Increasing the Damping Time for the ISI Watchdogs

Brian Lantz

T1700481-v3, Nov. 6, 2017

1 Summary

This documents changes to the ISI watchdog so that the system will be held in the 'damping' state more robustly during events which trip the watchdog. Operational experience, punctuated by several large earthquakes during O2, has led us to the consensus view that most of the time the seismic and suspension system is safer when the ISIs are damped rather than in full shutdown. As such, we are changing several parameters of the ISI watchdog to keep the system in the damping mode longer and make it simpler to return from full shutdown to damping. This technical note supplements ECR E1700367 and FRS ticket 9309. It is a part of the general O2-O3 updates.

Figure 1: Detail of the updated watchdog screen - the system is damping, the countdown timer is running, and a reset is pending.

2 List of changes

Several changes have been made. Recall that each stage is running its own watchdog code. The reset signal is sent to both, and one of the triggers for the stage 2 watchdog is the state of the stage 1 watchdog. The watchdog states are:

state 1: Armed - system is running normally and the watchdogs are monitoring for faults and saturations. On a saturation, trip to state 2.

state 2: Damping hold - isolation loops are turned off, but damping is left running. System is held here, ignoring saturations, until the hold timer runs out or the reset is pressed. On reset, return to state 1. If the hold timer runs out, proceed to state 3.

state 3: Damping only - Damping loops continue to run. Isolation loops are off. State 3 is very similar to State 2, except that the saturations are monitored again. System stays in state 3 until reset is pressed returning to state 1, or there are more saturations, taking the system to state 4.

state 4: Full shutdown - Damping loop outputs are blocked. Isolation loops are off. System stays in state 4 until the reset is pressed.

These states have not changed, but the details of the state progression have been updated. The updates are shown below.

Parameter	new 'LongDamp' mode	prior to Nov 2017
Hold time in state 2	60 seconds	3 seconds
Reset enabled in state 2	yes - 2 clicks, see discussion	no
Reset enabled in state 3	yes	yes
Reset enabled in state 4	always	only when there are no saturations
Delay before stage 1 trips stage 2	adjustable from 1 to 20 seconds	immediate
Final state for stage 2 if stage 1 trips	State 3 - damped	State 4 - Full shutdown
T240s monitored in states 2, 3 & 4	no	no

Table 1: Summary of changes to the watchdog

3 Installation

3.1 SVN updates

This update requires an SVN update and a rebuild of the models. No top level model changes are required.

There are 3 files in various folders of the seismic userapps which have been updated for the BSC. The BSC-ISI updates are in rev-16398. You need to SVN up the following files:

```
{userapps}/release/isi/common/medm/
  bscisi/ISI_CUST_CHAMBER_WATCHDOG.adl       / update to the watchdog screens

{userapps}/release/isi/common/models/
  isi2stagemaster.mdl    / both master models have been changed

{userapps}/release/isi/common/src/
   ISIWD_LONGDAMP.c   / new c-code.
```

There are 2 more files for the HAM-ISI update. These were committed on Nov 6, 2017. The HAM-ISI updates are in rev-16422. The HAM and the BSC source the same watchdog c-code.

```
{userapps}/release/isi/common/medm/
  hamisi/ISI_CUST_CHAMBER_WATCHDOG.adl      /  update to the watchdog screens

{userapps}/release/isi/common/models/
  isihammaster.mdl
```

3.2 BSC-ISI and HAM-ISI model updates

Each BSC-ISI model and each HAM-ISI model needs to be rebuilt. Please check that the code 'userapps/release/isi/common/src/ISIWD_LONGDAMP.c is sourced in the make process instead of the userapps/release/isi/common/src/ISIWD_GPS.c code.

3.2.1 MEDM updates

The BSC-ISI and HAM-ISI watchdog screens has been updated. They should just work when you reopen them. Please see the discussion section below on the new features.

3.2.2 Guardian Changes

I don't think any changes are required for the Guardian - but I have not tested this.

3.2.3 BSC-SDF and safe.snap

Once the models are rebuilt and restarted, remember to update the state definition files. There are a few new epics variables, one in particular needs to be set and monitored by SDF - *ifo*:ISI-*chamber*_ST1_WD_ST1_FLAG_GEN_DELAY_TIME is time delay between when stage 1 is tripped and the WD automatically trips stage 2. This number is set on the watchdog screen. It is not clear what the optimal value is - and this should probably be determined experimentally by tripping the WD a few times with various delays and monitoring the motion of the suspended mirror. This delay is forced to be between 1 and 20 seconds. My guess is 5 seconds is a good place to start, since it is long enough for the SUS to ring down a bit, but short enough that the stage 2 translations loops (which are likely unstable now) shouldn't have enough time to ring up too much.

4 Discussion

4.1 Design Notes - timing

This update was precipitated by the 2 week down time after the Montana earthquake on July 5, 2017 (local time) and by the manual resetting of various parameters (see Livingston alog 35846) during the M8.2 earthquake off the coast of Mexico on Sept. 7, 2017 (local time). We have all long believed that, if the software and hardware are working correctly, holding the ISI in the 'damped' state is safer for the ISI and SUS hardware than letting the platforms swing freely in the 'Full Shutdown' state. The CDS software and the ISI user software now seem to be pretty stable, so now it seems that the earthquakes are a more likely source of trouble. Hence the commissioning gap between O2 and O3 presents a opportune moment to relax the ISI user watchdog constraints.

A number of parameter changes have been made. First and most obvious, the time in state 2 'Damp-hold' has been increased from 3 to 60 seconds. This parameter is hard-coded in the watchdog c-code. We chose 60 seconds based on the ring-down history after the Mexico earthquake. The time history of the various sensors after that event is shown in figure 2 from Hanford alog 38985. The event started just before time 120 in that figure, and after 3 seconds the damping loops were turned off. Even with the damping loops off, by time 180 the sensors were mostly out of saturation and the platform is ringing down based on the small natural damping of the coil driver back impedance. Thus, the large drive events from the ground seem to have stopped, and if the damping loops were still on they likely would have remained running for some time. Note that the T240s do saturate at t=210 (about 90 seconds after the first trip) and several times thereafter. Once the isolation loop gain is set to 0 by the watchdog, the T240 signals are no longer considered by the watchdog, so, even if this type of saturation persists, they will not trip the watchdog from state 3 (damping) into a full shutdown. Thus, we feel that 60 seconds of damp-hold time should be sufficient to keep the system at 'damped' even during most large earthquakes. 30 seconds seemed too short- not allowing the earth to settle down, and 120 seconds seemed too permissive - adding risk that the system would allow run in the presence of faulty hardware or control loops.

We also note that, if the watchdog is reset during the 60 seconds of 'damp-hold' time, either by a quick operator or by a future software system, then a subsequent trip will put the system into the damp-hold state and the system can be held out of the 'full-trip' state. Jim Warner has implemented a single button on a top level-level MEDM operator screen for just this case (see

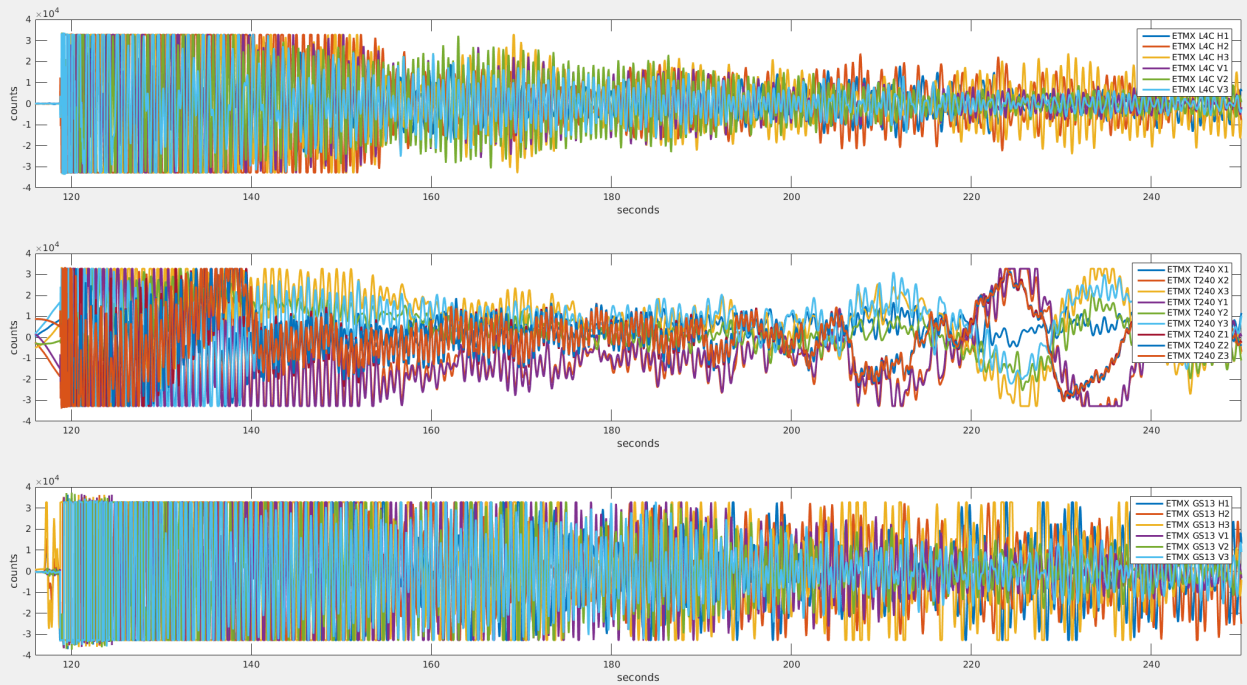

Figure 2: Signal in various sensors for the LHO ETMX chamber for about 120 seconds after the arrival of the ground shake from the M8.1 earthquake off the coast of Mexico. Even with the damping off, the signals on the L-4Cs (top trace) and the GS-13s (bottom trace) are back to within the saturation limit after about 60 sec (tick 180 on the graph). The T240s (middle trace) show continuing saturations. One the isolation loop gains are set to 0 by the watchdog (immediately in state 2) the T240 signals are not considered. This shows that 60 seconds should be adequate to keep the ISI system damping running, even during most large earthquakes. Figure from Hanford alog 38985 by J. Warner.

Hanford alog 38921), the seismon system is being tested to warn against distant earthquakes, and discussions are underway to detect and react to large, local earthquakes.

4.2 Design Notes - Reset

Since the 'damp-hold' time is now much longer, it is possible to take actions while the system is in this state, and so the 'Reset' has been enabled during state 2. Since the point of state 2 is to ignore continuing saturations we have developed a 'two-press' reset action for this state. The first press of the reset button leaves the system in state 2, but marks a 'reset-pending' request so that a reset to normal function of the ISI will be activated once the 60 second ring-down time has elapsed. A second press of the reset button will activate an immediate reset, which is particularly useful during testing.

It is very important to remember that a watchdog reset will unblock the damping loops (if the system is in state 4) and it will allow the Guardian to take the system to the requested state, but the watchdog reset Does NOT turn the isolation loops back on. If the requested state is 'isolated' then the guardian will automatically attempt to re-isolate the system once the watchdog is reset.

Another change to the resets is that now the system can be reset from full shutdown even if there are on-going saturations. Effectively, this will force the system to damp for another 60 seconds,

since a reset will push the system to state 1, and the on-going saturations will automatically trip the watchdog again into state 2, 'damp-hold'. When the source of saturations is large ground events, all of our experience shows this is the correct course of action, and so allowing the operators to do this seems wise.

The Watchdog MEDM screen has been updated to show the timing and reset status during the stage 2 'damp-hold' time. Figure 3 shows a detail of the BSC-ISI watchdog screen when the system is counting in state 2 and a reset has been requested. The countdown timers are different because this system has a 5 second delay before the stage 1 watchdog trips the stage 2 watchdog.

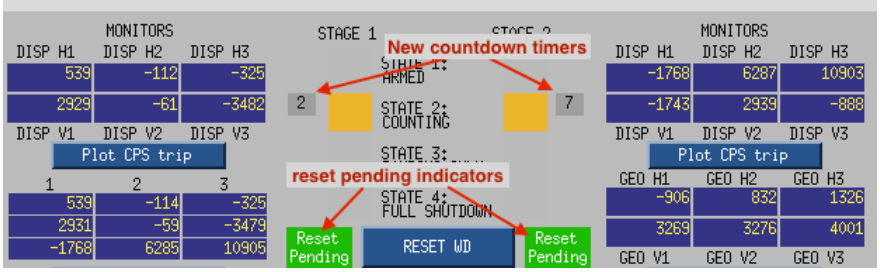

Figure 3: Detail of the updated watchdog screen highlighting the countdown timers for the two stages, and the pending reset indicators

4.3 Design Notes - Delayed trip of stage 2

The horizontal isolation loops on stage 2, and perhaps some of the other DOFs as well, are not stable when the rX and rY loops on stage 1 are off. Thus, we have set up the 2 BSC-ISI watchdogs so that when the stage 1 isolation loops are turned off, the stage 2 isolation loops are automatically turned off as well. This update improves that connection in two ways. First, the stage 2 watchdog is only triggered once when the stage 1 watchdog trips, so that stage 2 will stay damped, instead of being forced to 'Full-trip'. The stage 2 watchdog is still active, so other stage 2 saturations will still trip the stage 2 watchdog, as they should. Thus, stage 2 is still protected, but it will stay damped unless there is something seriously wrong.

The other change is that, since there can be large impulses when the isolation loops are shut down, there is a short delay between the trip of the stage 1 watchdog and the automatic trip of the stage 2 watchdog. It is not clear what the optimal delay is, so this can be set by the user in the range of 1 - 20 seconds. The simulink code for this is included below. The variable which holds the time delay is *ifo*:ISI-*chamber*_ST1_WD_ST1_FLAG_GEN_DELAY_TIME and can be seen on the Watchdog screen. The max and min values are hard coded into the simulink diagram.

Most of the code changes for this update are related to the stage 1 - stage 2 watchdog trips. These are included below in the code section.

5 MEDM screens

The MEDM screen for the watchdog has been updated slightly. We have added a countdown timer which is visible in state 2, a flag indicating that there is a reset pending, and a new input for the time delay for stage 1 tripping stage 2. The updated BSC-ISI watchdog screen can be seen in figures 4 - 6. The HAM Watchdog screen has also been updated, and a view of this can seen below in figure 7.

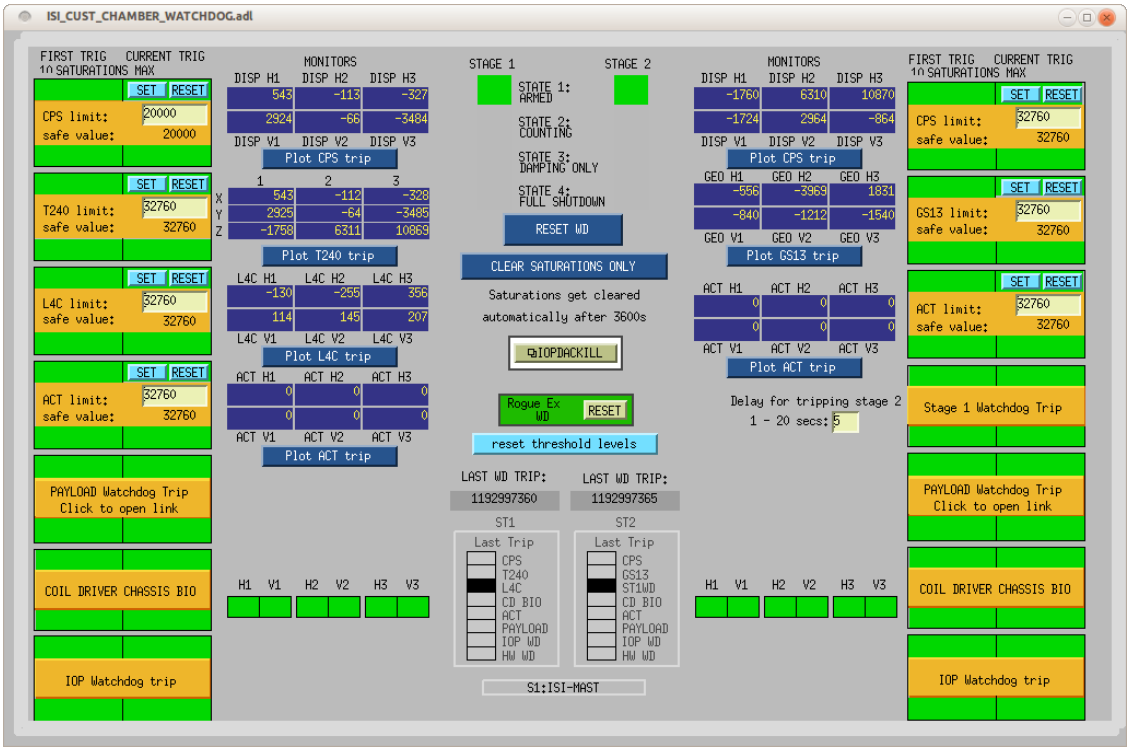

Figure 4: BSC-ISI watchdog when all is well. Note the new delay time input by the stage 2 indicator of a stage 1 trip.

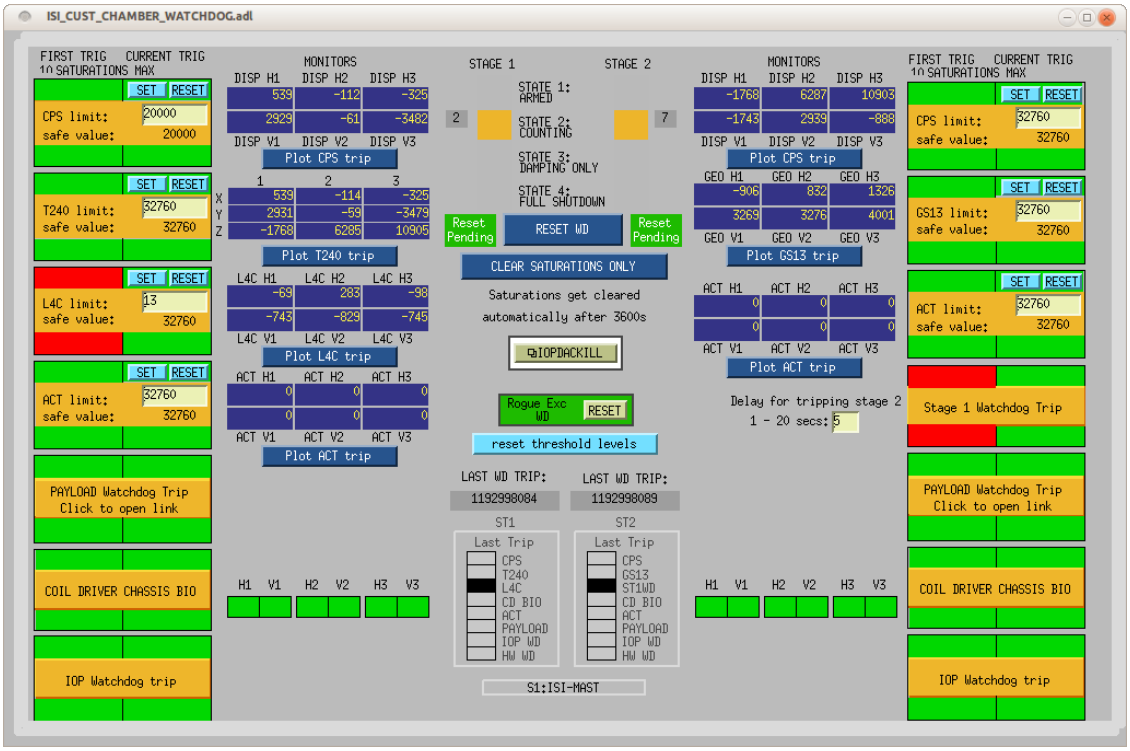

Figure 5: BSC-ISI watchdog several seconds after a stage one trip. There is a reset pending.

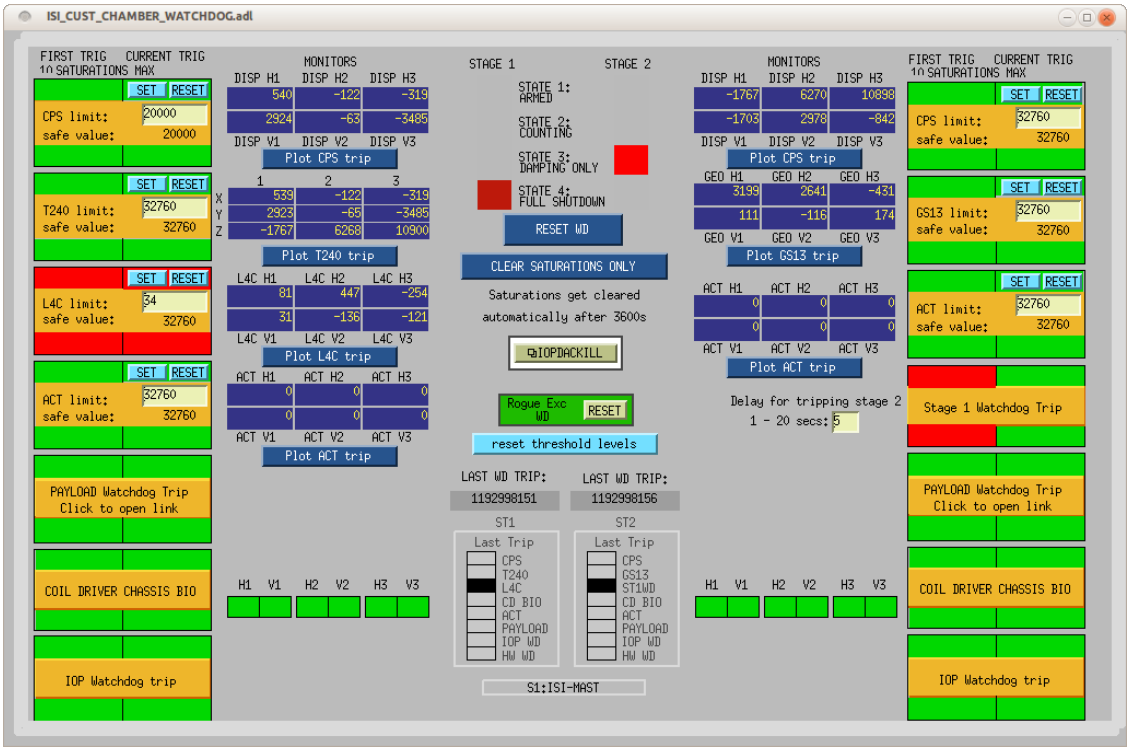

Figure 6: BSC-ISI watchdog after stage 1 has gone to full shutdown. Note that stage 2 is still damped

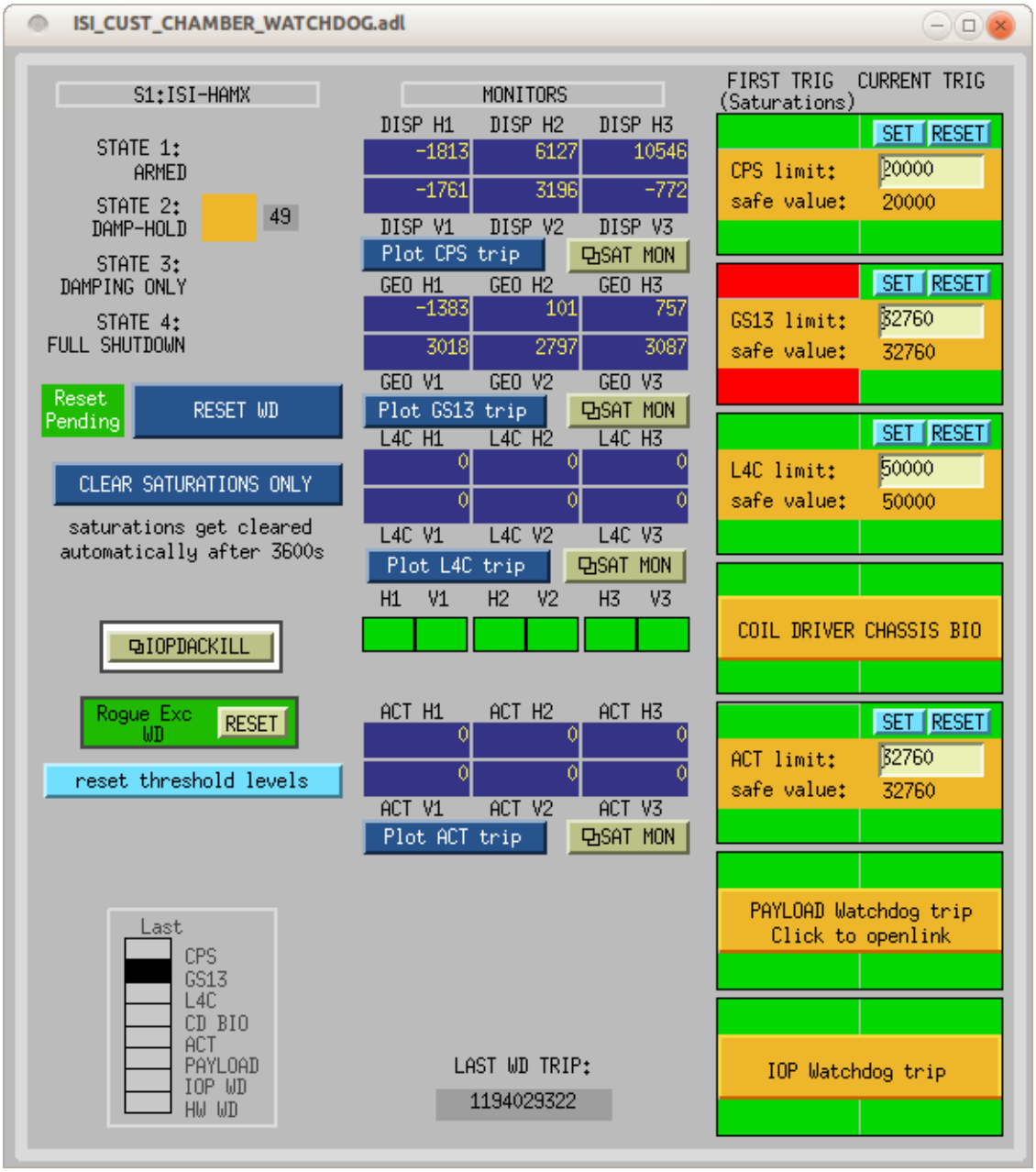

Figure 7: HAM-ISI watchdog screen in state 2 with a reset pending. In addition to the new countdown and reset-pending features, I've made minor cosmetic changes.

6 Simulink diagrams

The new watchdog c-code has two new outputs - a monitor for the time remaining in state 2 and a monitor for the watchdog pending flag. Because the number of outputs has changed, I have changed the name of the c-code file. These changes can be seen in the watchdog for block for BSC-ISI stage 1, shown below in figure 8. The only changes for the the HAM-ISI watchdog and for the watchdog on Stage 2 of the BSC-ISI are the new c-code and a expanded monitor block. There is a good deal of new code in the new ST1_FLAG_GEN code block. This path used to just send a steady '1' to the stage 2 watchdog when the stage 1 watchdog was in any of the tripped states. Now, it will send a short pulse of '1' to stage 2 after a few seconds of continuous stage 1 WD trip. This pulse will trip stage 2, but then it will return to a '0' level, so stage 2 will only proceed to 'Full-trip' if a fresh trigger arrives, e.g. from a stage 2 saturation. The code for this flag generator block is shown in figure 9. The delay time is held in a single epics variable called *ifo*:ISI-*chamber*_ST1_WD_ST1_FLAG_GEN_DELAY_TIME. If the user specifies a delay time outside of the min-to-max range specified, the simulink code will reset the value of this epics variable to the in-range value on the next clock cycle. I think this is new functionality; the code is shown in figure 10.

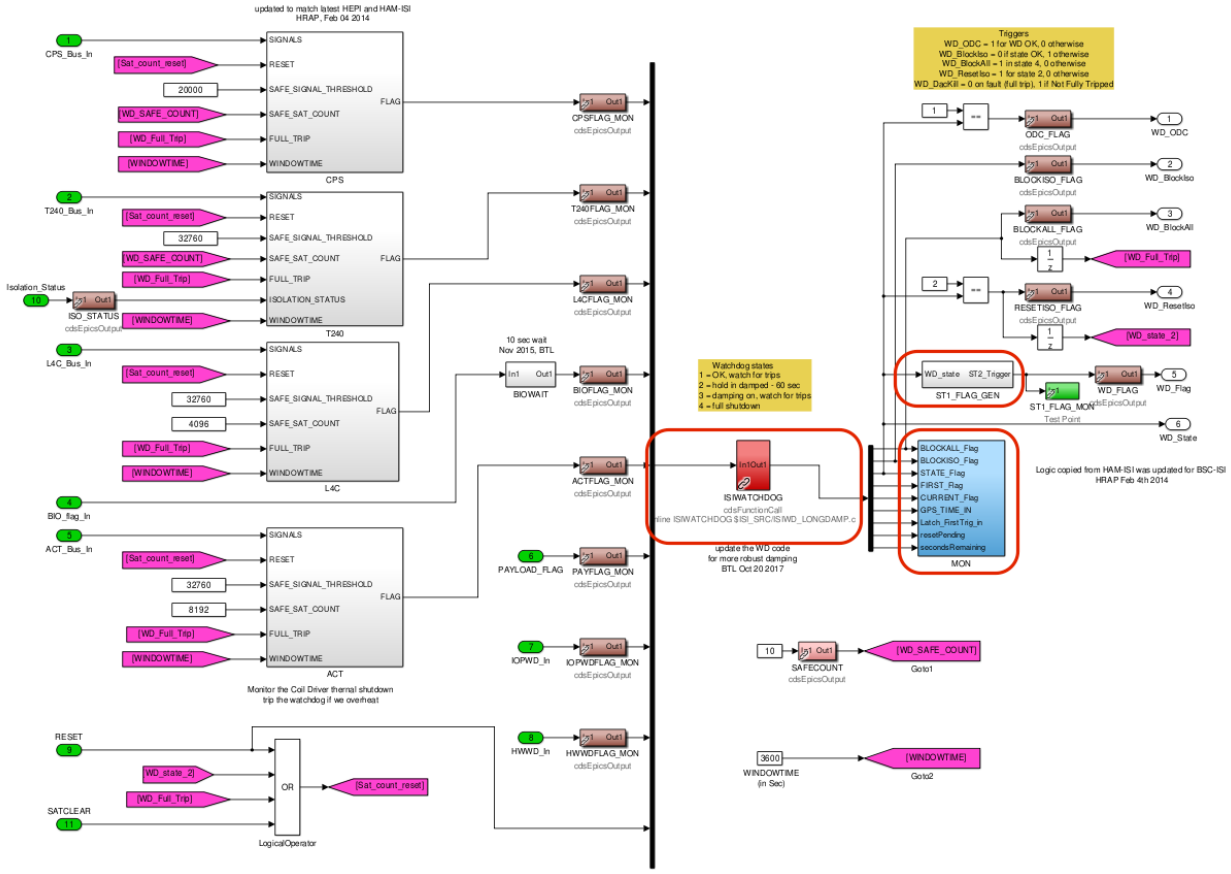

Figure 8: Watchdog block from stage 1 of the BSC-ISI. Highlights show the new c-code, the updated monitor block, and the ST1_FLAG_GEN block which calculates the trip signal to send to the stage 2 watchdog. The HAM-ISI watchdog and the stage 2 BSC-ISI watchdog block updates are very similar, those get the new c-code and the new monitor blocks.

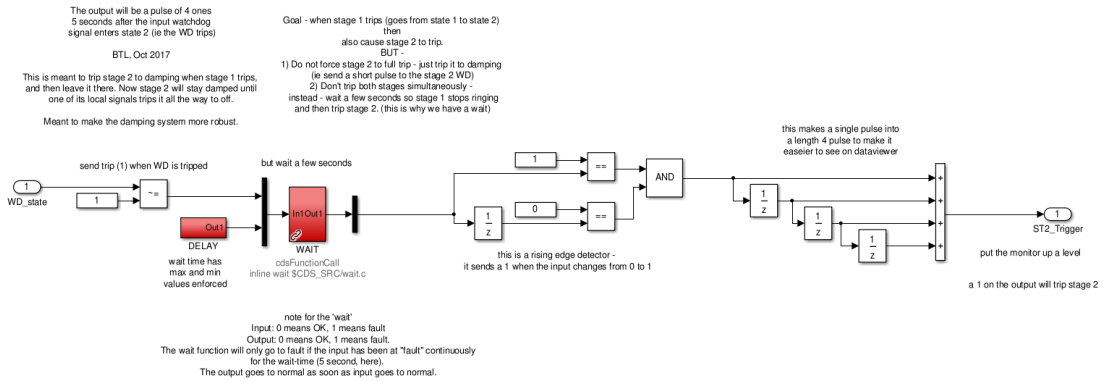

Figure 9: ST1_FLAG_GEN Simulink block. This generates a pulse when the stage 1 WD has been tripped for the specified DELAY time.

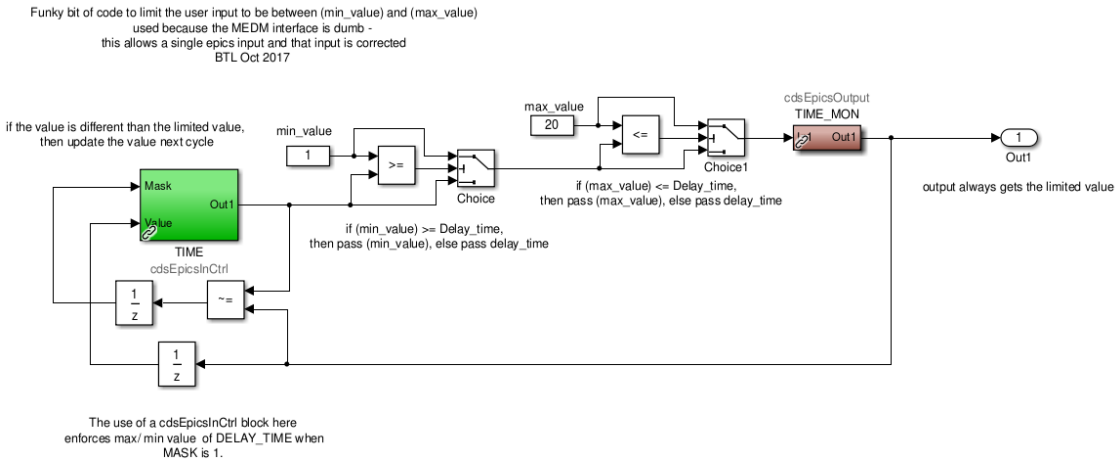

Figure 10: ST1_FLAG_GEN_DELAY Simulink block. This enforces the 1-20 second limit on the stage 2 trip delay.

7 Source Code

There is a new piece of c-code called **ISIWD-LONGDAMP.c**. It lives in userapps/trunk/isi/-common/src/. The code is:

```
/*
 * ISIWD_LONGDAMP.c
 * adapted fom ISIWD_GPS.c by BTL on Oct 19, 2017 -
 * see ECR E1700367 and note T1700481 - there are 4 changes
 *    1 - increase time in state 2, triggered-holding from 3 seconds to 60 seconds
 *    2 - set 2 resets in state 2, (resets are now allowed in state 2, even during
    saturations
 *          2a - first is a delayed reset, keeping the system in state 2
 *               for the alloted 60 seconds, but then applying a reset when the
    system goes to state 3
 *          2b - if reset is pressed a second time in state 2, then it applies the
    standard reset.
 *    3 - resets are allowed in state 3 and state 4 even if saturations persist
```

```
*         note this still clears saturations and unblocks the damping loop (for
    another 60 seconds)
*    4 - added two new outputs to monitor the trigger status and time remaining in
    state 2 (triggered-holding)
*
* ISIWD_GPS.c
* adapted from ISIWD.c by CJK on Mar 28, 2012
*
* update April 2014 - SB -
* - 1 new input: isoStatus - check if the ST1 ISO is on or off by looking at the
    sum of the gains
* If sum = 0, ISO is off (otherwise it's on)
* The T240s are NOT part of the WD if ST1 ISO is off
*
* update Jan 2014 - BTL -
* - 2 new input triggers, IOPWD and Hardware WD
* - new output: latched first trigger - helpful for user diagnostics on previous
    trip
* - use FE_RATE to get the model rate
*
* ISIWATCHDOG is a state machine that watches the displacement
* sensor, T240, L4C geophone, coil drive BIO, actuator, and payload
* watchdog flags, and blocks the isolation loop path and/or damping
* loop path according to the flags.
*
* The same code is used for stage 1 and stage 2, but with a modified
* input/output list for the signals being used. The order of the
* signals was changed by BTL on Aug 25, 2011 so that the two stages
* are more similar.
*
* INPUTS:
* argin[0] = displacement sensor saturation flag
* argin[1] = stage 1: T240 sensor saturation flag/ Stage 2: GS13 saturation
* argin[2] = stage 1: L4C sensor saturation flag/  Stage 2: monitor of stage 1
    tripped
* argin[3] = binary i/o flag for stage 1 / stage 2 coil drive thermal shutdown
* argin[4] = actuator saturation
* argin[5] = payload Watchdog flag
* argin[6] = IOP watchdog flag (Jan 2014)
* argin[7] = Hardware watchdog flag (Jan 2014)
* argin[8] = reset
*
* OUTPUTS:
* argout[0] = flag to block damping loop path
* argout[1] = flag to block isolation loop path
* argout[2] = state of watchdog
* argout[3] = 8 character bit-field state after the first trigger
*    1 = displacement sensors;
*    2 = T240/ GS13
*    4 = L4C / ST1 WD
*    8 = BIO, - coil drive shutdown
*   16 = actuators;
*   32 = payload;
*   64 = IOP WD
*  128 = Hardware WD
* argout[4] = triggers;
* argout[5] = GPS Time of last trip;
* argout[6] = latched value of first trigger - helpful for user to look at previous
    trip
```

```
 * argout[7] = resetPressedOnceFlag; (0 if not pressed at 1, 1 if pressed once)
 * argout[8] = secondsRemaining;
 *
 * $Id$
 */

// Include the GPS time from the main FE code
extern unsigned int cycle_gps_time;
#define MODEL_RATE FE_RATE
// Enumerate the various WD States
typedef enum {ARMED = 1, TRIGGERED_HOLDING, DAMPING_ONLY, FULL_SHUTDOWN } WDState;

void ISIWATCHDOG(double *argin, int nargin, double *argout, int nargout) {

  static WDState state       = FULL_SHUTDOWN;              // Start in STATE 4 (FULL
     SYSTEM SHUTDOWN)
  static int firstTrigger    = 0;              // Start with no indication of
     triggers
  static int cycleClock      = 0;              // Used to count cycles when in state
     2.
  static unsigned int trip_time   = 0;
  static int latchedFirstTrigger  = 0;              // Start with no indication of
     triggers
  static int resetPressedOnceFlag = 0;              // used in state 2. change to 1 on
      first reset
  const int state2HoldCycles = 60 * MODEL_RATE;       // Wait for [seconds] X (cycles
     /second)
  int secondsRemaining = 0;                         // time left in state 2


  int blockIsolationLoopsFlag = 1;              // Start script with isolation path
     BLOCKED
  int blockDampingLoopsFlag   = 1;              // Start script with damping path
     BLOCKED

  // Read inputs
  int cpsTriggered     = argin[0];              // Displacement Sensor Flag
  int t240Triggered    = argin[1];              // T240 or GS13 flag
  int l4cTriggered     = argin[2];              // L4C or ST1 WD flag
  int bioTriggered     = argin[3];              // Binary I/O flag
  int actTriggered          = argin[4];              // Actuator flag
  int payloadTriggered = argin[5];              // payload flag
  int IOPTriggered     = argin[6];              // monitor for IOP watchdog
  int HWWDTriggered    = argin[7];              // monitor for Hardward WD trip
  int resetFlag        = argin[8];
 // double isoStatus        = argin[9];          //monitor the status of the ST1 ISO
     gains

 //   if (isoStatus == 0) {                         // if ST1 ISO is off, T240 are out
     of the WD
 //    t240Triggered = 0;
 // }



  // Check for triggers - make a 8 character bitfield so we can tell which triggers
     triggered
  // The !! (double inverse) syntax guarantees that the given value is 0 or 1.
  // The << is a bitwise shift left, i.e. 1 << 2 == 4 and 1 << 3 == 8.
```

```
// So, if the displacement sensors and actuators trigger, triggers = 010001;
int triggers =
  (!!cpsTriggered) | (!!t240Triggered << 1) | (!!l4cTriggered << 2) | (!!
      bioTriggered << 3) | (!!actTriggered << 4) | (!!payloadTriggered << 5) | (!!
      IOPTriggered << 6) | (!!HWWDTriggered << 7);


// State transitions
switch (state) {
case ARMED: // To STATE 1 (ARMED)
  if (triggers) {
    firstTrigger = triggers; // Record the bit-field state after the first trigger
    latchedFirstTrigger = triggers; // same as first trigger, but persists after
        reset.
    cycleClock = 0;
    trip_time = cycle_gps_time;
    state = TRIGGERED_HOLDING;
  }
  break;

case TRIGGERED_HOLDING: // To STATE 2 (TRIGGERED; HOLDING 3 SECONDS WITH DAMPING
    ENABLED)
  if (cycleClock >= state2HoldCycles) {  // if the time is up, go to the next
      state
    if (resetPressedOnceFlag) { //if we pressed reset once, then reset the WD now
        = go to armed
      firstTrigger = 0; // reset the first trigger
      state = ARMED; // and reset the WD
    }
    else {            // otherwise, proceed on to state 3
      cycleClock = 0;
      state = DAMPING_ONLY;
    }
  }
  else if (resetFlag && !resetPressedOnceFlag) {  // first reset press in state 2
    resetPressedOnceFlag = 1;
  }
  else if (resetFlag && resetPressedOnceFlag) {  // second reset press in state 2
      will reset the WD
    firstTrigger = 0; // Reset first trigger
    state = ARMED;
  }
  break;

case DAMPING_ONLY: // To STATE 3 (TRIGGERED; MONITORING WITH DAMPING ENABLED)
  if (triggers)
    state = FULL_SHUTDOWN;
  else if (resetFlag) {
    firstTrigger = 0; // Reset first trigger
    state = ARMED;
  }
  break;

default:
case FULL_SHUTDOWN: // To STATE 4 (TRIGGERED; FULL SHUTDOWN)
  if (resetFlag) {
    firstTrigger = 0; // Reset first trigger
    state = ARMED;
  }
```

```
    break;
  }

  // State actions
  switch (state) {
  case ARMED: // STATE 1 (ARMED)
    blockIsolationLoopsFlag = 0;      // Leave isolation loop path open
    blockDampingLoopsFlag = 0;        // Leave damping loop path open
    resetPressedOnceFlag = 0;         // be sure this gets set
    secondsRemaining = 0;             // only relevant for state 2
    break;

  case TRIGGERED_HOLDING: // STATE 2 (TRIGGERED; HOLDING WITH DAMPING ENABLED - )
    blockIsolationLoopsFlag = 1;     // Block isolation loops path
    blockDampingLoopsFlag = 0;       // Leave damping loops path open
    cycleClock = cycleClock + 1;     // Keep counting, it hasn't been long enough yet
    secondsRemaining = (state2HoldCycles - cycleClock)/MODEL_RATE;  //
        secondsRemaining is an integer for UI
    break;

  case DAMPING_ONLY: // STATE 3 (TRIGGERED; MONITORING WITH DAMPING ENABLED)
    blockIsolationLoopsFlag = 1;     // Block isolation loops path
    blockDampingLoopsFlag = 0;       // Leave damping loops path open
    resetPressedOnceFlag = 0;         // be sure this gets set
    secondsRemaining = 0;             // only relevant for state 2
    break;

  default:
  case FULL_SHUTDOWN: // STATE 4 (TRIGGERED; FULL SHUTDOWN)
    blockIsolationLoopsFlag = 1;      // Block isolation loops path
    blockDampingLoopsFlag = 1;        // Block damping loop path
    resetPressedOnceFlag = 0;         // be sure this gets set
    secondsRemaining = 0;             // only relevant for state 2
    break;

  }

  // Output
  argout[0] = blockDampingLoopsFlag;
  argout[1] = blockIsolationLoopsFlag;
  argout[2] = state;
  argout[3] = firstTrigger;
  argout[4] = triggers;
  argout[5] = trip_time;
  argout[6] = latchedFirstTrigger;
  argout[7] = resetPressedOnceFlag;
  argout[8] = secondsRemaining;

  return;
}
```