LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Document Type	**LIGO-T000003-00-D**	3/22/00

Suspension Controller Tuning Procedure

Mark Barton, Gabriela Gonzalez, Steve Penn

Distribution of this draft:

DCC

This is an internal working note
of the LIGO Project..

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu/

1 ABSTRACT

The procedure for setting the input and output matrices in a suspension controller is given.

2 KEYWORDS

SUS, suspension controller, LOS, SOS

3 DOCUMENTS

LIGO-T980026-00: LIGO Data Aquisition System Final Design.

LIGO-T980020: GDS Reflective Memory Organization

4 VERSION HISTORY

2/18/00: Version used at initial training session at LHO.

3/8/00: More screenshots.

3/16/00: Update of LLO-specific parts, expanded software section.

3/22/00: Rev 00. More screenshots, corrections to section on `SensMatrix2.m` routine (formerly `PendMatrix1.m`).

5 OVERVIEW

5.1. Aim

The suspension controller electronics convert the OSEM sensor outputs into mirror monitor signals in a position-pitch-yaw basis, using a sensing (a.k.a. input) matrix of 12 weights. After position, pitch and yaw control signals have been generated, they are converted to coil drive signals for the magnet/coil actuators using an actuation (a.k.a output) matrix of 12 weights. The tuning procedure described here tries to compensate for differences in the sensors and actuators so that the monitor signals are as pure measures of position, pitch and yaw as possible, and that the control signals produce as pure displacements in position, pitch and yaw as possible. At the same time it tries to reduce unwanted couplings to the side sensors and actuators.

There are two distinct parts for the tuning procedure: the tuning of the sensing matrix and the tuning of the actuation matrix. As of the draft of this document of 2/16/00, the actuator tuning procedure used only information from the optical levers and so was independent of the sensing matrix tuning. However this may well change.

6 TOOLS

6.1. Hardware

6.2. Software

Apart from occasional trips to the LVEA or mid- or end-stations to set up cables, the entire procedure can be done from the control room using software available on the CDS workstations.

6.2.1. Directory Structure

At LHO, all custom software required for this procedure can be found in the directory

`/opt/CDS/d/ops/sustune/`

or in subdirectories thereof.

The equivalent directory for LLO is

`/opt/LLO/d/ops/sustune/`

All files in these directories should be owned by the user `ops`.

The `template` subdirectory contains software that is intended to be copied and modified for particular optics. Modified versions of software for particular optics should have that optic name (e.g., `ITMX2K`) as part of their name. This includes saved states for the Data Viewer and GDS tools as well as Matlab command files. Data files for particular optics should also normally be named after the optic.

The `burt` subdirectory contains request files (`.req`) and snapshot files (`.snap`) for the backup and restore utility.

6.2.2. Remote login and file transfer

The material in this section should not be relevant to the eventual final version of the procedure but is included for the convenience of people working on improving it, especially the custom software that has been written. Improved versions of the custom software will commonly need to be copied from one observatory to the other.

Observatory computers are divided into two classes: GC (general computing) and CDS. GC computers have full internet access. CDS computers are on a private network local to the observatory. Communication between the CDS network at an observatory and the outside world is via a gateway machine. These are `blue.ligo-wa.caltech.edu` at LHO and `london.ligo-la.caltech.edu` at LLO. (london is sometimes unreliable and `nashville` is an alternative.)

Remote logins can be made to GC computers running Unix using either `rlogin` or `ssh` (`telnet` is disabled). However `rlogin` is a security risk, especially from computers outside the observatories or at a different observatory, and `ssh` is highly recommended whenever available. Good computers for remote logins (i.e., fast, full complement of software) include `rainier.ligo-wa.caltech.edu` and `decatur.ligo-la.caltech.edu`.

The gateway machines do not support `ssh`. It is possible to `rlogin` to the gateway computers from any other Unix machine, but this is a security risk. To minimise the risk, first use `ssh` to a GC computer at the same observatory as the gateway and then `rlogin` to the gateway.

Files can be transferred to or from most GC computers with `ftp` (again, not including the gateways), but this is a security risk and `sftp` (secure ftp) should be used whenever available.

The gateway at LHO, `blue`, has both the GC and CDS filesystems mounted, so files can be copied in and out of the CDS network using `cp`. The gateways at LLO, `london` and `nashville`, have only the CDS filesystem, so the only way of doing transfers is to log in to one of those machines and use `ftp` (or `sftp`). Whether using `ftp`, `sftp` or `cp`, be sure to be logged on as `ops` before adding files, to ensure that they have the correct ownership.

6.2.3. Medm

The status of the mirror controllers and the optical levers is monitored and changed using Medm. If the full site display (`site.adl`, see Figure 1) is not already running, right click on the workstation desktop and choose “MedmLaunch: Full Site” (or TBD for LLO) from the popup menu. The suspension controller and optical lever displays can be found in popup menus on the site display.

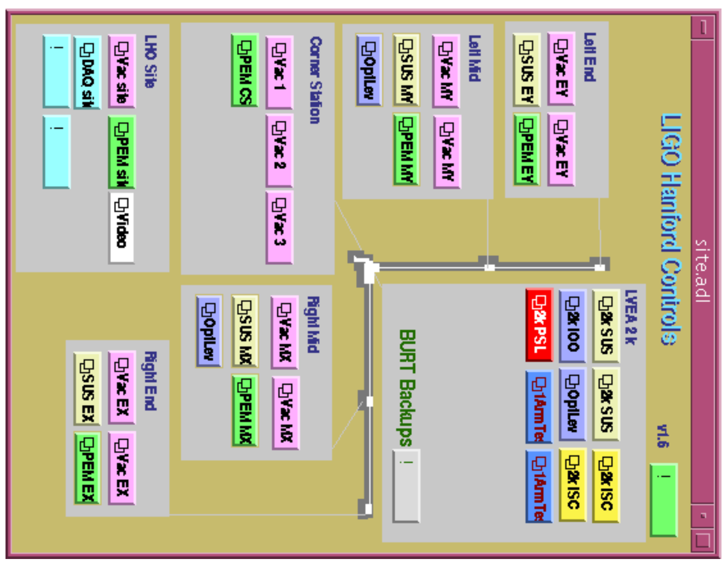

Figure 1: MEDM main site display

Each suspension controller display (see Figure 2) is organized as a block diagram. The sensing matrix is viewed and adjusted in three subdisplays, one for each monitor (position, pitch and yaw). Each subdisplay is accessed through an icon on the left of the position, pitch or yaw path in the diagram. Click once on the appropriate icon to examine or adjust the weights (see Figure 3). Each subdisplay has four weights, one for each sensor (UL=upper left, LL=lower left, UR=upper right, LR=lower right). The default value for these weights is 25.

The actuation matrix (see Figure 4) consists of four subdisplays, one for each of the UL, LL, UR, and LR coils. In each set, there are three weights corresponding to position, pitch and yaw. The four subdisplays are accessed through the icons after the LSC and ASC inputs and before the coil

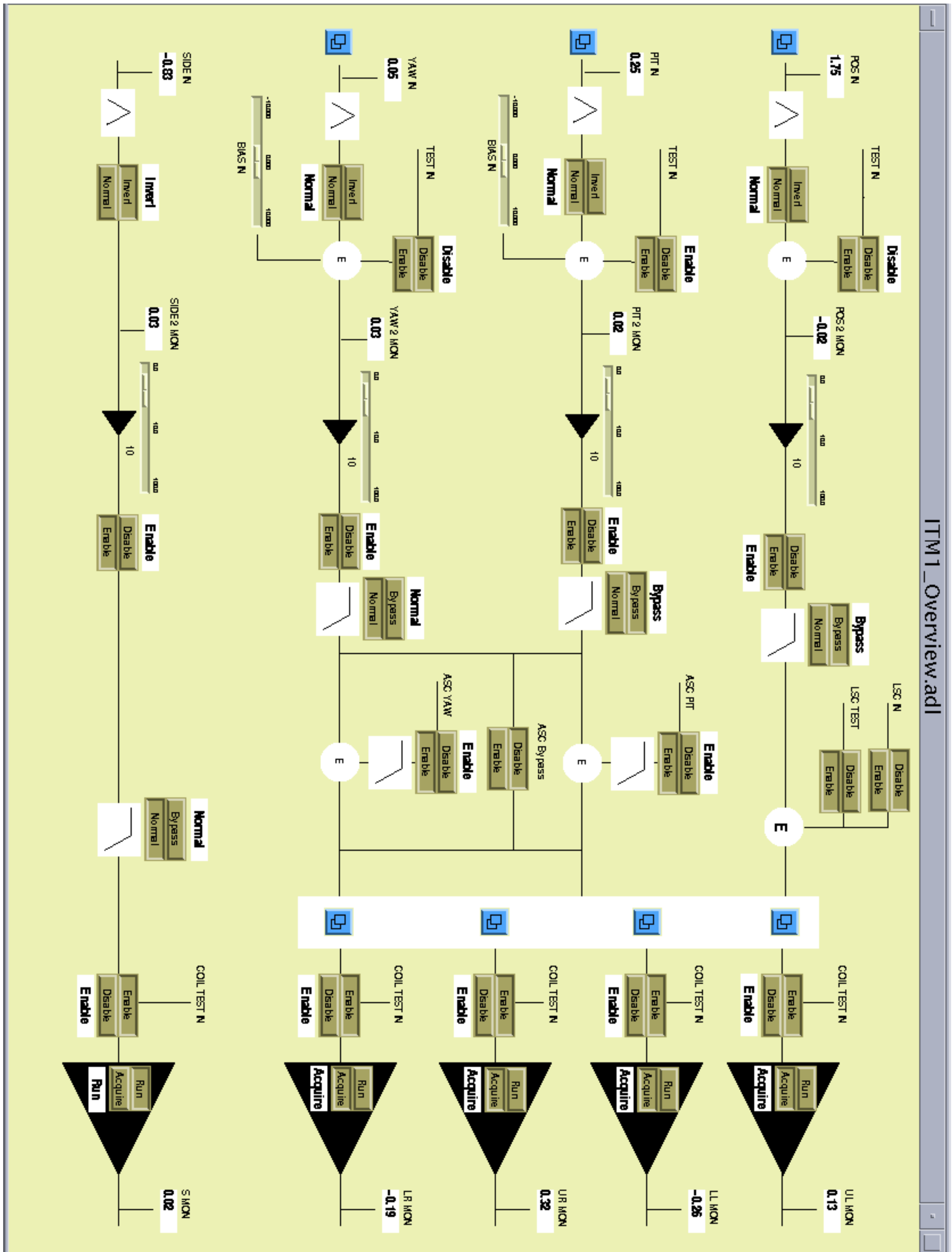

Figure 2: Suspension controller display for ITMx2K

test inputs in the controller diagram. The default value for these weights is 100 (although the controllers are commonly initialized to 50).

6.2.4. The Data Viewer

The DAQS Data Viewer is used to monitor the OSEM sensor and optical lever signals for plausibility. To start it, bring up a terminal window and type “dv”, then “start0” or “start1” depending on whether you want the output to appear on the left or the right screen of the workstation. (At

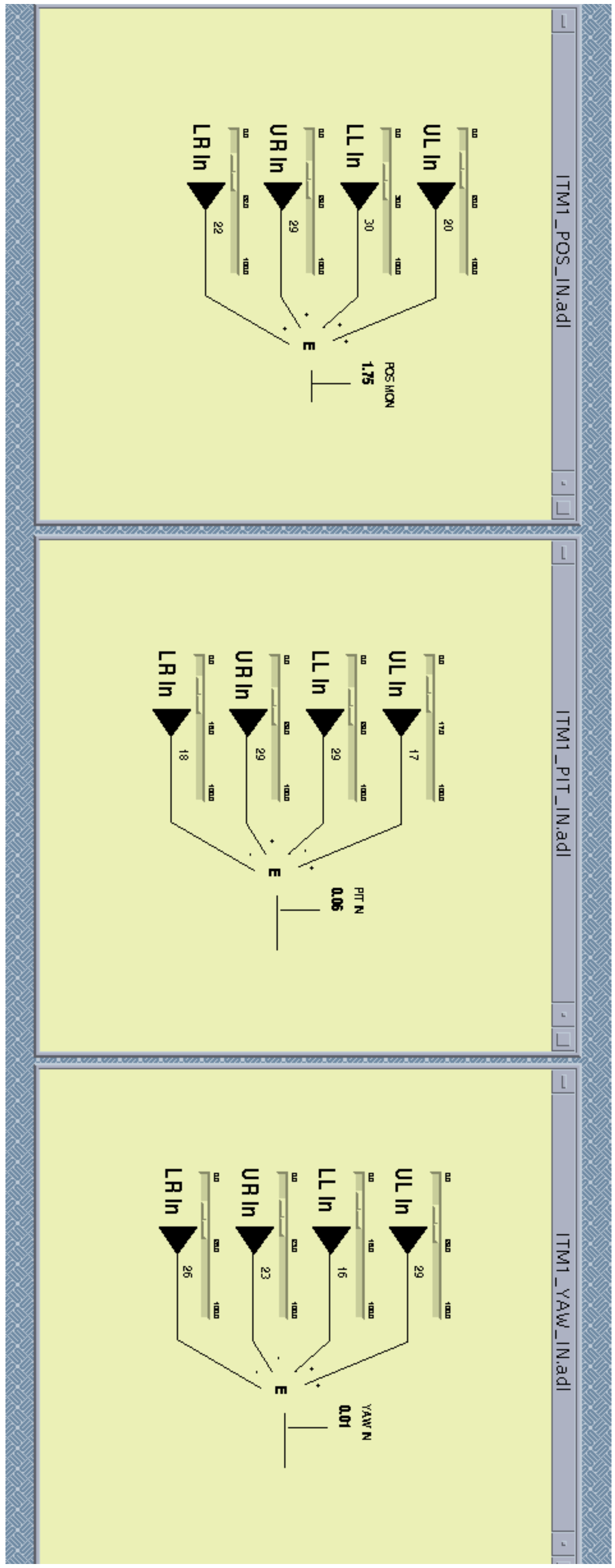

Figure 3: Sensing matrix displays for ITMx2K

LLO, right-click in the desktop and choose the appropriate command from the pop-up menu.) Master lists of the signal names available within DAQS are given in

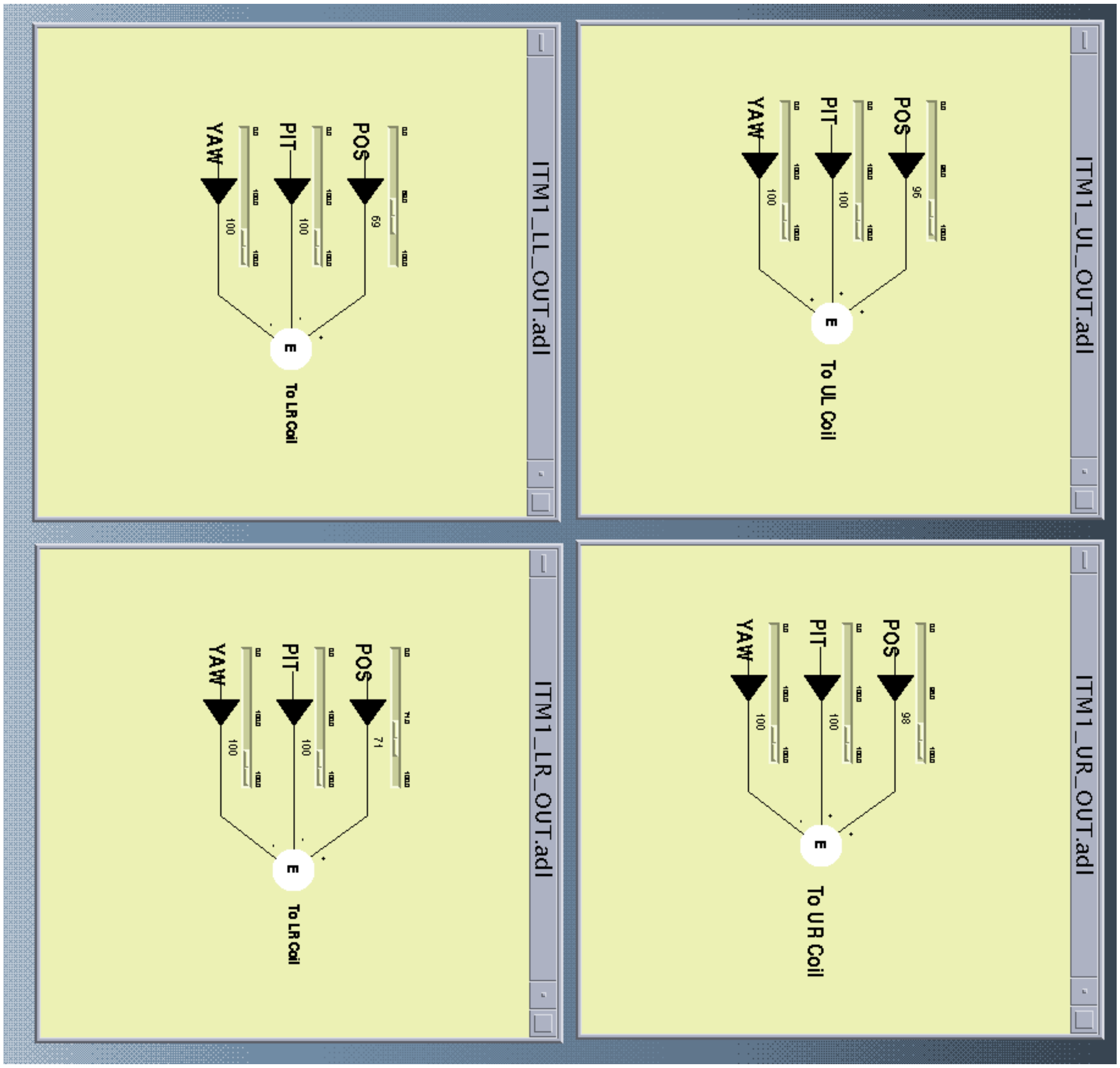

Figure 4: Actuation matrix displays for ITMx2K

<`http://gateway/daq/conf.fast`> (or from outside, <`http://blue.ligo-wa.caltech.edu/daq/conf.fast`> - TBD at LLO.)

and

<http://gateway/daq/conf.slow> (or <http://blue.ligo-wa.caltech.edu/daq/conf.slow> - TBD at LLO).

The example signal names in this procedure are given for ITMx of the 2K interferometer at Hanford. Signal names for other optics can be obtained by substituting appropriate values for “H2” and “ITMX”.

6.2.5. GDS Diagnostic Tools

The GDS Diagnostic Tools are used for the actual measurements in this procedure. To start the GDS tools, right click in the workstation desktop and choose GDS Diagnostics Tools from the popup menu. Signal names are as for DAQS.

6.2.6. Matlab

Much of the data analysis is done by programs running under Matlab. This is installed only on the machine control0 (currently not installed at LLO). To use it on any other machine, e.g., control7, proceed as follows:

1. Bring up a terminal window.
2. Do an rlogin or ssh to control0.
3. Type either “`setenv DISPLAY control7:0.0`” or “setenv `DISPLAY control7:0.1`” depending on whether you want to have Matlab graphical output appear in the left or right screen of your workstation.
4. Type “`matlab`”. Note that this does not run Matlab, it only sets up directory stuff.
5. Type “`start`”.

When Matlab starts, set the working directory to the `sustune` directory with a command like

```
>pwd /opt/CDS/d/ops/sustune/ % for LHO
```

6.2.7. XML Utilities

The command `/opt/CDS/a/gds/bin/xmldir` (`/opt/LLO/a/gds/bin/xmldir` at LLO) gives a directory of the information contained in an XML file.

The command `/opt/CDS/a/gds/bin/xmlconv` (`/opt/LLO/a/gds/bin/xmlconv` at LLO) converts an XML file to ASCII.

6.2.8. BURT

BURT is the backup and restore tool for use with Epics. It is useful for saving particular states of the sensing and actuation matrices and restoring them quickly when desired. It is run by giving the command

```
burtgooey &
```

in a terminal window. Full documentation can be found at <`http://www.aps.anl.gov/asd/controls/epics/EpicsDocumentation/ExtensionsManuals/Burt/BurtManual.book.html`>. Briefly, a BURT request (`.req`) file consists of a set of control names to be backed up (plus optional C-style comments). The control name for a control in a Medm display can be found by pressing the middle mouse button over the control. The name can also be conveniently entered into the request file by dragging with the middle mouse button from the control to a text editor window showing the file. BURT processes the request file, producing a snapshot

(.snap) file with the current values of those controls. Later, BURT can read the snapshot file and restore the saved values.

7 GENERAL SETUP

1. Make sure that all the servos are in RUN or ACQUIRE mode, whatever is the appropriate mode for that suspension. This is particularly important with the initial version of the controller electronics because (i) the coil readback circuits are commonly blown by transients when ACQUIRE mode is selected, and (ii) the bias voltages change between modes, putting the optical lever out of range.
2. If an optical lever is available, bring up the appropriate optical lever display. Check that the optical lever is mechanically zeroed, or at least within range (+/- 1), with a significant sum (>0.5).
3. Ask the appropriate person (Richard McCarthy at LHO or Rusyl Wooley at LLO) to connect fast optical lever DAQ channels to the mirror's optical lever, and confirm they are working (there are only four temporary channels allocated for this purpose, and the actual wire from the DAQ to the optical lever has to be switched from mirror to mirror).
4. If the GDS test outputs for the LSC test input (H2:LSC-ITMX_EXC) and the ASC pitch and yaw test inputs (H2:ASC-ITMX_P_EXC) are available, they should be used for excitation. Otherwise, wire up general-purpose GDS outputs (e.g., H2:LSC_MOD_1_EXC) to the appropriate test inputs of the controller.
5. Bring up the Data Viewer, and restore the saved state

```
/opt/CDS/d/ops/sustune/templates/allchannels.dvsave
```

. (or a saved state specific to the optic if it is available from an earlier round of tuning). Save it for future use with a name incorporating that of the optic, e.g., `ITMX2K.dvsave` (and somewhere outside the `template` directory).

6. The signal names in the template are the sensor channels, optical lever channels (the fast 2048 S/s channels in IOO, not the 1 S/s channels in ASC) and the excitation channel for the 2k ITMx: H2:SUS-ITMX_SENSOR_UL, H2:SUS-ITMX_SENSOR_LL, H2:SUS-ITMX_SENSOR_UR, H2:SUS-ITMX_SENSOR_LR, H2:SUS-ITMX_SENSOR_SIDE, H2:IOO-ITMX_OPTLEV_1, H2:IOO-ITMX_OPTLEV_2, H2:IOO-ITMX_OPTLEV_3, H2:IOO-ITMX_OPTLEV_4, H2:LSC_MOD_1_EXC. Change them to the analogous signals for the optic being tuned. (Note: optical lever signals are not available for all optics.) Set all the y-axis limits appropriately so that the traces are centred and fully visible. Ensure that the ranges for related channels are the same, e.g., 0/-32000 counts for the optical lever channels (~full range) and ~4000 counts around the mean for the sensors (a safe range for the mirror motion). Save the changes. See Figure 3 for a typical output display.
7. Make a note of the optical lever position and the offsets of the sensors. This is particularly important - these positions need to be restored later. The sensor offsets should all be between -10,000 and -25,000. Anything outside this range indicates trouble for this sensor: the magnet is too far from its correct position, and the mirror may need a specialized tuning.
8. Make a note of the initial values of all biases, all gains, and the input and output matrices for all degrees of freedom, manually or with BURT.
9. Do a quick test drive of the optic by reversing all the servos for a second or two and then turn-

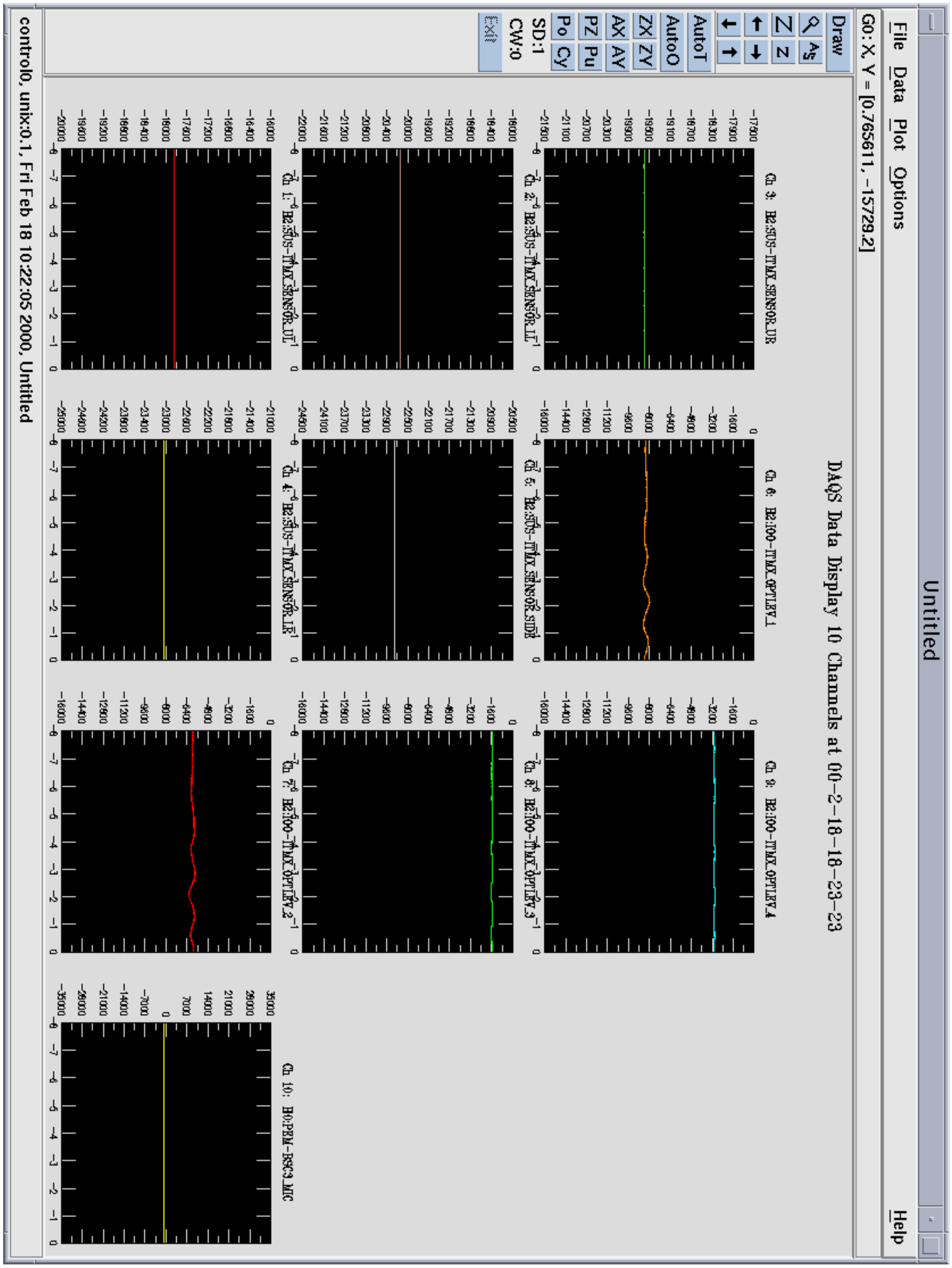

Figure 5: Typical Data Viewer output screen with no excitation

ing the servos back on. You should see all the sensors in the DV oscillate. If you don't then something is wrong in the hardware (talk to Richard McCarthy at LHO or Rusyl Wooley at LLO) or in the software (talk to Dave Barker at LHO or TBD at LLO). If everything looks as it should then reenable the servos and wait a few minutes until optic is fully damped and at rest.

8 TUNING THE SENSING MATRIX

1. Using a text editor, open the Matlab command file template

   `/opt/CDS/d/ops/sustune/getSensorDataTemplate.m`

2. Save a copy of the file with an appropriate name (e.g., `getSensorDataITMX2K.m`) and edit it as appropriate for the optic being tuned as explained in the template. This will be referred to as the DataGet script, after the name of the low-level routine used to get the data.
3. In the MEDM window for the optic, perturb the optic by reversing servos for a few seconds and then turning the servos off.
4. Make sure Matlab is running. Set the working directory to `/opt/CDS/d/ops/sustune.`
5. Run the DataGet script: at the Matlab prompt enter the name of the script chosen above (less ".m").
6. After the DataGet script finishes, there will be a data file in the directory `/opt/CDS/d/ops/sustune` with an appropriate name (e.g. `itmx_data`) chosen during the editing of the script in step 2. Load the data file into MatLab's workspace as follows:

```
> clear all;% clears the Matlab workspace, not required but useful
> itmx_data;% or whatever filename was chosen in step 2.
```

7. Save the data as a Matlab binary file with an appropriate name such as `itmx_bindata.mat`:

```
> save itmx_bindata data* % stores all variables (data*)
                          % in the binary file, itmx_bindata.mat
```

8. Give the Matlab command `SensMatrix2('DataFileName')`.

A typical run of `SensMatrix2` looks like the following:

```
>>SensMatrix2('itmx_drive_all');


The best fit pitch distance (nominally 7.2 mm) is  7.49 mm


 Best Transformation Matrix if Side Sensor is used:
  0.2480  0.1505   0.2608  -0.0724  0.2104
 -0.3210  0.1806  -0.0809   0.2795  0.2564
 -0.2064  0.0226   0.2420  -0.2152 -0.2865
  0.0141  0.1259  -0.0884   0.1382  0.9520


 Best Transformation Matrix if No Side Sensor information:
  0.2576  0.1438   0.2713  0.2211
 -0.3285  0.1924   0.2765  0.2585
 -0.2096  0.2401  -0.2138 -0.2886
```

```
Set the Sensor Gains to the following values

        UL     LL      UR     LR
Pos    25.9   14.4   27.3   22.2
Pit   -33.0   19.3   27.8   26.0
Yaw   -21.1   24.1  -21.5  -29.0


OL_Pitch,Yaw = R(0.216 rad) * [(1.91e+03,4.51e+04) .* OSEM_Pitch,Yaw +
(1.36e+03,-3.14e+03)]
```

9. Check to make sure that the nominal and best fit pitch distances are close.
10. Make sure that the sensor gain values appear reasonable (ie. approximately 25±10).
11. Don't worry too much as yet about the OL-to-OSEM coordinate transformation listed on the last line.
12. Check that the graphs you get are similar to Figures 6-9 below. Most importantly check:

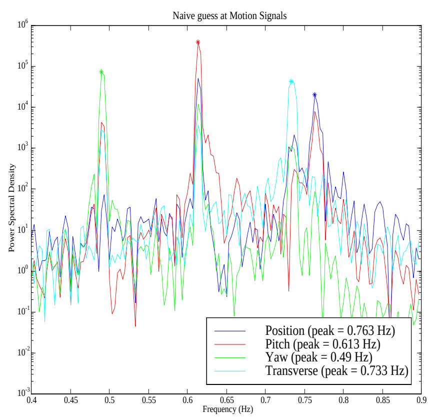

Figure 6: Typical output from `SensMatrix2.m` - I: Power spectra of the monitor signals derived from the untuned sensors. Note that, for example, at the pitch frequency (0.61 Hz) there is energy in not only the pitch monitor signal but also in the position, yaw and side signals. That is, the tuning is poor.

- that the peaks for the signals (indicated by *) have been correctly identified,
- that graph 3 (Fundamental modes) shows each signal with one dominant peak and that that peak has no strength above background from any other signal
- that the listed frequencies are typical (f_X Ý.76 Hz, f_PÝ.61 Hz, f_Y Ý .49 Hz, f_T Ý .73 Hz)

13. If everything looks good then bring up the three displays for the position, pitch and yaw components of the sensing matrix and enter the suggested sensor gains. Use BURT to snapshot these values for easy restoring later.

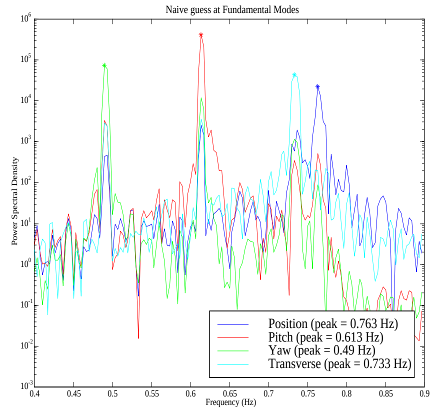

Figure 7: Typical output from `SensMatrix2.m` - II: As for Figure 6 but in terms of modified "position" and "pitch" coordinates which allow for position-pitch cross-coupling. The pendulum mode is a pure displacement in the new "position" coordinate, which is mostly a the original (purely horizontal) displacement, but with a small admixture of pitch. Similarly, the pitch mode, is a pure "pitch" displacement.

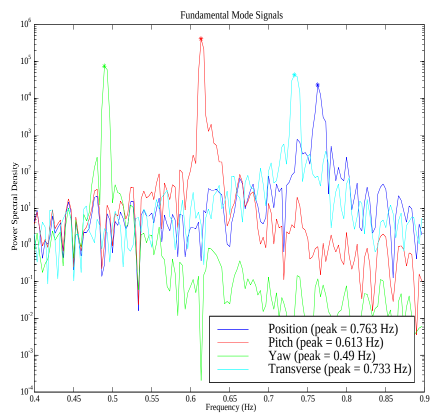

Figure 8: Typical output from `SensMatrix2.m` - III: As for Figure 7, with differences in the gains of the sensors corrected for. Note that, in contrast to Figure 6, each mode frequency has energy in only one coordinate, as desired.

9 TUNING THE ACTUATION MATRICES

Somewhat different techniques are required for the position, pitch and yaw parts of the actuation matrix

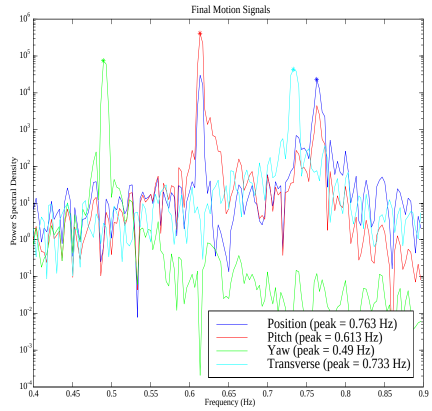

Figure 9: Typical output from `SensMatrix2.m` - IV: As for Figure 8, with the position-pitch coupling correction removed. This is what the normal modes look like in the original lab-frame position and pitch coordinates.

9.1. Position DOF

9.1.1. Method I, using optical levers only

9.1.1.1 Applicability

The optical levers give more accurate pitch and yaw information than the OSEM sensors and so this procedure should be used when they are available. Most large optics have optical levers permanently installed.

9.1.1.2 Overview

The first stage of this procedure involves making five measurements of the response of the optic in pitch and yaw when it is pushed in what is nominally position by the actuators: one with all the gains set at 100, and one each with perturbations to the gains of the top, bottom, left and right pairs. These are saved in files with names like OptLevLeft.xml, according to Table 1. The Matlab command file `ActuationGains.m` processes the data and writes another set of suggested measurements to the file `ActuationResults.txt`. The first of the suggested measurements is a first approximation to the optimum settings, which should be substantially better in pitch. If it is generally satisfactory, the procedure can be halted. However due to cross-coupling between the pitch and yaw degrees of freedom and the fact that the pitch correction is typically larger, the improvement in yaw is usually inadequate (or even negative) and another round of optimization is required. In this case the remaining measurements should be performed and saved to files, again as in Table 1. The second stage data is processed by the Matlab command file `ActuationGains2.m`, which is almost identical to `ActuationGains2.m` except that it looks in `ActuationResults.txt` to find the gain settings rather than assuming the initial value in Table 1. If necessary, a third or fourth stage of optimisation can be done using exactly the same procedure as for the second stage.

Table 1: Gain settings for which measurements are to be made, and the files to which the data should be saved

Measurement Number	*Stage 1* *UL UR* *LL LR*	*Stage 2, 3, etc* *UL UR* *LL LR*	*Filename*
1	100 100 100 100	As suggested in the file `ActuationResults.txt` written by the Matlab command `ActuationGains.m` or `ActuationGains2.m`.	OptLevInit.xml
2	90 90 100 100		OptLevUp.xml
3	100 100 90 90		OptLevDown.xml
4	90 100 90 100		OptLevLeft.xml
5	100 90 100 90		OptLevRight.xml

9.1.1.3 Choice of measurement frequency

The results from this procedure vary considerable depending on the frequency used for actuation, due to the position-pitch coupling. At high frequencies (above the pitch and pendumlum modes at around 0.7 Hz), the optic acts like a free mass and the gains are approximately equal because they are compensating mostly for differences in strength of the magnet-coil pairs. At low frequencies the compliance of the pitch and pendulum modes dominates and it is necessary to push substantially harder (of order 25%) on the upper magnets to keep the optic upright as it moves in pitch. The decision as to which optics should be tuned at which frequencies has not yet been made at the time of writing. In the interim, all optics should be tuned at both high and low frequencies as explained in the detailed procedure. Whichever of the resulting two sets of gains is finally chosen can then be entered into the controller at any time.

9.1.1.4 Optical lever signal ordering

This procedure assumes that the four signals from the quadrants of the optical lever photodiode are numbered in order around the circumference, and that the the photodiode is mounted at either 0 or 90 degrees to the vertical. The algorithm used by the analysis software will be oblivious to left-right and/or up-down flips of all four quadrants, and will notice and correct for a 90 degree rotation, but may give nonsensical results if there is a flip of just two quadrants.

9.1.1.5 Detailed procedure

1. Ensure that the excitation channel is active and connected to the “LSC test in” LEMO input on the front panel of the mirror controller.
2. In GDS tools, open the file `OptLev.xml` in the directory `/opt/CDS/ops/sustune/` or TBD.

3. In channels 0 through 3 in the excitation pane select the optical lever signals (e.g., H2:IOO-ITMX-OPTLEV1, H2:IOO-ITMX-OPTLEV2, H2:IOO-ITMX-OPTLEV3 and H2:IOO-ITMX-OPTLEV4). Do not include other channels - the analysis software assumes that there will be exactly four channels in the output file.
4. In the measurement pane, choose a power spectrum measurement ("Fourier Tools") and measure the noise floor from approximately 0 to 4 Hz.
5. For tuning at high frequency, examine the power spectrum from the previous step and choose a frequency near 2 Hz where there is a sharp dip in the spectrum (i.e., somewhere between seismic stack resonances). Typical frequencies are 1.8 Hz for a BSC stack, or TBD Hz for a HAM stack). It should be above and well clear of the pendulum, pitch and yaw modes at 0.5-0.75 Hz. There is no point in going very much above 2 Hz because the response for a given level of actuation drops as f^-2, whereas the noise floor drops more slowly. For tuning at low frequency, use 0.2 Hz.
6. In the measurement pane (see Figure 10) choose a sine response with 5 averages of 10 cycles each, and 100% settling time. The channels selected should be the four optical lever channels, which are H2:IOO-ITMX-OPTLEV1, etc.
7. In the excitation pane (see Figure 11), choose the appropriate channel for excitation (e.g., H2:LSC_MOD_EXC_1), and for readback (e.g., H0:PEM_BSC_MIC_3), a sine wave with the frequency determined in the previous step and an amplitude of 2 V for high frequency or 0.5 V for low frequency.
8. For convenience in case the session is interrupted, save this configuration as a file with a name that reflects the optic name, e.g., `OptLevITMX2K.xml`.
9. Bring up the mirror controller display, and click on the control to enable the LSC test input. Set the servo gains to a value of 5 (very low). Make sure the mirror stays lightly damped and within the optical lever range. If not, increase the gains until the minimum value for which the mirror stays in the optical lever range.
10. Click on the four icons for the actuation matrix (in the white vertical bar between the LSC, ASC inputs and the coil test inputs), and position the resulting windows on the screen as their name indicates: UR in the upper right, LR, in the lower right, etc. Set all the gains in these panels (three in each window) to 100. Minimize or close the controller display window.
11. Make sure you have all the mirror sensors, the optical lever channels, and the readback in DataViewer. Press "Start" in the GDS window. You should see the readback signal showing a sine wave that will be clipped at the maxima and minima: no need to worry about that. The optical lever channels should also show some kind of oscillation at the same frequency. Make sure the signals stay within the range of the optical lever (i.e., they don't get close to zero or -32000 counts). When they are near the end of the range, the signal flattens. If any of the optical lever signals offsets are near zero, they will always get a flattened signal: this happens when the beam is far from center, and the beam is mostly on one or two quadrants only. If that's the case, then check that the signals that do have a significant offset (and thus light on that quadrant) do not get close to zero or -32,000 counts. See Figure 12.
    When the mirror motion is moving within the optical lever range, the sensors do not show much of an oscillation - that's OK.
12. Wait till the oscillation in the readback channel is gone (~1 minute), indicating the first measurement is done.
13. In the results pane (see Figure 13) check that the coherences are reasonable (i.e., near 1.0 unless the corresponding magnitude is very small).

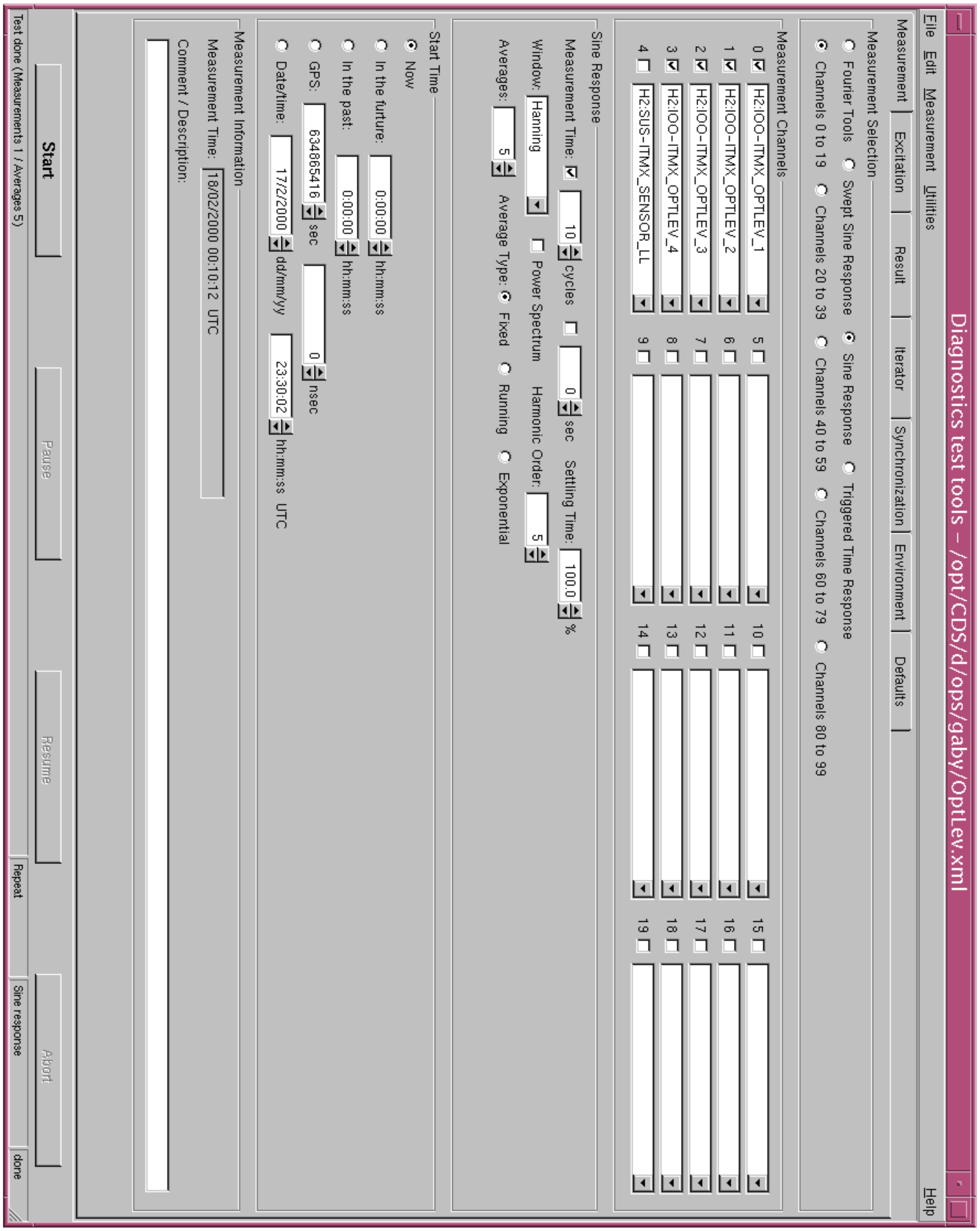

Figure 10: Typical Measurement pane settings in GDS for a measurement in the actuation gains tuning procedure.

14. In the GDS window, go the "File" menu and use "Save As" to save the results in a file called `OptLevInit.xml` in the directory `/opt/CDS/d/ops/sustune/` or TBD.
15. Repeat the measurement using different settings of the position weights in the actuation matrix and saving the results to different files as described in rows 2 through 5 of Table 1.
16. In the Matlab window, give the command `ActuationGains`. The pitch and yaw couplings for the five measured configurations are computed, together with a suggested set of five more configurations to be measured. The results are displayed on the screen and written to a text file

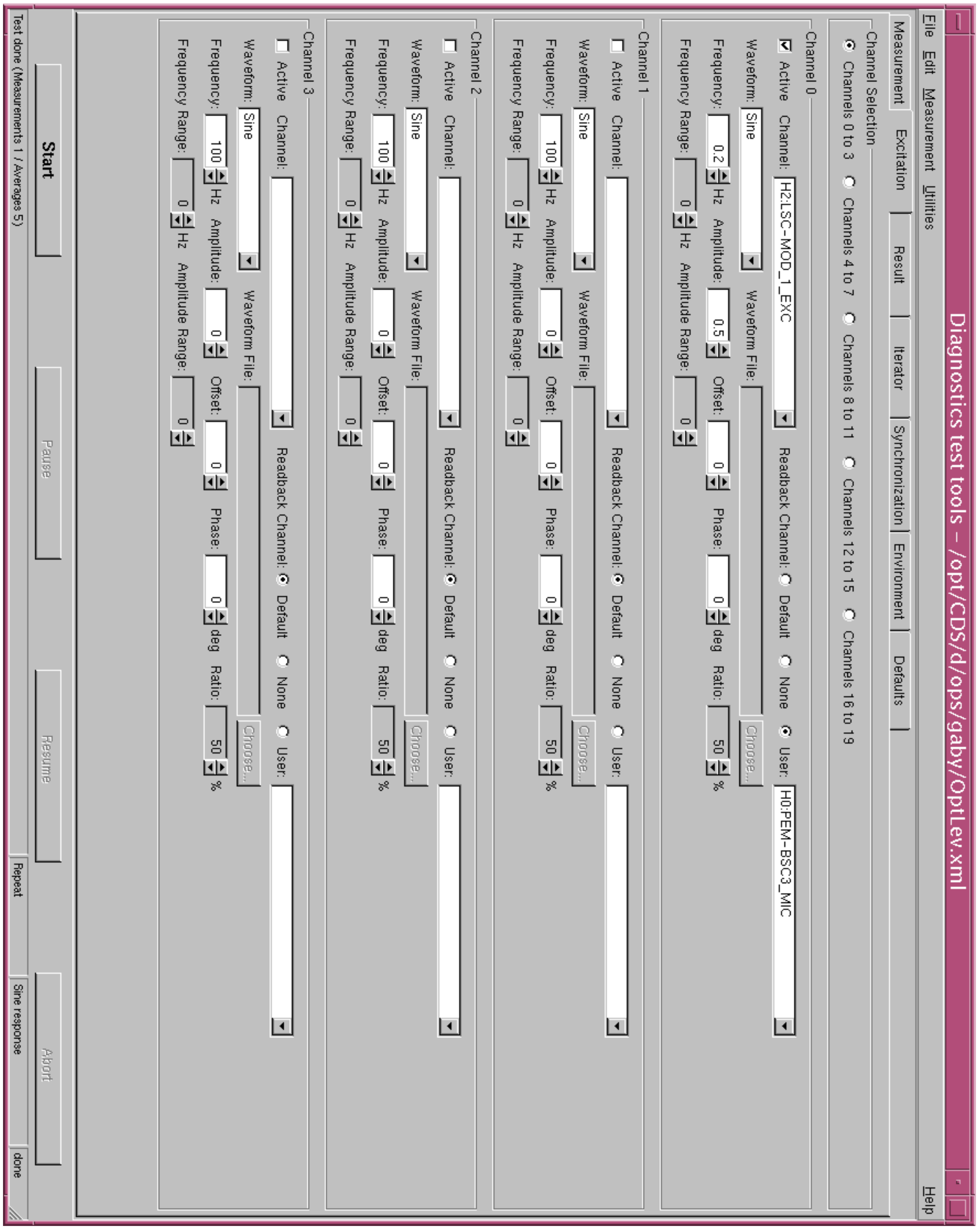

Figure 11: Typical Excitation pane settings in GDS for a measurement in the actuation gains tuning procedure.

`ActuationResults.txt`. Print the file or make a note of the information.

17. Make the first suggested measurement and save it as `OptLevInit.xml`.
18. Give the Matlab command `ReadXMLOptLev` (or TBD). (This is similar to `ActuationGains` but processes only a single measurement.) If the pitch and yaw couplings reported are better than TBD, stop.
19. Otherwise, make the remining four suggested measurements using the same procedure as above, saving the results to files `OptLevUp.xml`, `OptLevDown.xml`, `OptLev-`

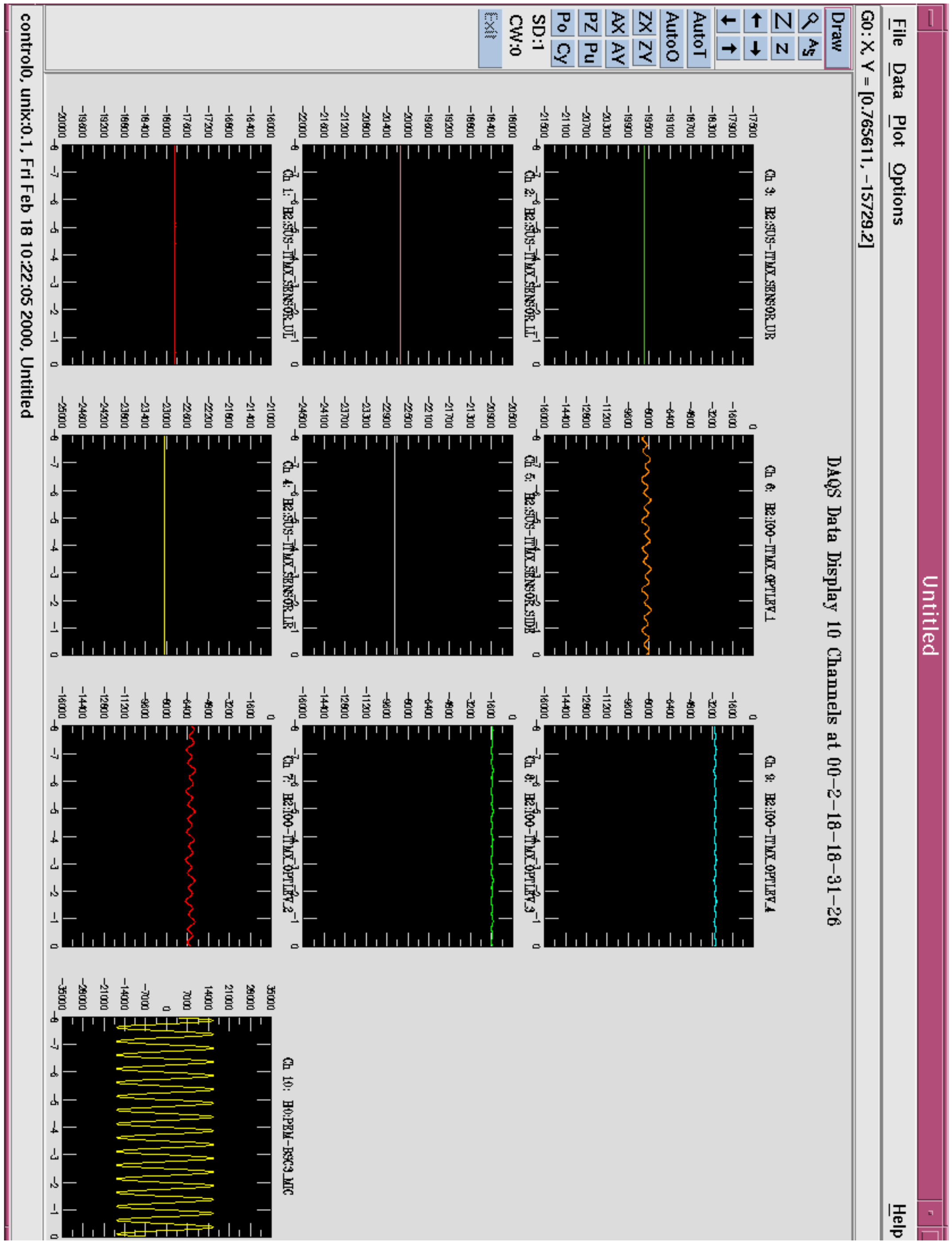

Figure 12: Typical Data Viewer display for position actuation matrix tuning, with typical (non-excessive) levels of excitation..

`Left.xml` and `OptLevRight.xml`.

20. In the Matlab window, give the command `ActuationGains2`. This does the same calculations as `ActuationGains` except that it reads the `ActuationResults.txt` file to find out the configurations rather than assuming the values in Table 1.
21. Repeat from step 16 until satisfied.

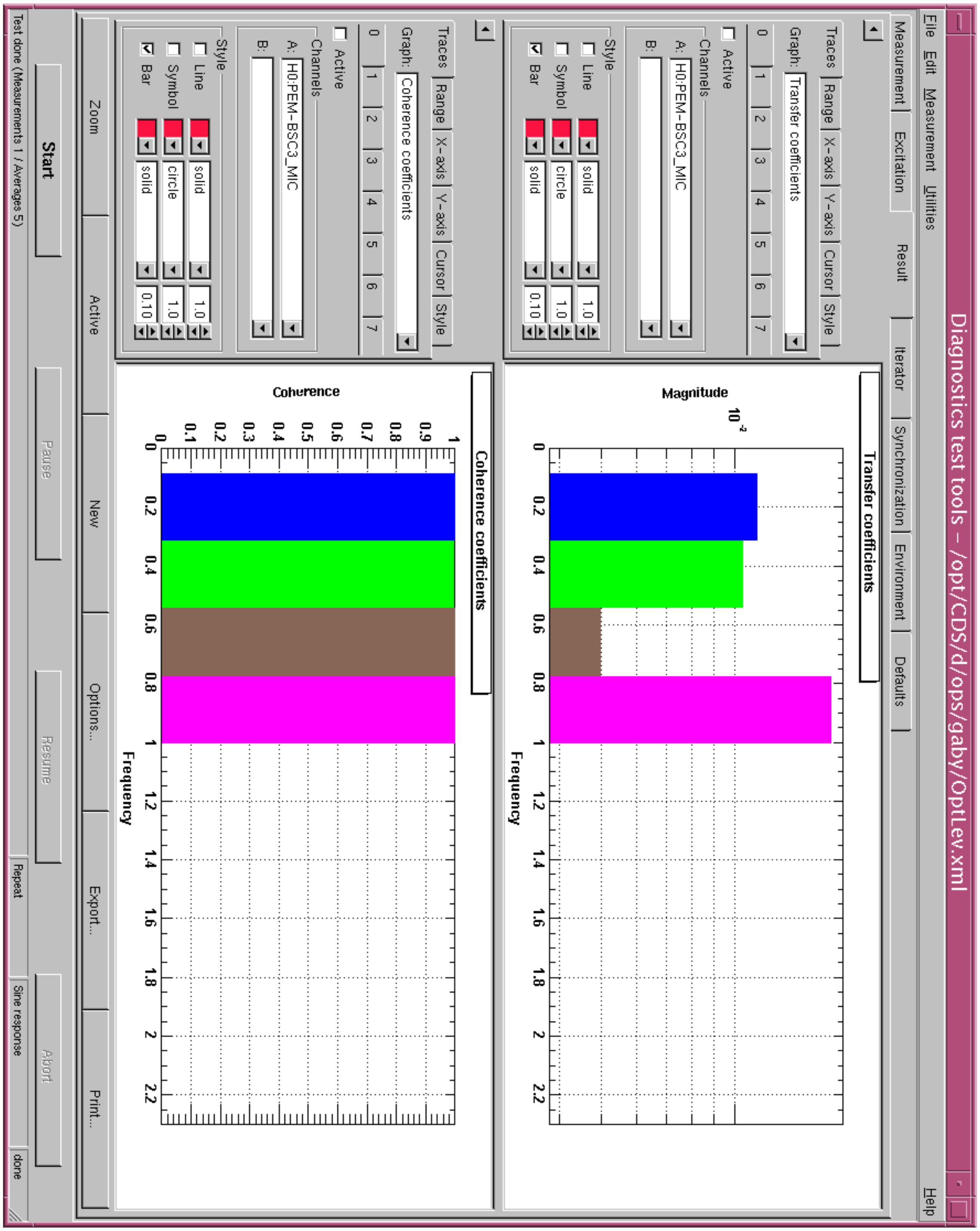

Figure 13: Typical Results pane display in GDS for a measurement in the actuation gains tuning procedure.

9.1.2. Method II , using OSEM sensors only

This method can be used on all optics.

1. TBD.

9.2. Pitch and Yaw DOFs

1. TBD.

10 DAMPING GAINS

1. TBD.