Response to HAM-ISI Watchdog Committee Recommendations
Assembled for the SEI team by Brian Lantz,
LIGO-E0900075-v1, March 9, 2009

There was recently a review of the watchdog software for the HAM-ISI for use in Enhanced LIGO. The review committee assembled a list of required actions and suggested actions in LIGO E0900006-v1, *Review of HAM-ISI "Watchdog" safety software*. We have addressed the two required actions, and several of the suggested actions. This document describes those actions.

A. Required Modifications

1. Implement forcing of default trigger thresholds

We have implemented code so that the watchdog levels are not at the nominal values (either high or low) they are returned to the nominal values at the rate of 1 count per second (approximately). The nominal values are all 20,000 counts. The return rate is really 1 count per cycle of the checker script, the uncertainty arises because the timing is set by a 'sleep' command and the run time of the script.

This functionality is accomplished by adding code to the 'checker' script. Thus, there is only one auxiliary script which requires maintenance and verification of operation. By setting the return rate to 1 count per second, (rather than, say, 3,600 counts once per hour) it is easy for the level-twiddler to be sure that the script is working, since the computer should immediately start stepping the levels back to their nominal values.

See Jeff Kissel's entry in the LLO log on Jan 25, 2009 for details of the implementation and testing. (http://ilog.ligo-la.caltech.edu/ilog/pub/ilog.cgi?group=detector&date_to_view=01/25/2009&anchor_to_scroll_to=2009:01:26:11:54:19-kissel)

This method was chosen because it was quick, convenient, simple to understand, and easy to maintain. It increases the importance of having the 'checker' script running. To that end we have done several things.
1) Implemented and tested the crontab functionality to automatically start the checker script.

2) Updated the watchdog MEDM screen with a 'heartbeat' which should blink once per second if the checker is running.
(see the picture below, and Jeff's LLO entry on Feb 2, 2009
http://ilog.ligo-la.caltech.edu/ilog/pub/ilog.cgi?group=detector&date_to_view=02/02/2009&anchor_to_scroll_to=2009:02:02:23:48:19-kissel)

3) Updated the site overview screen (at LHO) with a heartbeat with should blink once per second if the checker is running.
(see Coreys LHO entry on Feb 20, 2009,
http://ilog.ligo-wa.caltech.edu/ilog/pub/ilog.cgi?group=detector&date_to_view=02/20/2009&anchor_to_scroll_to=2009:02:20:10:43:10-corey)

4) Use the count-per-second return rate to make it easy to see that the levels are returning to the nominal value.

5) Finally, it is important to educate the operators about this system because of the 'blink-if-OK' nature of the operation. We prefer 'show-alarms-if-broken' indicators, which is the way the watchdog alarms themselves work, but we do not see a reasonable way of performing that here.

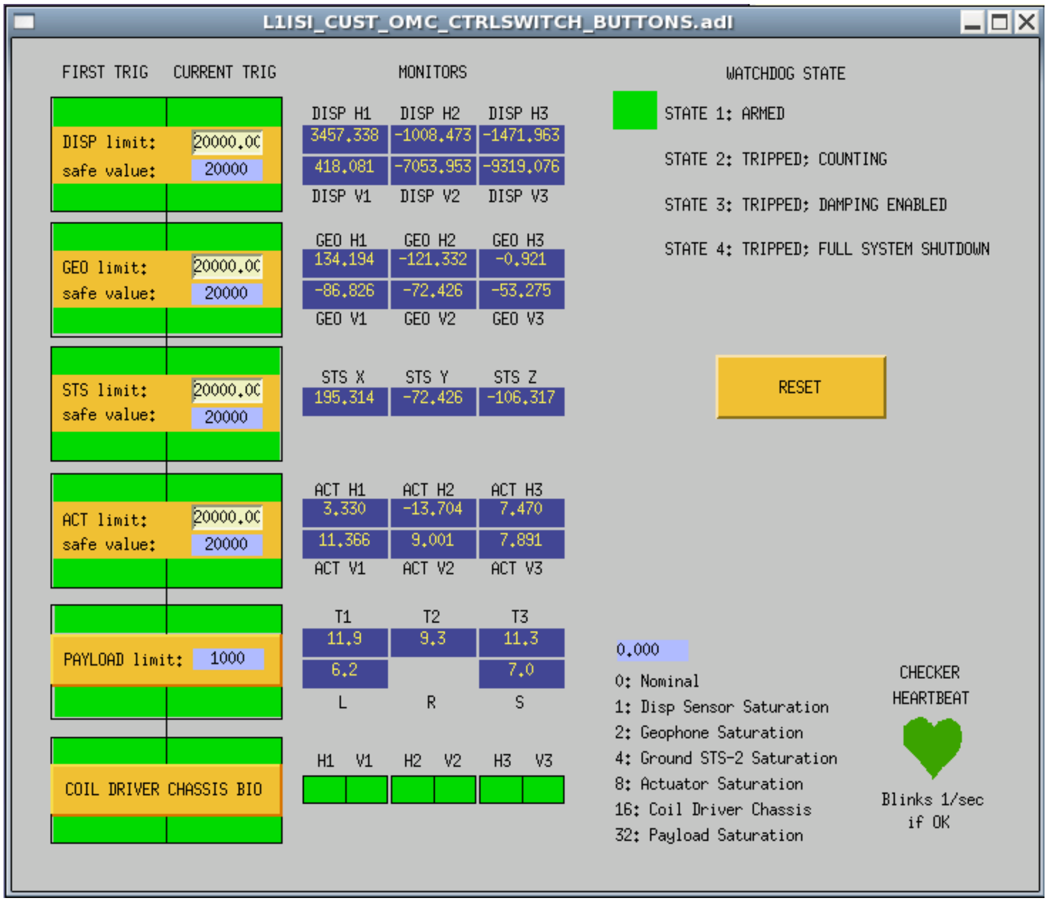

Figure 1. Watchdog screen. The checker heartbeat is in the lower right, and blinks between dark grey and green 1/sec if the checker script is running. The watchdog now watches 6 sets of inputs: the feedback displacement sensors, the feedback geophones, the floor-mounted STS-2 used for sensor correction, the actuator drive signals, the OMC watchdog, and the over-temperature monitor for the ISI coil drivers.

2. Ensure watchdog fault conditions are detected by the alarm handler

Two things have been done to help ensure that a tripped watchdog will be seen by the operator. First, the ISI watchdog status is made available to the operator via a new indicator on the Hanford site overview screen, as shown below in figure 2.

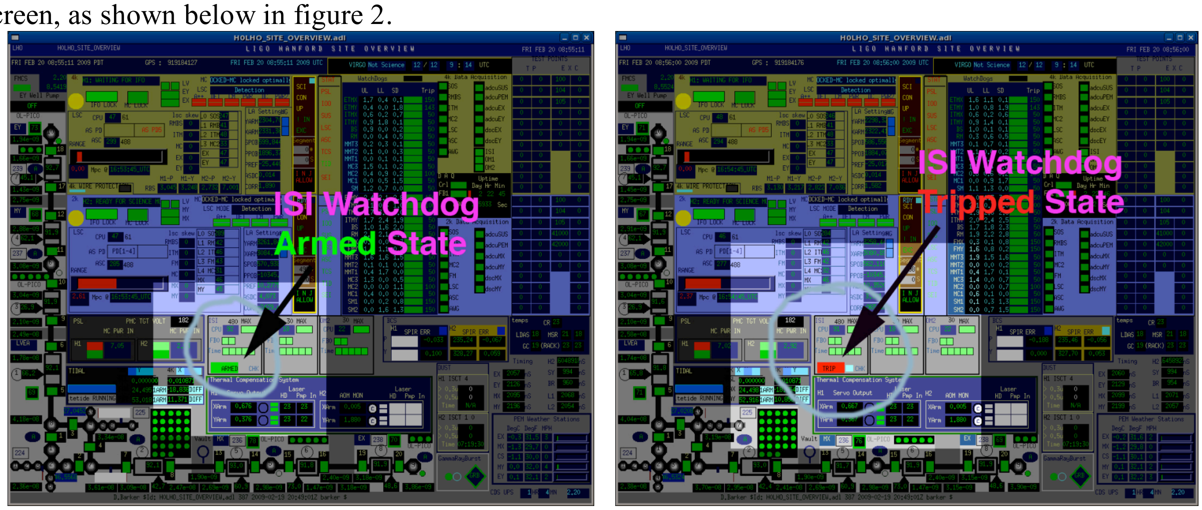

Figure 2. LHO site overview screen, showing the HAM-ISI watchdog status.

These screen shots were taken from Corey LHO elog entry on Feb 20, 2009, at
http://ilog.ligo-wa.caltech.edu/ilog/pub/ilog.cgi?group=detector&date_to_view=02/20/2009&anchor_to_scroll_to=2009:02:20:10:43:10-corey

Second, the watchdog level has also been added to the LHO alarm handler, but this is not yet a reliable approach. During commissioning, this alarm is disabled, because it tends to go off enough to be annoying, and we don't want people to learn to ignore it. Also, with the Advanced LIGO control design system (Borkspace), rebuilding the ISI model deletes the necessary alarm information from the epics database. We have manually updated the database, but this is not very stable since we have to remember to manually edit the epics database to re-add the alarm whenever we do a model build.

This is ameliorated somewhat by the fact that the ISI model is now pretty stable, so we don't plan to do many more builds. This issue is in the queue for CDS to fix (bug 212 in bugzilla).

B. RECOMMENDATIONS AND CLARIFICATIONS

3. Justify complication caused by providing intermediate "damping only" fault states

In our experience, in the case of faults, it is usually best to leave the damping loops on. This is especially true once the system is commissioned, and the major source of watchdog triggers is either large environmental disturbances or drive commands. In these cases, a controlled shutdown is safer than simply disabling all the drivers, because if there are any offsets being maintained by the controllers, or there are large external drives, the damping system allows the table to return smoothly to its neutral position.

An example of the impact of the damping loops on the response of the system to a step drive is shown below in figure 3. This plot is drawn from an entry in the LHO log by Jeff Kissel on June 4, 2008.

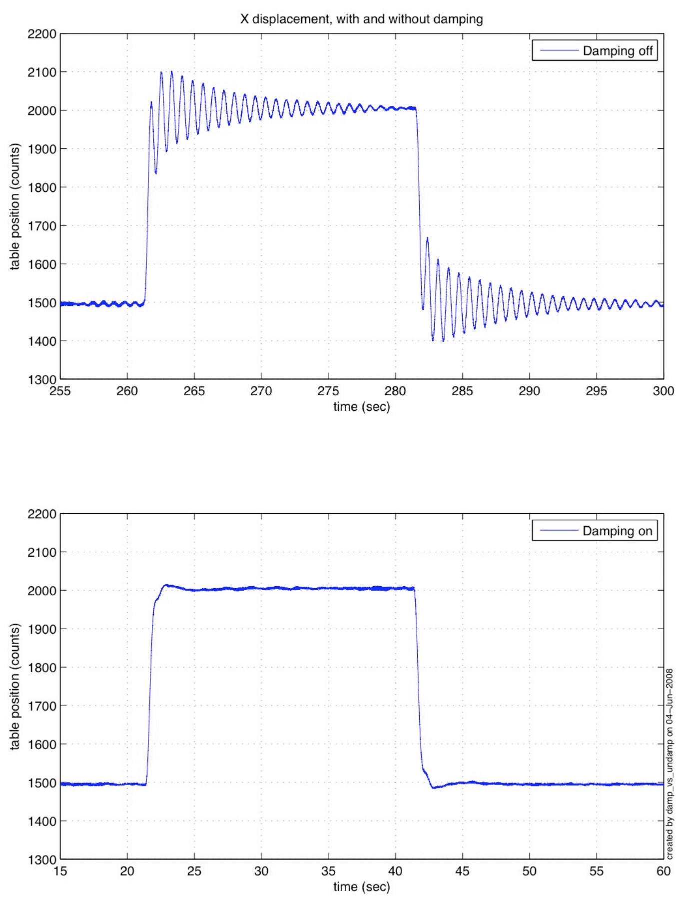

Figure 3. System response to a step function, with and without damping. In each case, a 1000 count drive was sent to the X-coordinate-direction actuator path for about 20 seconds. The plot shows the displacement motion of the table as measured by the x-coordinate direction displacement sensors. In the upper trace, the damping loops are off, and in the lower trace, the damping loops are on (the isolation loops are off in both cases). Clearly the damping loops reduce the peak accelerations and the rms acceleration of the table after the step.

The step example is quite important, since that is similar to the type of event one would expect when the watchdog is triggered, because if there is any bias being held by the isolation loops (to align the OMC, for example), then that bias command will be immediately turned off by the watchdog. There is some lowpass filtering of the signal in the coil drivers (which is present for these measurements), but the damping loops clearly reduce the amount of shaking on the table when a command bias is introduced or removed.

The damping loops are quite robust. They can be turned on and off in any combination in any order, which has been demonstrated repeatedly. A failed sensor is practically equivalent to turning off the input to the control filter, which is how we typically turn on and off individual damping loops. They damping loops operate even when the sensors are driven into saturation, as can be seen in figure 4.

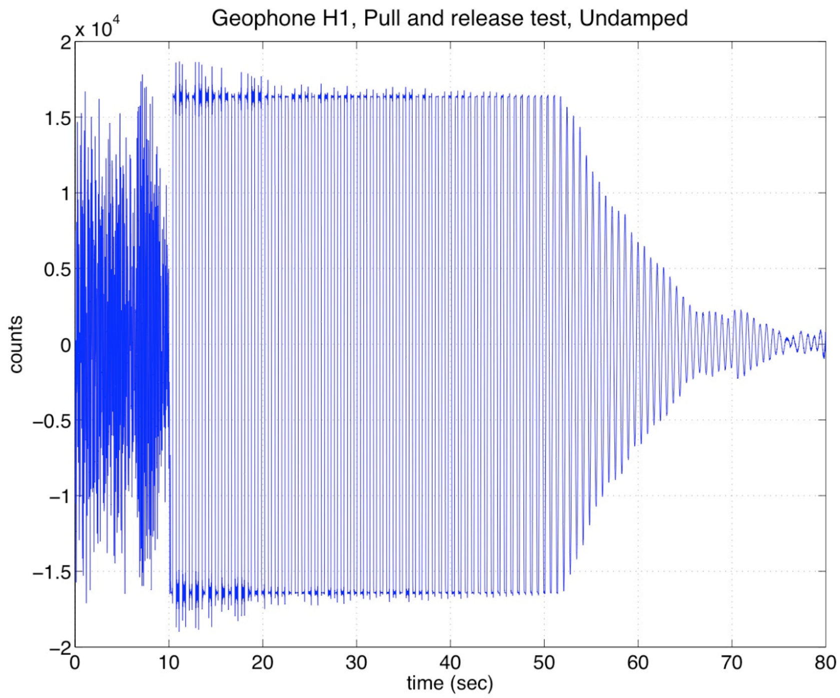

Figure 4a. Undamped response of the ETF table, as seen by geophone H1, when the table is pulled sideways to one stop and released (released at about 10 sec on this plot). The sensor goes in and out of saturation for about 40 seconds, then continues to ring down for another 25 seconds or so.

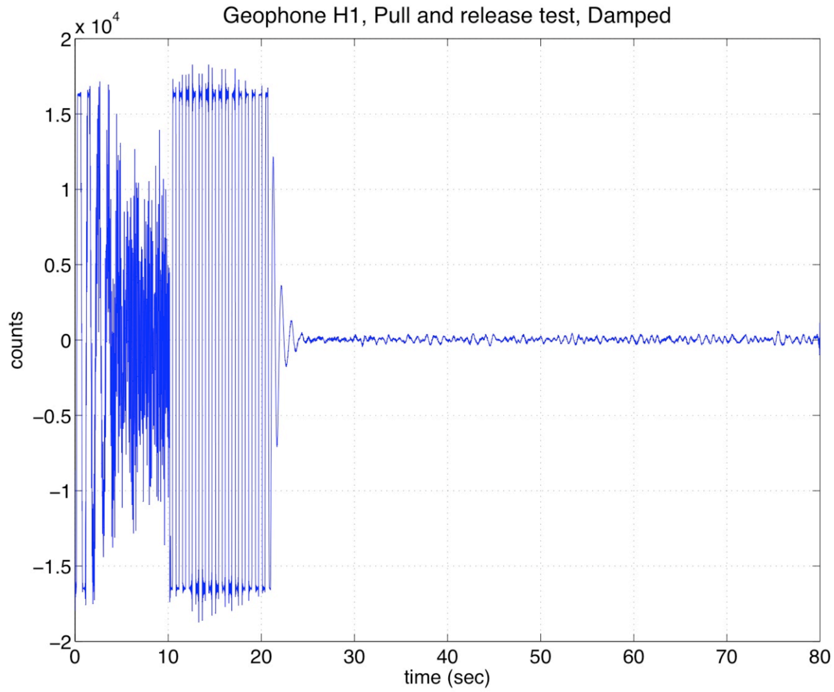

Figure 4b. Damped response of the table to the same hold-and-release described in 4a (above). The H1 geophone moves in and out of saturation for just over 10 seconds. The damping loops are active, and even though the sensor spends much of the time saturated, the damping loops have reduced the saturated time by roughly a factor of 4. Once the saturations end, the system is back to ambient noise in about four seconds.

By comparing the time traces in figures 4a and 4b, we see that even with sensor saturations, the damping loops perform very well. This experiment is described in more detail in the ETF log at http://ligo.phys.lsu.edu:8080/ETFseismic/1077.

The damping loops do not have any DC gain, and the loop gain is greater than 1 only at the main body modes of the table around 1 Hz, so turning them on (and bringing them out of saturation) does not seem to cause any strange behavior (often seen with integrator windup). In addition, since the gain is rolled off above 1 Hz, we have never experienced a payload change which affected the stability.

Of course, there are fault conditions which will make the damping loops unstable. Increasing the gain by more than a factor of about 3 will cause oscillation, as will changing the sign of the gain, or neglecting the turn on the filter bank. At the Stanford ETF, we have (accidentally) demonstrated that plugging the V2 control signal into actuator V3 and vice-versa will also cause instability.

In general, the damping loops remove energy from system and reduce velocities. Since shutdown of isolation loops can be a step function in the command signal, damping reduces the glitches associated with isolation loop shut down. Most of the shutdowns are associated with overdriving the transfer functions for plant measurement or for other various drive commands (step location of ISI table for testing DARM pickup to OMC motions) in this case it is clearly better to have damping loops on. At Stanford we leave loops on while re-balancing table, since it makes things behave better. (At the observatories, the tables will be loaded up while the lockers are engaged, but re-balancing might be done with damping loops running, even with a payload in place.)

On balance, it has been our experience that the system is safer with damping loops which will run even when the watchdog shuts off the isolation loops. The 3 second hold time seems a good compromise between calming the platform down after the shut-downs (e.g. excessive drives by over-enthusiastic commissioning teams), and preventing larger, but less-likely problems (e.g. computer freak-outs, cable swaps) from damaging the system.

4. Verify stability of "damping only" loops under correlated sensor loss
See discussion above, this is tested just about every time we turn the damping loops on and off.

5. Consider adding "dead channel" test
We are currently developing and off-line test of the sensors and actuators. An online test like this would be useful, but it may be a while before we get to it.

6. Reconsider execution platform for auxiliary scripts
At LLO:
The LLO log entry on Feb 26, 2009 by Keith Thorne says:

Keith, Jeff
We have moved the ISI checker from 'borkspace' to 'lloscript'
All the code is in /cvs/cds/llo/scripts/l1/ISI

Using crontab, a 'chk_daemon' is run every second
So the following was added to the 'controls' crontab

* * * * * /cvs/cds/llo/scripts/l1/ISI/chk_daemon

This process will restart the 'checker' script if it is not running.
There is a 'checker_status.log' log file

http://ilog.ligo-la.caltech.edu/ilog/pub/ilog.cgi?group=detector&date_to_view=02/26/2009&anchor_to_scroll_to=2009:02:26:10:08:21-keitht

At LHO:
The LHO entry on Feb 6, 2009 by Dave Barker says:

The ISI watchdog checker script
(/cvs/cds/lho/scripts/h1/ISI/checker) is
running on control14 as user controls.
controls is also running a cronjob
every minute which will restart checker
if it dies.
This function will be moved to the h1
linux scripts machine when that is running.

http://ilog.ligo-wa.caltech.edu/ilog/pub/ilog.cgi?group=detector&date_to_view=02/06/2009&anchor_to_scroll_to=2009:02:06:16:16:16-barker

7. Develop a more robust way to insure that checker script is operating
see discussion in the 'Required Modifications'

8. Track watchdog variables in the conlog (Kissel/Thorne/Barker)

9. Detect and trigger on STS-2 saturations as well as ISI sensors
This has been implemented (see screens above).

10. Describe prior fault incident(s) (e.g., 12 November 2008) and show how comparable faults are now precluded.

The fault on November 12 was caused by a poor implementation of the reset logic for the watchdog. To reset the watchdog, a level must be set high. That signal level should then be set back to low, otherwise the watchdog is being continually reset, and does not perform as a watchdog. The fault in November was caused because the reset command was not set back to the run state.

This has been fixed by creating a new part called 'cdsEpicsMomentaryBinIn'. In addition, the HAMISIWATCHDOG.C code now reads the reset trigger, and then sets it back to run mode every clock cycle with the following code lines:

```
// Check and record the RESET button value
  int resetFlag =  pLocalEpics->isi.OMC_RSET; // Epics variable for reset button
  pLocalEpics->isi.OMC_RSET = 0;              // Set reset button epics variable to 0
```

Rolf points out that something more general ought to be implemented for Advanced LIGO, so that the cds part will reset itself, and not require hardcoded names in c-files.