Discussion of the CART basis offset for the ISI Capacitive Displacement Sensors

Brian Lantz, Celine Ramet, Hugo Paris
T1300559-v1, June 25, 2013

1 Summary

This document describes the change in the biasing for the CPS sensors of the ISI. The biasing of the CPS sensors is important because it determines the closed loop (i.e. standard operation) location of the ISI platform at low frequency. Originally the CPS biasing was done at the CPSINF (**IN**put **F**ilter) bias point, i.e. in the local basis. Since the sensor calibration is done by the input filters, this bias is not easily calibrated. We propose to move the biasing to a point in the diagram where the CPS signals are calibrated and projected into the cartesian (CART) basis. We will also enable long term monitoring of the unbiased platform location in this calibrated, cartesian basis which should aid installation and monitoring. The ECR for this change is E1300548.

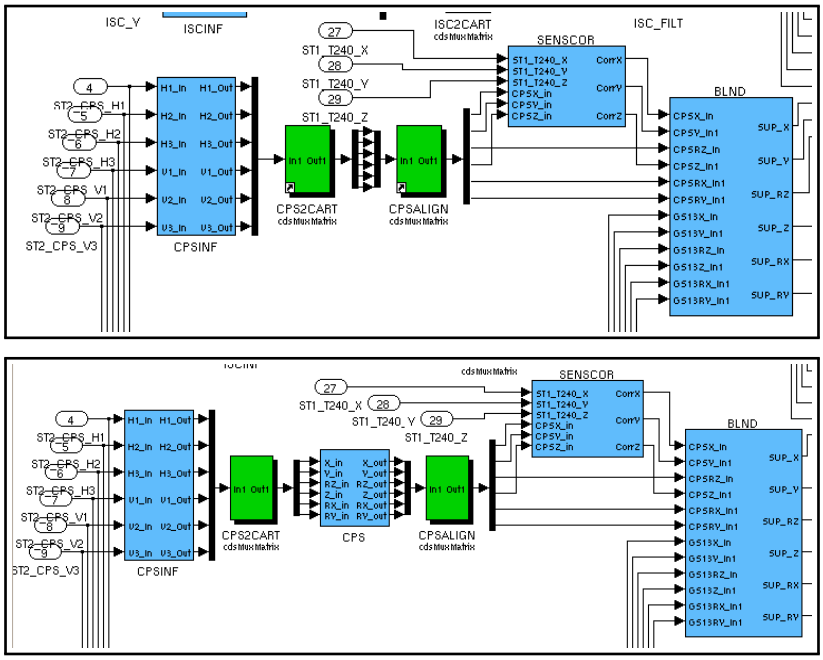

Figure 1: Detail of the simulink diagram before (above) and after (below) the modification. The new Cartesian-basis monitor and bias point is named CPS and is placed just after the matrix which projects the CPS signals in the cartesian basis.

2 Simulink Changes

The simulink changes are simple. A new block named 'CPS' is added just between the CPS local-to-cartesian projection matrix and the CPS alignment matrix as shown in figure 1. The block is named 'CPS' so that the Epics variables will have sensible names, as shown below. The CPS block is shown below in figure 2.

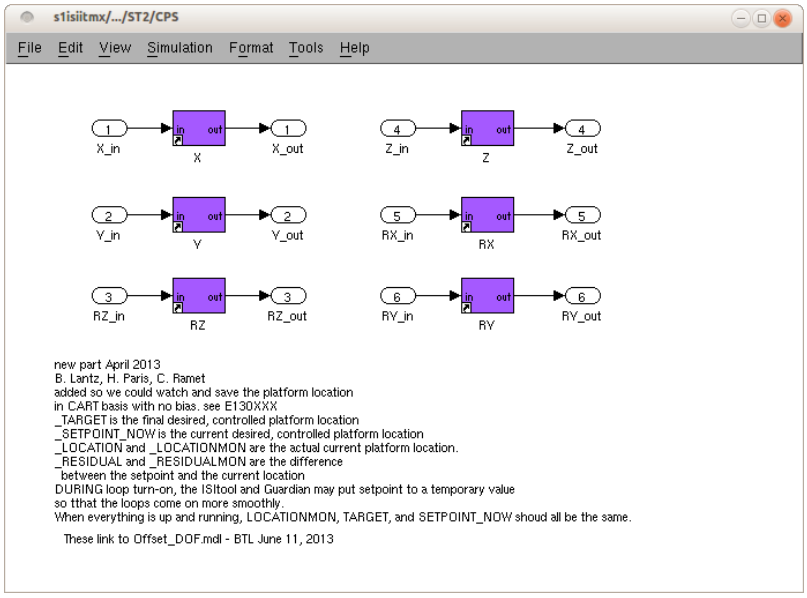

Figure 2: The CPS block is a set of 6 of the Offset_DOF library parts.

A new library part was created to monitor and control the biases for each degree of freedom. The library part is named OFFSET_DOF.mdl and is located at:
{userapps}/release/isi/common/models/OFFSET_DOF.mdl
The new part is shown below in figure 3

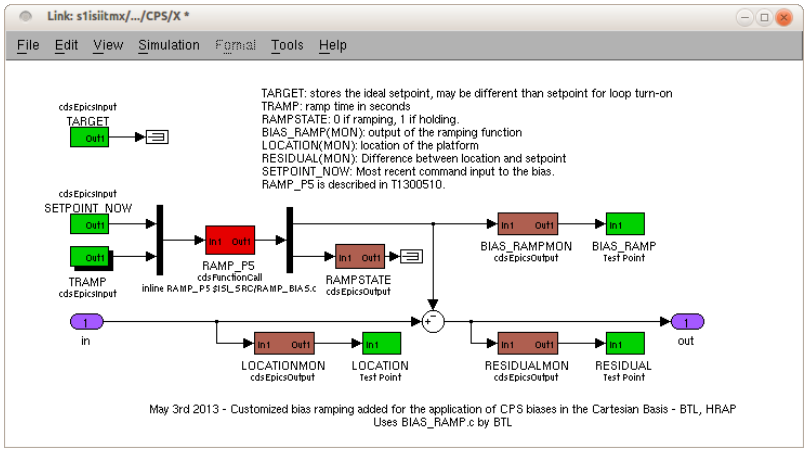

Figure 3: The new OFFSET_DOF library part is used to monitor and control the offsets for each cartesian DOF.

As can be seen in figure 3, the new part is essentially a way to subtract an offset from a channel. This will cause the LOCATION to be driven to the SETPOINT_NOW value when the channel is under feedback control. Several features have been added to make the part more usable. First, there are several monitor points. Second, the desired bias (entered in SETPOINT_NOW) is follow by a new ramp part. When the SETPOINT_NOW variable is changed, the output of the ramping part smoothly goes from the current output to the new setpoint in a time defined by TRAMP (in seconds). The ramp output (i.e. the actual bias) is monitored by the BIAS_RAMP testpoint and the BIAS_RAMPMON epics variable. While the output is ramping, the RAMPSTATE is 0. While it is holding a fixed output, the RAMPSTATE is 1. The ramping c-code is described in LIGO-T1300510. Finally, there is an 'extra' epics variable called TARGET which is used by the isolation loop turn-on code to save the location that the system should go to when the turn-on sequence is complete. This allows us to turn on the isolation loop with a temporary bias (set to the current platform position). Once the loops are on, the platform can be moved smoothly to the location stored in the TARGET variable by selecting a reasonable ramp time and changing the SETPOINT_NOW to the value in TARGET.
The variables in the new library part are:

1. **LOCATION(MON):** LOCATION and LOCATIONMON are the testpoint and epics variable for the current platform location. These will be in nanometers or nanoradians. e.g. S1:ISI-ITMX_ST1_CPS_X_LOCATIONMON.
2. **SETPOINT_NOW**: Epics variable defining the current desired setpoint for the location. The setpoint is an input to the ramping function, and when the SETPOINT_NOW variable is changed, the setpoint is ramped smoothly from the old

value to the requested value in time TRAMP.

3. **TRAMP**: Time, in seconds, for the output of the ramp part to move to a new value entered in the SETPOINT_NOW variable.
4. **BIAS_RAMP(MON)**: testpoint and epics variables monitoring the actual bias.
5. **RAMP_STATE**: epics variable which is 0 when the ramp part is changing the bias, and 1 when it is holding a constant value.
6. **RESIDUAL(MON)**: testpoint and epics variables monitoring the output of the summing junction. The difference between the location and the setpoint.
7. **TARGET**: epics variable used to store the ideal setpoint. Allows temporary setpoints for loop turn-on, etc.

3 MEDM Changes

The overview screen has been changed to add a small 'BIAS' block in the CPS path between the projection matrix and the alignment matrix. Figure 4 shows where the new block has been added for stage 1 of the BSC-ISI.

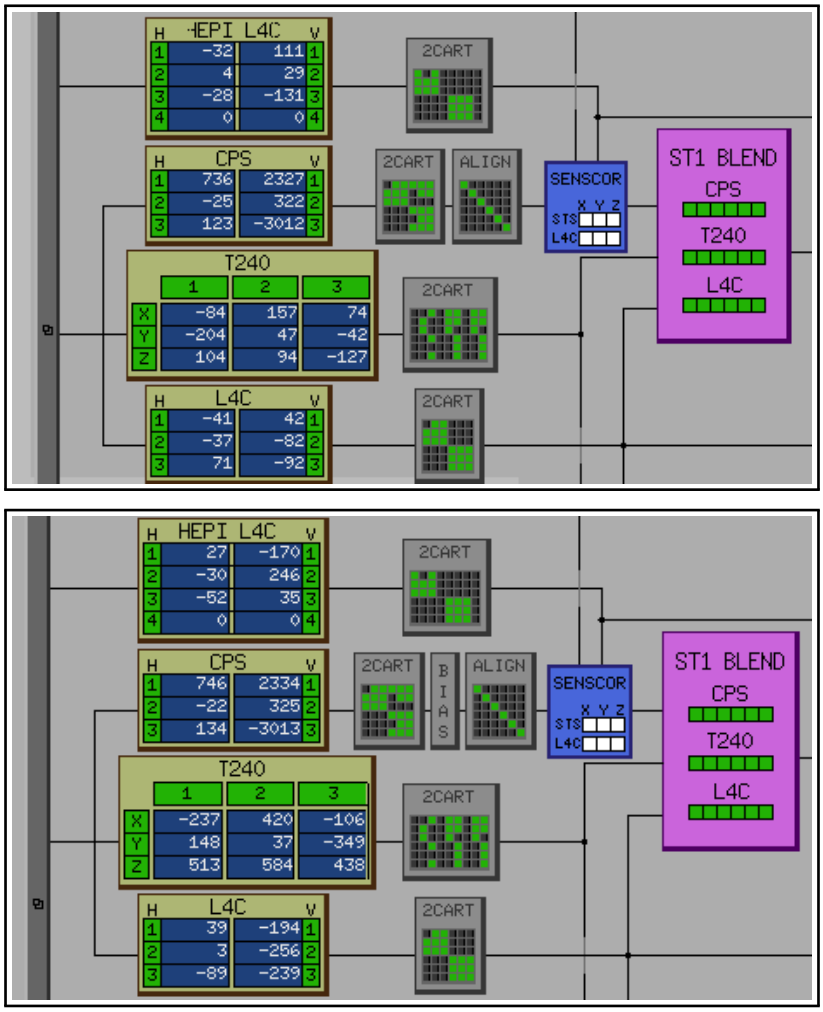

Figure 4: Detail of the MEDM overview screen for the BSC-ISI before (above) and after (below) the modification. The new BIAS button is between the CPS2CART matrix and the CPSalign matrix. Pressing the BIAS button opens a related display.

Clicking on the 'BIAS' button takes one to a screen displaying the locations and offsets for that particular stage. An example biasing screen is shown below in figure 5.

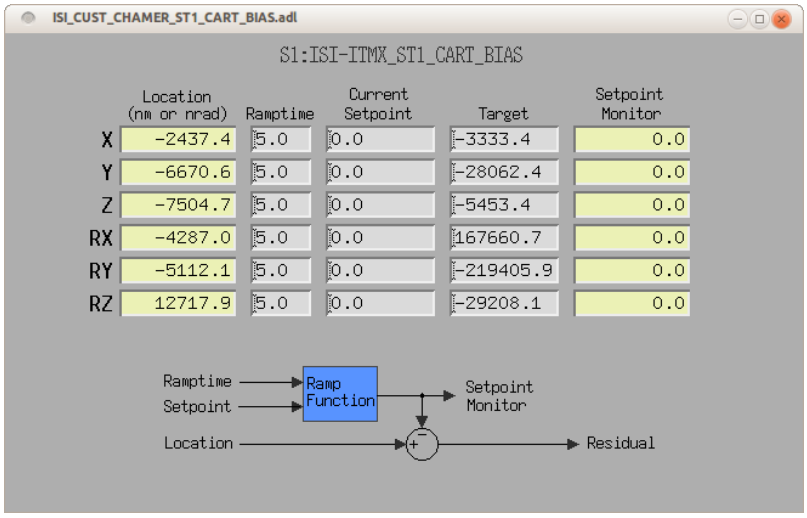

Figure 5: MEDM screen for the Stage 1 CPS Cartesian biases.

4 Script Changes

There are several impacted scripts. The main are the BSCISItool and HAMISItool. These scripts have been modified to adjust the turn-on process. The turn-on process is described in LIGO-E1300550. The code has been modified to check if the CART basis parts have been implemented for the stage in question, by checking for the existence of a particular the epics variable. For the BSC-ISI this is *<IFO>*:ISI-*<CHAMBER>*_*<STAGE>*_CPS_X_TARGET (e.g. S1:ISI-ITMX_ST1_CPS_X_TARGET) and for the HAM-ISI it is *<IFO>*:ISI-*<CHAMBER>*_CPS_X_TARGET (e.g. S1:ISI-HAM2_CPS_X_TARGET). If the CART basis variable is not present, we use the old code. However, if this CART basis variable is present, then we use the CART basis to do the isolation loop turn-on biasing, which is (generally):

1. Make sure the isolation loops are off. We assume the damping loops are on.
2. Set the cartesian SETPOINT_NOW variables to be the platform's current location, so that the residual is near zero before the loop comes on.
3. Turn on the isolation loops.
4. Set the SETPOINT_NOW to be the TARGET value. This platform will move to the specified position.

There are two helper scripts to manage the bias points:
setCartBiasSetpoints.pl sets the 6 setpoints for a stage to be the average of the recent

location.
storeCartBiasTargets.pl saves the setpoints into the targets.