server but the number is always increasing. If the shutter server is restarted while these other servers are running they will start to do very odd things. You can ask these servers to re-open their connection with the shutter server using the command `oss`.

If, on rebooting the machine, some shutters don't work it maybe that the on-board computer for those shutters is in a strange state. In that case, stop the server, remove the module, turn off the shutter power, wait a few minutes, and then reverse the process. That is, turn on the power to the shutters, load the module and start the server. Alternatvely, if the number of shutters not working is small, you can just walk over to them, unplug them and then plug them back in.

8.2 The OPLE Server

The `ople` server is a text based server and is responsible for controlling the delay lines and longitudinal dispersion correctors. It can be started from the command line using the command `xople` or from the `Servers` menu under X windows. There can be up to three `ople` servers running at once, one for each beam combiner running on the system. These are called primary, secondary and tertiary. Note that there must always be a primary `ople` server running.

Figure 15. Screen shot of the `ople` server

The `ople` client is available from the `Observing` menu under X windows or by using the command `xoplegtk`. You can have up to three `ople` servers running at any time, the

primary, the secondary and the tertiary. This is only useful if you are running more than one beam combiner at the same time.
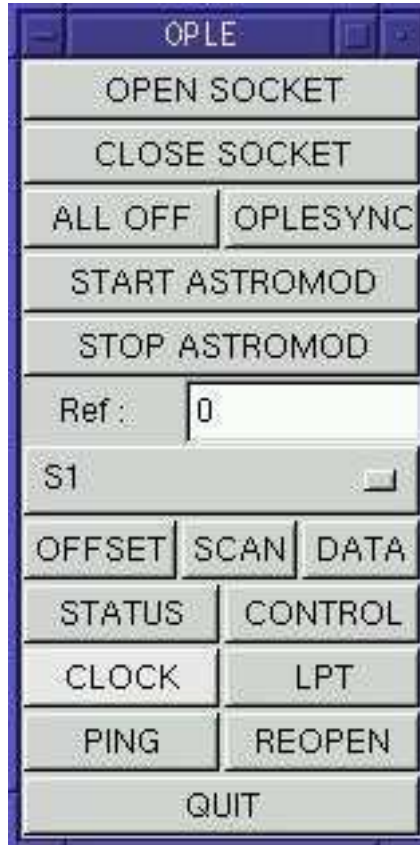
The `ople` server on startup attempts to open up all the sockets and serial ports it needs, though as with many of these systems, it can fail. One important thing to remember is that the cart hardware is actually controlled by the three VME base processors in the big rack of electronics for the OPLEs and that our `ople` server simply makes requests of these computers. If these computers are not running when the `ople` server starts it will complain that it can not speak to them. You close or open this socket connection from the GUI or by using the server commands `oo` to open it and `co` to close it. You must make sure and close this socket before turning off the OPLE electronics.

Another socket connection you need to be aware of is the one referred to as the `tracking socket` and it is a link between the `irimage` server and the `ople` server used when scanning for fringes. Fringe scans will fail if this socket connection is not working. Refer to the section on the `irimage` server for more information.

It is the `ople` server that calculates the cart positions and astrometric model and sends them to the carts. It also calculates the velocity required to meet the next target.

It is also possible to send an offset to this model position. To achieve this, each cart keeps track of two independent targets, one called the `open loop` target and the other the `manual` target. Each of these targets consists of a time, a position for that time, and a velocity,

which define a straight line in delay line space for the cart to follow. These two targets are summed and become the actual target for the cart. In normal operation the open loop target is used for the astrometric model and the manual target is used for fringe scans and delay offsets. The `CONTROL` panel of the `ople` GUI allows you to turn these various targets on and off and to set the manual target. The `OFFSET` button simply sets the manual target.

A final word about the delay lines. Never, ever move them for the first time on a given day without being inside the delay area and with your eyes on the carts themselves. There is a rare but persistent boot problem with the VME computers that inverts the sign for one of the sensors and this causes the carts to jump and bump all over the place. This is a very bad thing, and if it happens stop all the carts, close the socket and re-boot the VME computers. Then try again. If they move smoothly the first time, they will continue to do so all night.

You should keep an eye on the delay lines as much as you can while observing. Never change direction quickly, always stop them, wait and then restart them. If you are using a long east/west baseline the carts can be moving pretty quickly to track the fringes. Please ensure that you don't ask the carts to reverse direction quickly when searching fro fringes.

8.2.1 Things that can go wrong with the `ople` server

There are lots of things that can go wrong with the delay lines, in fact, if you do not find fringes it's a pretty safe bet that the problem lies with the delay lines, or something to do with them. I will try and list here as many of the common faults as I can, but I bet there are more to be found.

8.2.1.1 The carts make bad noises and vibrate a lot

The sign of the voice coil sensors has reversed on boot up. You should have been reading the earlier section. Stop all the carts as soon as possible and reboot the VME rack. Make sure things are OK before leaving the lab.

8.2.1.2 The OPLE computers/clock have stopped

There is a bug in the OPLE VME computer code that means it freezes at UT midnight. Check the time, if it has gone past UT midnight (as defined by the VME rack, remember it may not be set to the real time) you will need to reboot the VME computers.

If it has not gone UT midnight it probably means that the GPS clock has stopped also, go and check that.

8.2.1.3 When I start the `ople` server only a few carts show up in STATUS

Each of the three VME computers controls two carts and if they are not all running you will not see status messages from all six carts. Sometimes one of the computers fails to startup properly. One solution is to reboot the VME computers, but you can try to do something a bit more graceful first.

To begin with, the VME computers will not run unless the GPS clock itself is running so check that first. If you are sure the GPS clock is running you can open up a text terminal and connect to one of the VME computers using the command

```
rlogin vme1
```

which should give you a command line prompt of the kind `->`. There are only three commands you need to know here. The first is a request to list all current threads on the CPU and that command is the single character `i` followed by a return key. You should see a list of many processes, more than thirty. If you only see a small number of processes listed, like about ten, it means that this CPU did not start all of the required jobs. You can ask it to try and start them using the command `PowerOn`. Wait for few seconds and try listing the threads again and see if they have increased in number. Repeat this process with the other two VME computers `vme2` and `vme3`.

If all this fails to get things going, reboot the VME computers.

8.2.1.4 The `ople` server has locked up

This happens once every few nights and we still are not sure of the cause. The symptoms are that the `ople` server stops listening to the GTK GUIs and will not respond to the keyboard either. Despite this it still seems to be happily reading the astrometric data and send things on to the VME computers. It may not be clear that this has happened as by looking at the status window nothing seems wrong. You will probably discover that the `grand_wazoo` complains or can not ping the `ople` server. At the time of the most recent update of this document this problem seems to have been solved.

The first thing to try is to hit the `ESC` key on the `ople` server a number of times. This sometimes brings it back to life. If this still fails the only thing to do is to kill the server, which can be done from the socket manager, manually with a `kill -9` command on `ople` or using the X windows kill menu option. You should then remove the socket manger entry for `ople` and restart the server. It will in all likelihood complain of lock files and so on which you can safely ignore. Once the server is running reopen the connections to it on the `ople` GUI and the `grand_wazoo`. It is also wise to tell the `irimage` server to reopen the tracking socket using the command `ots` on `irimage` or by using the `TRACK SOCKET` button on the `grand_wazoo`.

It is likely that you will have to re-sync the clock, restart reading the astromod and put the carts back into tracking mode to continue observing.

8.2.1.5 The `ople` clock has stopped

Make sure the GPS clock itself is OK. If so, there is a known problem[1] with the CCD driver that can stop the `ople` clock. You need to re-initialize the clock software using the command `cinit` on `ople`. You should re-sync the clock once you have done this.

8.2.1.6 The delay lines move to odd places or complain about astromod

Make sure the `ople` server and the OPLE rack are both properly synced to the GPS clock. If these clocks are different very odd things can happen.

8.2.1.7 I checked that and it still complains about astromod

The `astromod` program is now obsolete as these calculations are now done within the `ople` server itself. It is still possible to use the `astromod` program, but it is not recommended. If you do have this issue, it may be that the clock on the computer running `astromod`,

---

[1] which I think we have solved.

normally ctrscrut, has a bad clock setting. The `astromod` program uses the normal Linux clock and date as a starting point and if they are wrong the model will be no good. Since the `ople` server now does these calculations itself, rather than read them from a file, this problem is obsolete.

8.2.1.8 `ople` complains about a cable puller

This could mean one of two things. The best thing is that it is a simple mistake, so try restarting tracking on that cart and see if things improve. If that works all is well. If it doesn't you will need to go into the lab and see what has gone wrong.

8.2.1.9 The carts don't seem to move.

Stop everything. Go into the lab and see what is wrong. Have the cables got tied up? Is there something on the rails? Never try to keep moving them if the metrology signals do not show movement when you expect to see it.

8.2.1.10 The tracking error jumps up every 10 seconds

This happens when the `ople` server sends a target to the VME computer with a time stamp in the future. There is a bug in the VME code that causes this to deliver very strange values for the demand position. This is because it uses an unsigned integer for time and performs the calculation based on the difference between the time now and the time tag of the target. I believe the problem has been resolved. If it does happen again please let me know. I leave the follow in for completeness.

The way the astrometric model works is that it calculates cart positions for the current star based on the current baseline model for every ten seconds. The `ople` server takes these numbers and works out the velocity required to get the cart to where it should be in the next ten seconds. Every now and then the timing for this gets mixed up. Go to the `ople` GUI and hit the **STOP ASTROMOD** button, wait a few seconds and then hit the **START ASTROMOD** button. This should solve the problem. You can get the same effect using the commands `stop` and `start` on the `ople` server.

If this doesn't work check that the `ople` computer clock, the `gps` clock, and the `vme` clocks agree on the time. In the `ople` server type the command `mon`. This will give you a display of the cart positions, targets, errors and the time. Naturally, this time, and the time of the `ople` server should agree. If they don't, try resynching all the clocks again. The command `syncople` typed in the `ople` server window will force the VME clock to be the same as the `ople` server clock.

In some rare cases the `ople` computer has been left running on the local CPU clock, rather than the CHARA clock. You can check this by opening a terminal on the `ople` computer and typing the command `lsmod`. If you see

```
Module                 Size   Used by
chara_rt_clock_card    10160   0 (unused)
mbuff                   6048   1 [chara_rt_clock_card]
rtl_sched              42720   0 (unused)
rtl_fifo                9888   0 [chara_rt_clock_card]
rtl_posixio             7152   0 [rtl_fifo]
rtl_time                9872   0 [rtl_sched rtl_posixio]
rtl                    26464   0 [chara_rt_clock_card rtl_sched rtl_fifo
```

```
    rtl_posixio rtl_time]
    3c59x                           26048    1 (autoclean)
```

all is well. If you see module called `prio_clock.o` the system is running on the local
CPU clock and you should either remove that module, or simply reboot the machine.

In 2007, the code was changed so that the `ople` server calculates and send new astro-
metric targets to each cart 2 second apart. Therefore, each cart is updated every 12 seconds
now, not 10 as quoted above.

8.2.1.11 The metrology doesn't work

No fringes? Carts not finding the home sensor when tracking to zero? Don't see anything
in the metrology monitor window? You have metrology problems. Adjusting the metrology
is a fiddly process and not for the faint-hearted. I will not attempt to write up everything
here that needs to be done, but I will try and list a few things that often work.

The two main problems encountered are a lack of signal strength in general and a process
called self-interference. The former can be due to mis-alignments in the laser box itself or
in the feed optics. The later is only due to the feed optics. Note that since 2009, when
Laszlo added some band pass filters to the metrology electronics, the self interference has
been much less of a problem.

Since mid-2007, the reference signals are generated outside of the rest of the metrology
optical system. Still, sometimes they can be low. This is a problem that needs to be fixed
by someone trained in the adjustment of the reference system so if you have trouble with the
reference signals you need to let someone who knows how to fix it know about the problem.

The definition of a low signal depends on the delay line in question. In general you want
a signal of 1 volt peak-to-peak for things to work, with an absolute minimum of 0.5 volts.
We have found that the two southern lines and the W1 line can have signals in the range
of many volts, while the east lines have lower working signals.

The first thing to test is that there is sufficient light coming out of the laser box itself.
Disconnect the output fibers from the distribution box and use the meter to test the signal
strength in the two fibers, which should be around 7dB. If they are much lower than this
the laser box itself needs alignment. Go find Judit or Theo and ask them to fix the problem.
Under no circumstances should you play with the optics inside this box.

If the laser signal strength is good it may be that the metrology is not well aligned with
the rails. Move the cart to the back of the track and place the two standard alignment
masks on the rails, one in front above the home switch and one in back as done during the
standard alignment procedure. Now disconnect the matching 'S' fiber from the distribution
box and connect it to the visible green laser. You should see the green laser light on both
targets. The alignment with the first mask, the one near the front of the rails, is not critical.
So long as it is within a few mm of the correct position leave it there. The position of the
spot on the far mask is more important. Adjust the MET1 mirror for that delay line until
the spot is well centered. Note that defining well centered is not easy with the green laser
as the beam quality is poor. Just do the best you can.

Now remove the mask at the back of the track and check that the green spot hits
somewhere near the target on the rear of the front mask. If it doesn't, there is a problem
with the OPLE optics. Go find Judit and get her to help. If it does hit this target, remove
the target and move the delay line to near the front of the rails. Now re-connect the 'S'

fiber to the IR laser and see if there is now a metrology signal of some kind. You may have to adjust the MET2 mirror of this line to get a strong signal. If you get no signal at all, go back to the green laser and use the MET2 mirror to autocollimate the green laser beam. With this done go back to the IR laser and see what you can do using the MET2 mirror. If all this fails you are in deep, so try starting it all over again. A good rule to remember is that the MET1 mirror adjusts how the laser goes down the rail, while the MET2 mirror adjusts how it gets back into the metrology box.

Even after all this, sometimes the metrology signals are poor. A final thing to try is to put a corner cube directly in front of the output hole on the metrology box in question and see how the signals are. If there is a strong signal with the corner cube it means you have probably done something wrong in the alignment of the laser with the rails. If you do not see a strong signal from the corner cube there is a more serious problem, because it lies within the box itself. Of course, you should make sure all the fibers are connected in the right way and have no dust on the ends. On rare occasions the output fibers are bumped and are not in the correct position. Unfortunately they can move little bit, even when in their sockets. You might, if you are feeling brave, try fiddling a bit with the output fibers and see if that helps. Even removing them and putting them back into place can sometimes help. The worst cart in this regard is W1 whose signal can go from many volts to millivolts just by moving the output fiber around.

If, given all of this, you still see no signal from the corner cube, the last resort is to try and move the 'P' input fiber. This is the top-most input fiber on the X/Y stage. Be extremely careful with this as you can easily get lost. Never move it very far. You should be able to tweak up the signal with the corner cube in place by adjusting this fiber, but please only try this if you have tried everything else. Once you have found a good strong signal you will need to check that the laser is still aligned with the rails.

So now we have a metrology signal, but we are not sure how well it works. The last part of the alignment is to try and tweak up the signal and minimize the self interference. You can now sit near the `ctrscrut` screen and watch the oscilloscope. Make sure you are looking at both the reference and the unknown signals on the scope and move the cart back. You will need to adjust the MET2 mirror for this cart and try and find a position where the signal strength is good and there is minimal self interference in the unknown signals. This can be very frustrating, believe me. I have found that for both E1 and W1, while you can get a lot of signal strength, the best performance is at a position with less signal strength but also much less self interference. The two southern lines work best and seem easiest to align while E2 has low signal strength but only a little self interference. Bear with this, you'll find the sweet spot eventually.

Once you are done, home the cart and then move it back and watch the metrology as is moves. If you have the patience go back all the way and then track to zero. If you are short on patience just go back as far as you can bear. Now ask it to track to the zero position and wait for it to get there. The home sensor should light up. If it doesn't, something broke and you need to tweak things some more, though I would recommend trying to re-home and do the experiment a second time.

This was added 2005-07-07. Here is described another technique that has been successfully used for some of the cranky metrology systems (read, the east lines). Park the cart on the back limit switch. Set the metrology monitor to the reference and unknown signal of the system being tweaked. Adjust the MET1 mirror to the front target using the green

laser described previously. Remove the target. Power up the metrology laser and maximize the signal by moving mirror MET2. Adjust the metrology head P-fiber to maximize signal strength. Now send the cart to home and further tweak the metrology head P-fiber to minimize the self-interference. Do this in small increments. When the zero-crossings are minimized, **STOP**! Further refinement at this stage can lead to saturation when the cart is at the front. It should get to home without having any skipped zero-crossings. Once the cart is homed, send it all the way to the back switch. Going backwards, the self-interference will be worse. If the signal undulates but avoids skipping a zero-crossing (i.e., the metrology monitor signal doesn't turn red), resist the temptation to make the signal better. Chances are, any changes here will make things inviable at the back end. If the metrology monitor goes red, adjust the metrology head P-fiber to compensate. Avoid moderate to large adjustments. If the P-fiber had to be adjusted, repeat from the homing process; otherwise, the job is done.

8.2.1.12 Metrology looks OK but things still aren't working

As noted above, you can have self-interference in both the reference and the unknown signals of metrology, though we only normally monitor the unknown. See the notes in previous section.

8.2.1.13 The cart is moving very slowly when slewing

Could be you changed direction too quickly. Stop the cart and wait for a few seconds then restart it. If it still moves slowly stop all carts and go into the lab and see if you can find something in the way.

8.2.1.14 W2 cart has intermittent tracking problems

W2 cart is cart number 6, and was added to the system after it was delivered by JPL to us. Unfortunately, the CPUs and electronics where designed for only five carts and so W2 runs on the same CPU as S1 and the master control program itself. This means the CPU is slightly overloaded and if you have S1 and W2 running together W2 will sometimes not receive enough CPU time to track properly all the time. If S1 is not running this should not be a problem. The solution is to use W2 as the reference cart, leave it switched off and use the "AUTO" setting on the central scrutinizer.

8.2.1.15 One of the carts oscillates about 1 cm around the target.

If a cart is very close to the target when you start tracking the cable puller can get confused and the whole thing goes into an oscillation mode. This is the 1 cm odd oscillation. The fix is to back the cart up, or move forward at least 10 cm, stop the cart and then start tracking again. After this things should be fine. If the target is moving quickly it is best to ensure the cart approaches the target position while moving the same direction it will move when tracking. Otherwise it will change directions too quickly and they don't like that very much at all.

8.2.1.16 When scanning for fringes on a large baseline the cart doesn't track well

On large baselines the carts need to move very quickly. This is especially true of E1/W1. This means, when you scan for fringes, you may be asking the cart to change directions

very quickly, and this is never a good thing. Stop the cart, move it away from the tracking position in the opposite direction of target motion, that is if the target is moving forward, move the cart back and visa versa. Often, simply stopping the cart and waiting for the target to move a few cm will have the same effect. This way, when you start the scan, the cart will already be moving in the correct direction and will not cause a sudden motion change. You will need to increase the length of the scan as this procedure will increase the amount of time it takes for the cart to reach the right position in the scan.

8.2.1.17 The OPLE server or VME rack has crashed

If too many commands are sent to the VME rack it can fill the input buffer and crash the machine. This is most common when the `MIRC` beam combiner is running. The maximum rate for changing targets, at the time of the most recent update, is 3Hz. We are now working on improving this rate.

8.2.1.18 One cart tracks but the errors are large, like a many microns

It could be that someone has bumped the gain setting on one of the high voltage amplifiers in the rack. The LED display on these amplifiers ahould show about 17.5 volts when the cart is turned off. Ensure the BIAS is all the way anti-clockwise, that is off, and move the gain until the voltage is about 17.5. Sometimes the display misbehaves, if you can not get it to work, look at the position of the gain knob and ensure it is about the same as the others. If this fails the PZT itself might be broken, so use this cart as reference.

9 Irimage Computer

The `irimage` computer is responsible for talking to the IR camera called NIRO and controlling the dither mirror for collecting data with the IR beam combiner. It is also used by the FLUOR team when they search for fringes.

9.1 The Irimage/Classic Server

The `irimage` server is a text based server and is responsible for controlling the collection of data from the IR beam combiner. It can be started from the command line using the command `xirimage` or from the `Servers` menu under X windows. Since the beginning of 2008 the `irimage` server has now been replaced by the `classic` server and is no longer in regular use. Since the beginning of 2010, the `classic` server is actually a specialized version of the new `climb` server.

The `classic` server is very much like the old `irimage` server with a number of changes. Most of these changes are based on removing a large of amount of legacy code from the old system. The largest change relates to the timing of the data samples. In the old `irimage` server, the data stream was forced to always be 1000Hz, independent of the true sample time of the camera. If the camera sample time was less than 1000Hz, the data were interpolated so that they conformed with the 1000Hz data rate. In the new `classic` server the timing of the data stream is determined by the NIRO camera itself. Another major difference is the data format used by the two systems, with the new `classic` server using the standard FITS file format, rather than the older and less portable binary files used by `irimage`.

Many other small changes have occurred "under the hood" but the look and feel of the two systems is very similar.

Figure 17. Screen shot of the irimage server.

Be sure to have the power to the dither mirror switched off when ever you are not actually using it, and in particular when you reboot `irimage`. It is also very important that you never switch the dither mirror power on or off, unless you are sure that no signal is being sent to it and that the low pass filter in the lab is on. In the bottom right of the status window for `irimage/classic` you will find an indication of whether the dither mirror is off or on. Finally, note that unless the NIRO DOS machine is up and the `niro` program is running you will get no data from NIRO. This is teh default for NIRO.

As noted above, the `irimage/classic` servers expect to be able to connect to the shutter server, and will refuse to run if it can not establish this connection. If this connection is lost for some reason you can re-open it using the command `oss`.

The `irimage/classic` client is available from the `Observing` menu under X windows or by using the commands `xirimagegtk` or `xclassicgtk`.

Figure 18. Screen shot of the `irimage` GUI.

9.1.1 Things that can go wrong with the `irimage/classic` servers

Like most things in the system, there are quite a few things that can go wrong with `irimage/classic`, and next to the `ople` system it is the most likely system to fail. It probably has the most spectacular crashes to date. I will attempt to list the known problems below.

9.1.1.1 Irimage/Classic server gets to "`working...`" and stops

This normally means that the CHARA clock on the `GPS` computer is not running. It may also mean that the NIRO DOS box is not running or has crashed. It's waiting for a few seconds while NIRO is being initialized and if the clock is not running it will sit there forever. Check that the `GPS` clock is running, and if it isn't try running the `autolist` on `GPS`. If this fails, ensure that the `NIRO` computer is running.

9.1.1.2 Irimage/Classic doesn't respond, or has crashed

THIS SECTION NOW OBSOLETE. Note that communications between the DOS box and teh rest of teh world is now done using a socket and not the serial port.

This is related to the fact that we use a DOS machine to drive NIRO and it communicates with the rest of the control system via a serial port. For some reason we have never worked out, this DOS program can get upset and not only crash the DOS machine, but somehow cause a fatal error in the RT-Linux world on `irimage/classic`. There is nothing you can do but go into the lab and reboot both computers. Once you have done this restart the `irimage/classic` server and then use the REOPEN buttons on all the clients that had been talking to it, most likely the `irimage/classic` GUI itself and the `grand_wazoo` program.

9.1.1.3 The picture mode looks backwards or the visibility estimate is strange.

You have probably lost the connection between the `irimage/classic` system and the shutter server. Use the command `oss` to try and re-connect. Without the shutters working the calibration sequence will fail and you will get ridiculous visibility estimates.

9.1.1.4 When I use the HOLD and SEND buttons, the `ople` server does not respond

You have lost the so called `tracking socket` between the `irimage/classic` and `ople` servers. You can use the TRACK SOCKET button on `grand_wazoo` or the OTS button on the `irimage/classic` GUI to re-establish the connection. Alternatively you can use the command `cts` followed by the command `ots` on the `irimage/classic` server itself.

9.1.1.5 Irimage GUI does not update/Irimage complains of lost sockets

Another mystery bug. Sometimes, the `irimage` server loses track of the GUIs connected to it. This is odd, because the code that does this is the same on all of these servers and `irimage` is the only one that does this. The server code has been modified to identify this problem and will react by disconnecting the socket. You should try the REOPEN button on the GUIs and see if the problem goes away. Of course, wait until you have finished collecting data if you are in the middle of a something first. If this fails, the only thing you can do is to restart the `irimage` server.

At the time of the latest update this problem seems to have been solved.

9.1.1.6 The total counts plot is all over the place

Sometimes, most frequently when you change NIRO mode, the serial connection between NIRO and the `irimage/classic` server gets out of sync. You need to stop whatever you are doing and use the RESYNC button in the NIRO window of the `irimage/classic` GUI. Alternatively you can type `ssync` at the `irimage/classic` server prompt.

With the new socket connection this does not seem to happen any more.

9.1.1.7 The NIRO counts are very low or the spots are way off

This is now obsolete.

Make sure you have the right pixels selected in the NIRO window. Sometimes they revert to a very old default. At the time of writing this document the correct values are 2 pixel in size, and the corners are at (9,14) and (9,22). Be warned that these numbers change fairly often and I will not attempt to update this document each time. We do update the

`autoexec.bat` on the NIRO computer, so the safest thing is to look at the numbers there and force them to be the same on irimage.

9.1.1.8 Those blue lines on the `irimage` display are really annoying

These blues lines show the demodulated fringe envelope and are intended to help, but if you don't like them you can make them go away by typing `envoff` in the server window. To get them back type `envon`. This only works on the `irimage` server. If you are using `classic` you'll just have to put up with the blue lines.

10 Whitezone Computer

The computer known as `whitezone`[1] used to run a web server, originally used for the remote operations but has now been retired. If you have a hankering for a really slow machine ask Nils, I am sure he will let you have it.

---

[1] "the white zone is for loading an unloading.... if you have to load or unload go to the white zone. You'll love it, it's a way of life." FZ again.

11 Ctrscrut Computer

The computer called `ctrscrut` was originally intended to be the central control computer of the array, and for the first year or two, it was, hence the name. It is still the file server for the control system and responsible for data archiving. It also runs many servers for the system, as well as serving as an alternative control station if you need to work close by to the lab. Never reboot ctrscrut unless you really have to as all the other computers in the array will stop until it has completed the reboot.

Note that when you `ssh` through the firewall from the outside world to `michelson` you are really connecting to `ctrscrut`. These is also a machine called `ctrscrut` on the office side of the firewall, and this is the machine you reach when you connect to `pease`.

I will list all the servers running on `ctrscrut` below, but please bear in mind that this list is constantly changing, so when in doubt look at the `/etc/rc.local` file on this machine to see the most up to date list of servers and the commands to use to restart them. There are a total of 32 serial ports on `ctrscrut` so it is likely that any new serial port based server will run on this machine.

11.1 Socket Manager

As stated above, the socket manager runs on `ctrscrut`.

11.2 Remote Power Controller (RPC) Units

These daemon-like servers running on `ctrscrut` control RPC units in the rack which in turn control the power to the CCD, cooler, fans, the communications to the telescopes for tip/tilt signals and many other things. Like the RPC servers on the telescopes it is quite stable and you should be able to restart it using the socket manager. The manual command to start these servers is

```
# Start the first RPC unit

/bin/rm -f /var/lock/LCK..ttyS0
/usr/local/bin/tsockman remove RPC_RACK_1
/usr/local/bin/rpc_server /dev/ttyS0 /ctrscrut/chara/etc/rack_1.cfg &> /var/log/rack_1

# Start the second RPC unit

/bin/rm -f /var/lock/LCK..ttyS1
/usr/local/bin/tsockman remove RPC_RACK_2
/usr/local/bin/rpc_server /dev/ttyS1 /ctrscrut/chara/etc/rack_2.cfg &> /var/log/rack_2

# And the server controlling CCD power and other things

/bin/rm -f /var/lock/LCK..ttyC13
/usr/local/bin/tsockman remove RPC_LABPOW
/usr/local/bin/rpc_server /dev/ttyC13 /ctrscrut/chara/etc/labpow.cfg &> /var/log/labpo

# And the server controlling RACK3 Power
```

```
/bin/rm -f /var/lock/LCK..ttyC21
/usr/local/bin/tsockman remove RPC_RACK_3
/usr/local/bin/rpc_server /dev/ttyC21 /ctrscrut/chara/etc/rack_3.cfg &> /var/log/rack_
```

and the client can be started from the `Observing` menu under X windows or with the command `xpower`.

Figure 19. Screen shot of the power client

11.3 ESP servers

The `ESP` servers control the Newport motorized micrometers in various places in the lab. These include the `retro` server that moves the corner cubes on the beam sampling table to find internal fringes, the beam samplers, the periscope mounts, the longitudinal dispersion correctors, and two units in the lab for controlling the light source injection and the beam combiner. At the time of the most recent update of this document, the only one of these servers that the average observe needs to worry about is the beam sampling system.

The beam sampling system, behind the beam reducing telescopes, is the part of the array that allows us to select which telescopes we wish to use for observing. It is actually controlled by a separate unit built by Newport that sits on top of the OPLE electronics rack, and this is connected to one of the serial ports on the rear of `ctrscrut`.

The GUI used to talk to the server can be started from the `Observing` menu under X windows or by using the command `xbsgtk`.
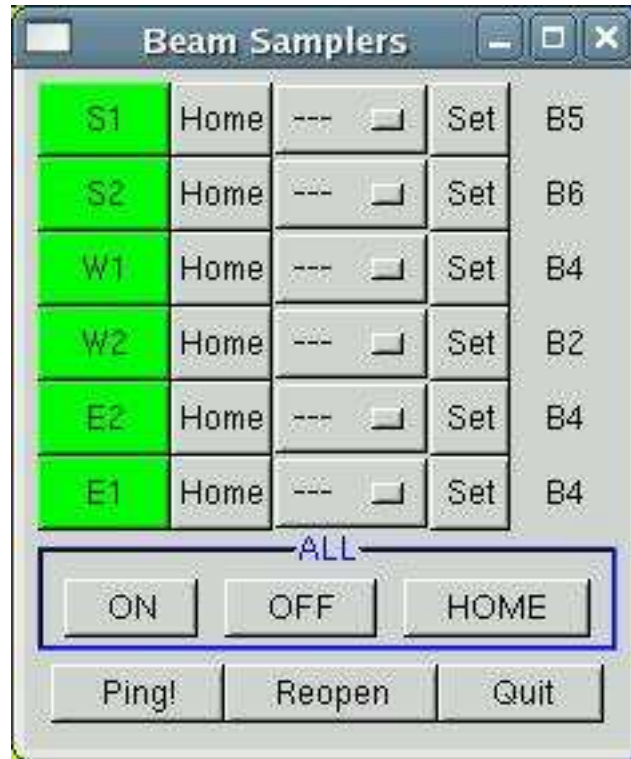
Figure 20. Screen shot of the beam sampler GUI.

The ESP servers and clients have been remarkably stable[1]. and are started automatically on boot up. If one fails, you should be able to restart it using the socket manager. If that also fails, the commands to start it manually are

```
# Start the esp server for RETRO
/usr/local/bin/tsockman remove RETRO
rm -f /var/lock/LCK..ttyC0
/usr/local/bin/esp_server /dev/ttyC0 retro &> /var/log/retro.log &

# Start the esp server for BS
/usr/local/bin/tsockman remove BS
rm -f /var/lock/LCK..ttyC1
/usr/local/bin/esp_server /dev/ttyC1 bs &> /var/log/bs.log &

# Start the esp server for PERI1
/usr/local/bin/tsockman remove PERI1
rm -f /var/lock/LCK..ttyC2
/usr/local/bin/esp_server /dev/ttyC2 peri1 &> /var/log/peri1.log &
```

---

[1] Which may have something to do with the fact that it was written by Dave Berger instead of me.

```
# Start the esp server for PERI2
/usr/local/bin/tsockman remove PERI2
rm -f /var/lock/LCK..ttyC3
/usr/local/bin/esp_server /dev/ttyC3 peri2 &> /var/log/peri2.log &

# Start the esp server for LDC1
/usr/local/bin/tsockman remove LDC1
rm -f /var/lock/LCK..ttyC4
/usr/local/bin/esp_server /dev/ttyC4 ldc1 &> /var/log/ldc1.log &

# Start the esp server for LDC2
/usr/local/bin/tsockman remove LDC2
rm -f /var/lock/LCK..ttyC5
/usr/local/bin/esp_server /dev/ttyC5 ldc2 &> /var/log/ldc2.log &

# Start the esp server for BC1
/usr/local/bin/tsockman remove BC1
rm -f /var/lock/LCK..ttyC7
/usr/local/bin/esp_server /dev/ttyC7 bc1 &> /var/log/bc1.log &

# Start the esp server for BC2
/usr/local/bin/tsockman remove BC2
rm -f /var/lock/LCK..ttyC17
/usr/local/bin/esp_server /dev/ttyC17 bc2 &> /var/log/bc2.log &
```

11.4 Picomotor Servers

The ctrscrut computer also has several incarnations of the picomotor server, one for each
of the picomotor controller boxes in the lab. At the time of the latest update of this
document there are a total of six. These servers allow you to move mirrors around, and
unless you are very sure of which one you are moving this can be quite dangerous. Like
all of the daemon like servers the picomotor servers are started at boot time, but like the
laser metrology server they don't like to start-up with the controller boxes turned off. They
are, fortunately, very stable and if they do crash can almost always be restarted using the
socket manager. If that fails the commands to restart them all are

```
# Start PICO server for PICO #1
/bin/rm -f /var/lock/LCK..ttyC8
/usr/local/bin/tsockman remove PICO_1
/usr/local/bin/pico_server /dev/ttyC8 /ctrscrut/chara/etc/pico_1.cfg &> /var/log/pico_

# Start PICO server for PICO #2
/bin/rm -f /var/lock/LCK..ttyC9
/usr/local/bin/tsockman remove PICO_2
/usr/local/bin/pico_server /dev/ttyC9 /ctrscrut/chara/etc/pico_2.cfg &> /var/log/pico_

# Start PICO server for PICO #3
/bin/rm -f /var/lock/LCK..ttyC10
```

```
/usr/local/bin/tsockman remove PICO_3
/usr/local/bin/pico_server /dev/ttyC10 /ctrscrut/chara/etc/pico_3.cfg &> /var/log/pico

# Start PICO server for PICO #4
/bin/rm -f /var/lock/LCK..ttyC11
/usr/local/bin/tsockman remove PICO_4
/usr/local/bin/pico_server /dev/ttyC11 /ctrscrut/chara/etc/pico_4.cfg &> /var/log/pico

# Start PICO server for PICO #5
/bin/rm -f /var/lock/LCK..ttyC12
/usr/local/bin/tsockman remove PICO_5
/usr/local/bin/pico_server /dev/ttyC12 /ctrscrut/chara/etc/pico_5.cfg &> /var/log/pico

# Start PICO server for PICO #6
/bin/rm -f /var/lock/LCK..ttyC25
/usr/local/bin/tsockman remove PICO_6
/usr/local/bin/pico_server /dev/ttyC25 /ctrscrut/chara/etc/pico_6.cfg &> /var/log/pico
```

The clients for the picomotors are available via the `Observing` menu under X windows or can be started with the commands `xpico1`, `xpico2`, and so on.

Figure 21. Screen shot of the picomotor client

11.5 Laser Filter Wheel

The alignment laser has a filter wheel in front of it containing a number of neutral density filters. This filter wheel is controlled by a server running on `ctrscrut`. It is often useful to

add neutral density when you are looking at the laser, for example in the acquisition image, so that it does not saturate the detector.

Figure 22. Screen shot of the laser filter wheel client

This server/client pair is very stable, though sometimes does fail to move things around. If you suspect a problem hit the IDN button, which should respond with a message like

```
IDN: THORLABS FW102 Filter Wheel Version 1.0
```

If you something else, or nothing at all there is a problem. Try using the REOPEN COMMS button. If that fails you may have to restart the server using the socket manager. The manual command to start the server is

```
# Laser filter wheel
/bin/rm -f /var/lock/LCK..ttyC14
echo "Starting thorlabs_fw102_server"
/usr/local/bin/thorlabs_fw102_server /dev/ttyC14 /ctrscrut/chara/etc/nd_wheel.cfg &> /
```

The client is normally started from the X windows menu, though the command to start it from a terminal is xfilter.

11.6 Metrology Laser Server

The metrology laser server monitors the state of the metrology laser and is able to turn the laser itself on and off. It does this by putting it in and out of standby mode. This server should start up automatically when you reboot the machine, but if the laser controller itself is not powered up at the time it may fail. It can also be started using the socket manager or manually with the commands

```
# Now start up the METLAS server
/bin/rm -f /var/lock/LCK..ttyC16
/usr/local/bin/tsockman remove metlas
/usr/local/bin/metlas_server /dev/ttyC16  &> /var/log/metlas.log &
```

It is unusual for the metrology laser server to crash or stop operating. If you do need to restart it, make sure that the laser controller box is powered up by turning the key on the front panel. You should also ensure that the two blue boxes near the laser are also turned on. If the laser controller is not powered up the server will have trouble starting. The client

for the metrology server can be started from the `Observing` menu in X windows or can be started manually with the command `xmetlas`.
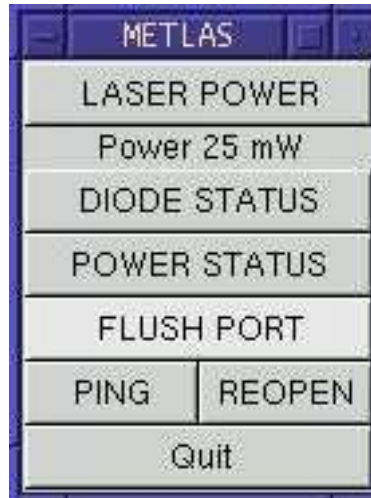
Figure 23. Screen shot of the metrology laser client

While the metrology laser server does its best to check on current status of the laser power it fails to get things right far too frequently. A good indication that this has happened is that it tells you that the power output is 0 or the `LASER BUTTON` stays yellow. This only happens if the controller box is not switched on or if the server has failed to get the true condition of the laser. Clicking on the `DIODE STATUS` and `POWER STATUS` buttons should help, though you may have to do this a number of times. For some reason only understood by the laser manufacturers, if the system tells you the power is around 25 mW it means the laser is off. When the laser is on the power should be close to 200 mW.

11.7 POP and M10 Target servers

At the time of the most recent update of these documents these servers are not in regular use by observers, though they should be later this year (2006). The POP server/client control the positioning of the POPs and allow you to turn the LEDs on and off. The M10 target server/client moves the targets inside the turning boxes up and down. Both are essential for changing POPs, though for now this is only ever done during the day, and almost always by Judit. Until this procedure is automated, or at least easier to do, please resist the temptation to play with them. For completeness the commands to restart these servers are

```
# the server controlling the POPs
/bin/rm -f /var/lock/LCK..ttyC15
/usr/local/bin/tsockman remove pop
echo "Starting pop_server"
/usr/local/bin/pop_server /dev/ttyC15 &> /var/log/pop.log &

# Server for M10 targets
/bin/rm -f /var/lock/LCK..ttyC20
/usr/local/bin/tsockman remove target_M10
```

```
/usr/local/bin/target_server /dev/ttyC20 /ctrscrut/chara/etc/target_M10.cfg > /var/log
```

11.8 The Metrology Monitor Server

There is a digital oscilloscope sitting near the the OPLE electronics rack that can be connected to two of the laser metrology monitor outputs on that rack, and normally it should be connected to the two unknown outputs for the carts you are observing with that night. This system will allow you to switch this connection around remotely, nd even monitor all twelve signals at once. This monitor is helpful for checking that the metrology signals are not too low, and do not contain too much self-interference.
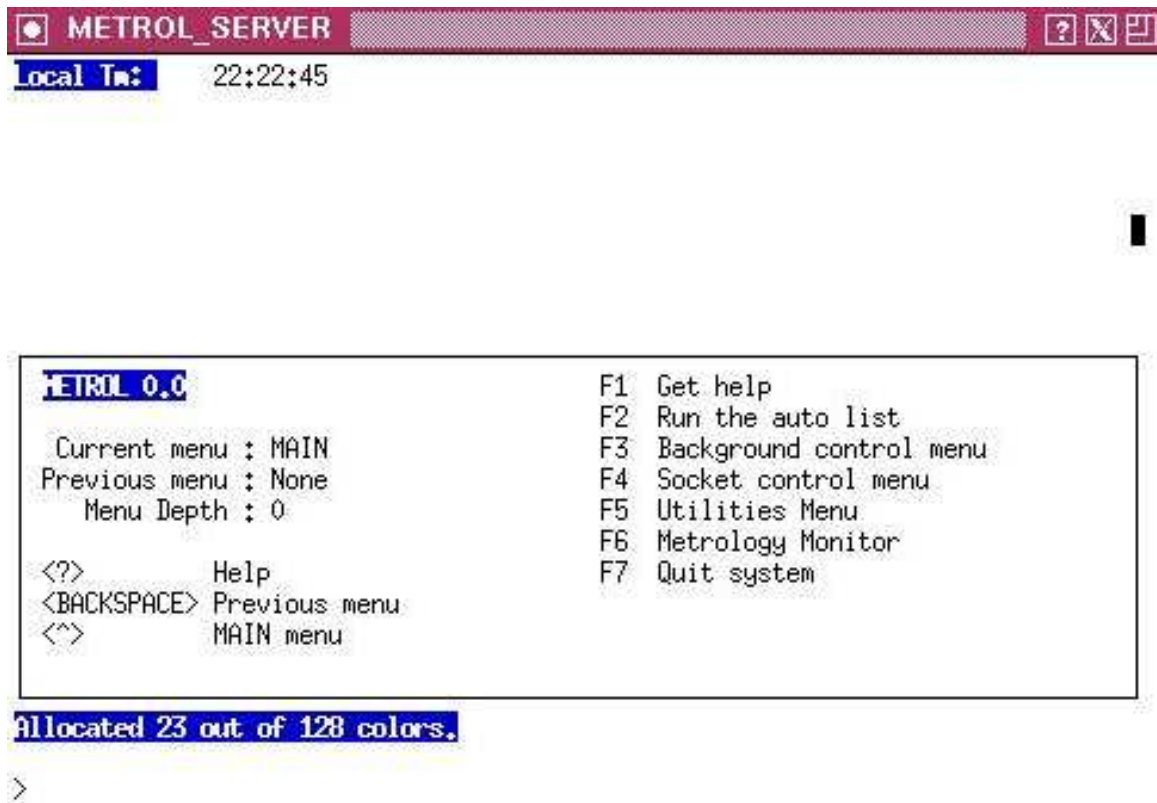
Figure 24. Screen shot of the metrology monitor server

There is a text based server program that can be started from the Server menu under X windows, or with the command xmetrol from a command line that allows you to see the signals on a remote computer screen. When the server first starts, it tries to connect to the scope through a serial port and then download the default setup. If the scope is turned off at the time it will complain, but you can always run the initialization again from the GUI.

This GUI, like all others, can be started form the `Observing` menu under X windows or with the command `xmetrolgtk`.

Figure 25. Screen shot of the metrology monitor client

The metrology server and client are very stable[2]. The only problem you may encounter is that it will not display things where you are sitting, which is probably because it is already displaying it somewhere else.

11.9 The Iris Server

The main iris for the laser source can be controlled using the `iris` server. This allows you to set the iris to a range of sizes, and also to the default beam size. Mostly this is done to get a clear diffraction spot in the beam center at any particular position for beam alignment.

If you need to manually restart the iris server check the `rc.local` file on `ctrscrut` and check the most recent version of the command. At the time of the latest update of this document, this command was

```
# Start the iris server

/bin/rm -f /var/local/LCK..C27
/usr/local/bin/tsockman remove iris
```

---

[2] Even though Dave didn't write it!

```
/usr/local/bin/iris_server /dev/ttyC27 >& /var/log/iris.log &
```


Figure 26. Screen shot of the Iris client.

12 Zoot Computer
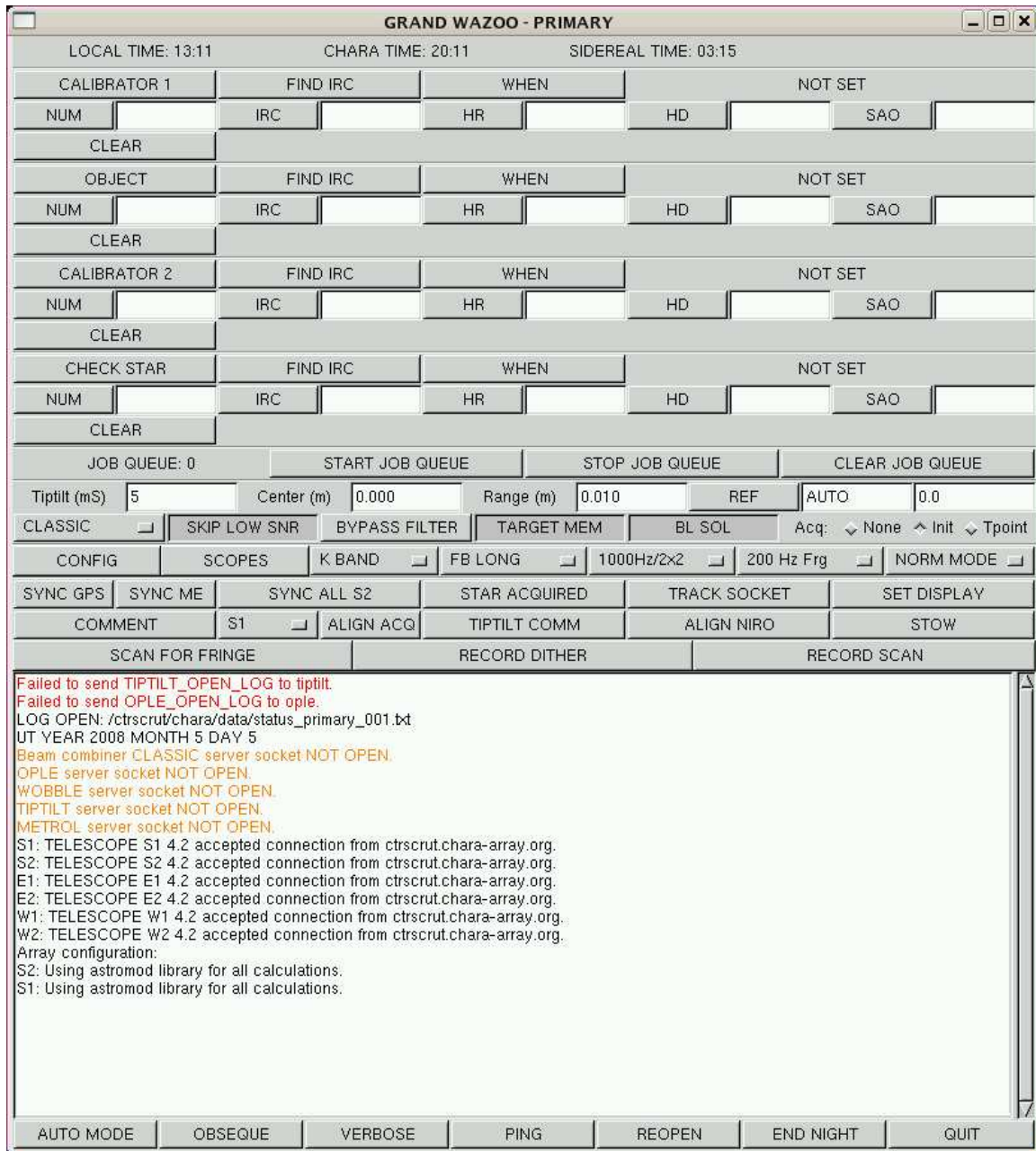
Figure 27. Screen shot of the `grand_wazoo`.

Which leaves us with the last important machine in the array control system, `zoot`, the one used to do the observing when running things from within the control room. There is an identical machine in the remote observing room called `aroc`, and my comments here apply pretty much to that computer also. This computer mainly displays things sent to it by the various servers in the system, runs a large number of GUIs that talk to the servers, runs the main observing program `grand_wazoo`, and also plays the mp3 music files.

There is an alternative machine for use as an observing station called `wazoo` that can do everything that `zoot` can do, although it only has two screens.

Much as I'd like to, I will not try and set out a complete observing manual here, that is for another document, but I would like to list a few potential problems you may encounter and a few tricks you might try while observing.

1. Don't forget that while everything is displayed on `zoot` and all the GUIs are running here, including the `grand_wazoo`, nothing that actually does anything is happening on this machine. So, for example, when the `grand_wazoo` runs the astrometric model, the program actually runs on the machine `ctrscrut`, so if you wish to play with `astromod` manually it is important that you do it on the correct machine. The `astromod` program is now obsolete and all astrometric calculations are now performed by the `ople` server.

2. If the telescope pointing is not as good as you'd like, it can often help to re-initialize a telescope right after you have used the paddle functions to get the star into acquisition. Don't re-initialize once you have closed a servo on the star, as this will make things worse, not better. Please make sure the scope is pointing to the star it thinks it is. Initializing the scope in the wrong place is extremely dangerous.

3. If you fail to find fringes, one of the most common faults is with the delay lines, often the metrology. Always check to see if the carts are tracking and the tracking errors are at the nanometer level. A good test is to turn off all targets and ask the carts to track to zero. Wait for them to get there and see if the home sensor lights up. If it doesn't, there has been a metrology failure, re-home them and see how things go.

4. Another common problem is forgetting to sync the clocks. Check that all clocks agree on what time it is. Sometimes the time and date do not propagate to the TCS computers. You can manually force this using the TSYNC button on the telescope gui.

5. Often when scanning you hit the `SEND` button and nothing happens to the delay lines. You can fix this by re-opening the tracking socket between `irimage` and `ople`.

6. Remote observing takes a great deal of bandwidth. It is a good idea to disconnect the "concam" or cloud monitor computer as it uses a lot of bandwidth and contributes nothing to our observing throughput.

7. The CHARA-CLASSIC beam combiner system always tries to save metrology data during each observation for later use in base line solutions. This can take a great deal of network bandwidth and it is unwise to do this when other beam combiners, notably MIRC, are running. You can turn this feature off on the `grand_wazoo` but hitting the `BL SOL` button.

8. When `zoot` has rebooted the displays often are not initialized correctly. To fix this, log in as root and edit the file `/etc/X11/XF86Config`. At the top of the file you will find a comment that says:

        #NOTE: If one of the screens doesn't work (usually the second one)
        #comment out the following three Screen lines, run X Windows, and then
        #exit out again. Uncomment the lines and proceed as normal.

   If you follow these directions all four screens will work. Naturally you will need the root password to do this, and naturally I will not write it down here.

If you are having a problem you can not fix there is often, though not always, someone to call. Most of us are skype users, so it is best to check and see if anyone is on-line. At

least that way we have the option of ignoring you. Experienced observers will also often make themselves available by phone, but please don't abuse this. It can get very tiring to have to answer questions late at night having already spent all day at work. A good general rule is not to call after 10pm, unless you have been specifically told otherwise. Remember, some of us have a life.

While I am on this subject, it is extremely common to be called at home to fix a problem already listed in this document. Please make sure you have exhausted everything listed here before you annoy someone at home. Note that the information you are looking for may not be in the section you expect. No document is perfect.

Finally, if you hit a wall and can not fix the problem, please be conservative and work on the principle of doing no harm. No science is worth hurting yourself, or causing major equipment damage.

Good luck Mr. Phelps.

Concept Index

Table of Contents