

SVT Commissioning and Operation

Omar Moreno

Santa Cruz Institute for Particle Physics
University of California, Santa Cruz

omoreno1@ucsc.edu

January 27, 2014

Heavy Photon Search Software Review



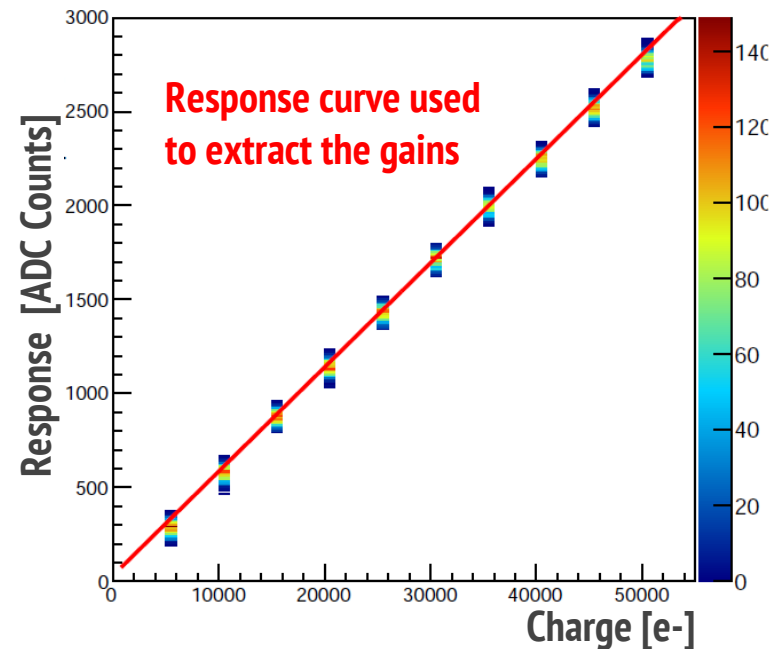
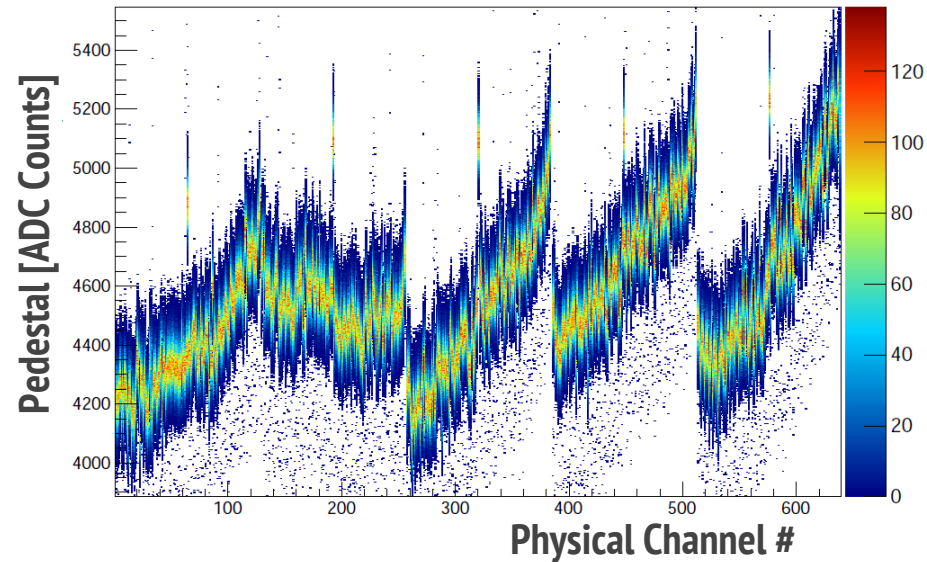
Overview

- ❑ Calibrating the SVT
 - ❑ What calibration constants are needed?
- ❑ Timing In
- ❑ Alignment of the SVT
- ❑ Configuring the SVT
- ❑ Monitoring
 - ❑ Online
 - ❑ Event Display
 - ❑ Offline

Calibrating the SVT

- Calibrating the SVT will require
 - Extraction of the pedestals and noise for each of the 23004 SVT channels
 - Extraction of the gains and offsets
 - Runs at different t_0 values used to study the pulse shape
- All calibration constants will be loaded into the conditions database
 - Each calibration run will be assigned a unique identifier that will be related to the appropriate range of runs taken during that period
- Thresholds for all channels will also be calculated and loaded to the conditions database

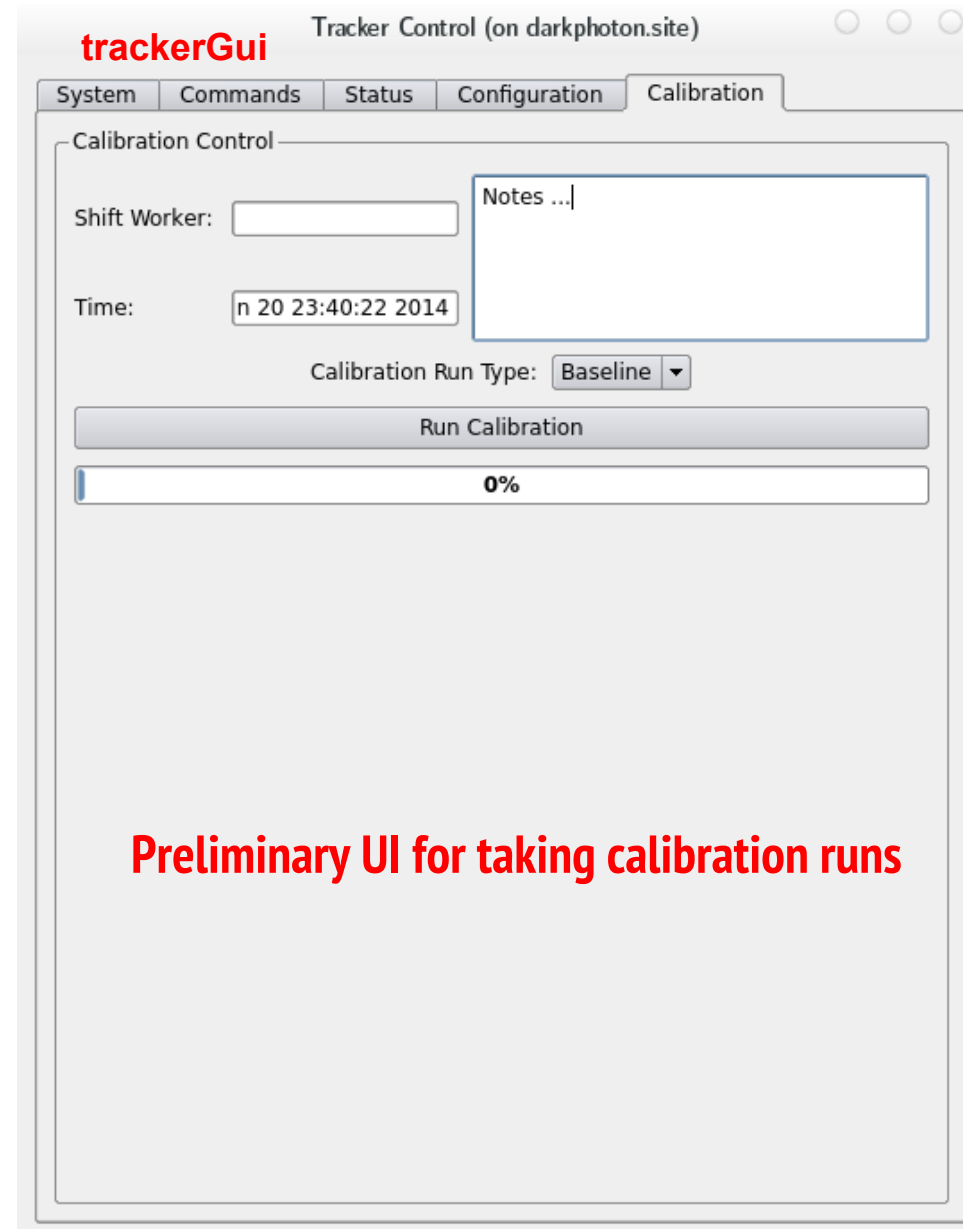
The code used to extract all calibration constants has been developed and was extensively tested during the test run → API to access the database is under development.



Taking Calibration Runs

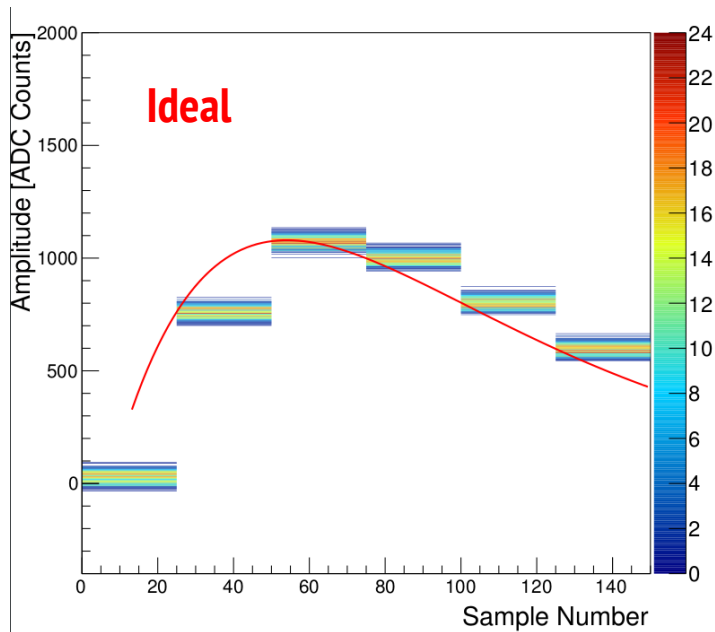
- ❑ Several calibration runs are expected to be taken by shift takers during the engineering run and beyond
- ❑ Currently, calibration runs can be taken manually via the trackerGui (SVT DAQ interface) or by using python scripts
 - ❑ All calibration constants are stored locally in text files
- ❑ The desire would be to automate the calibration taking process and integrate it into the trackerGui in order to allow shift workers to easily take calibration runs

Preliminary work has been done on a user interface that will allow a shift worker to easily take calibration runs

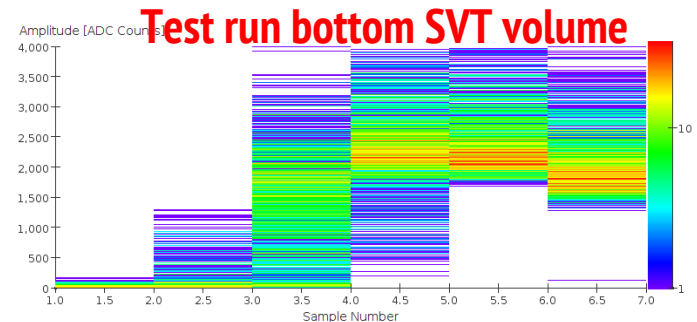
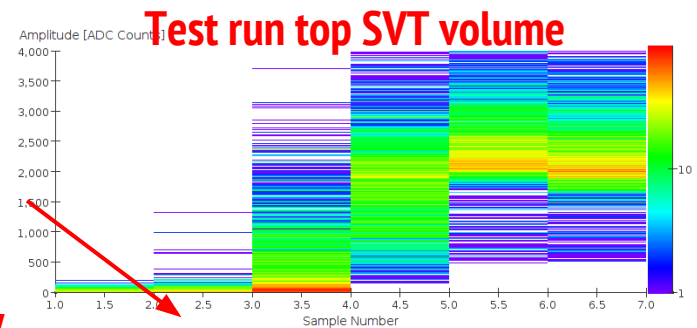


Timing In the SVT

- ❑ The output of the APV25 (SVT readout chips) shaper is sampled at 40 MHz with each sample stored in a 192 cell analog pipeline that is not read out until a trigger signal is received
- ❑ The readout position along the pipeline is set depending on the trigger latency
 - ❑ The analog pipeline allows a trigger latency up to $\sim 3.5 \mu\text{s}$
- ❑ Setting the proper latency i.e. “Timing In” is then crucial to ensuring that the signal that corresponds to the trigger is read out and to the quality of the SVT data
- ❑ Simulation will be used to develop a timing in procedure ➡ data taken during the test run will be used to test it
 - ❑ Will look at how hit rates, number of tracks per event etc depend on the latency chosen



**Top and bottom
timed in
differently.
Was this
optimal?**



Alignment of the SVT

- ❑ The alignment constants [36*6 (3 translational, 3 rotational) = 216] for the SVT will need to be determined prior to running
 - ❑ The alignment constants will be stored in the conditions database so they are accessible by the simulation and reconstruction code
 - ❑ Alignment of the SVT will be done by using the track fitting algorithm General Broken Lines (GBL) in conjunction with Millipede
 - ❑ Should bow constants be included? The effect of bowing can and should be studied with simulation
- ❑ A subset of the alignment constants of the first three layers will need to be recalculated every time the SVT is extracted from the beam
 - ❑ This calculation should be integrated into the monitoring app or into the motion control gui so it can be done by shift workers
- ❑ More details will be given in Pelle's talk

Configuring the SVT

Simple trackerGui used during the test run

Shows the status of the SVT i.e. indicated whether the SVT is configured, any errors that may have occurred and how many APV25's are synced

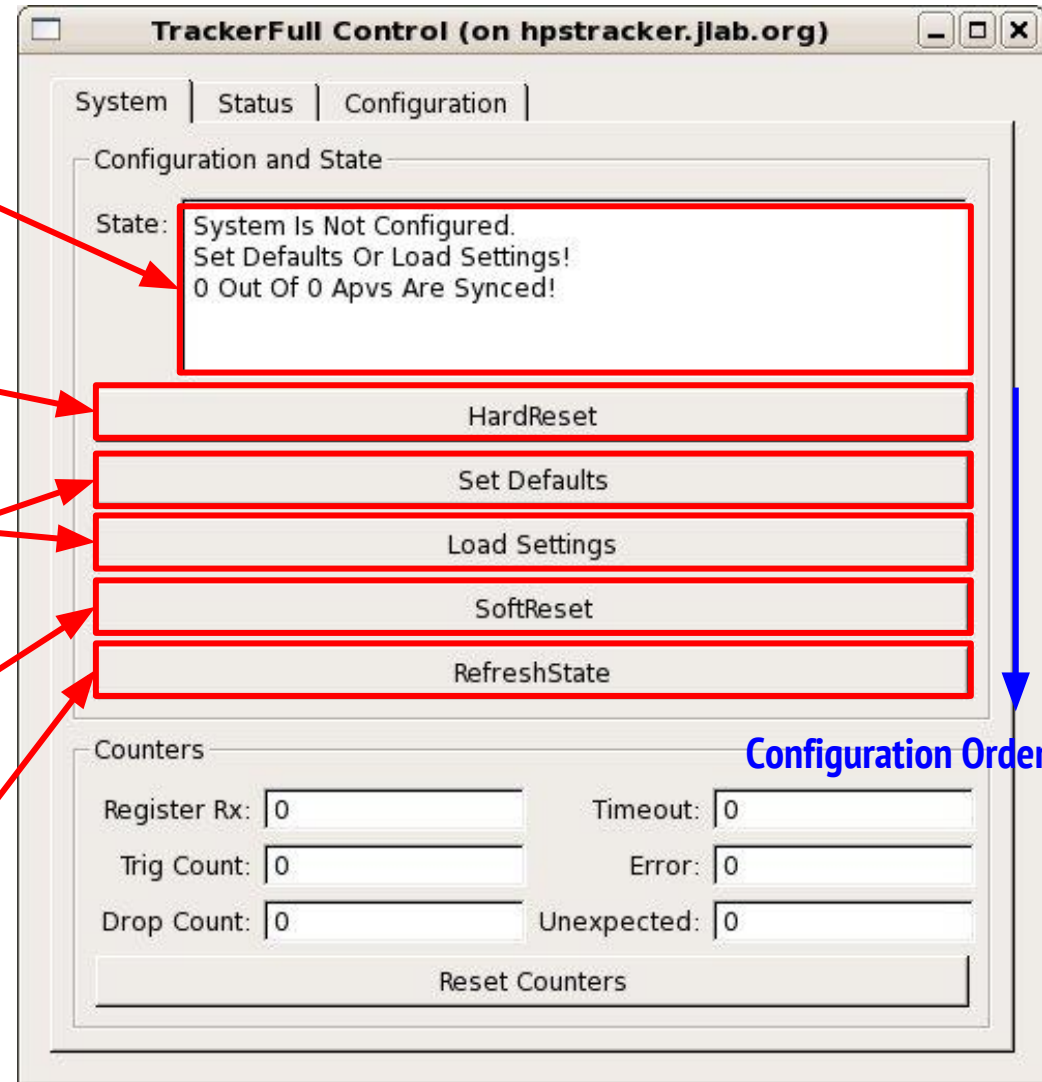
Send a reset signal to all APV25's ➡ has the same effect as power cycling

Loads the default detector settings

Resets the position of the trigger and writer pointers i.e. attempts to resync all of the APV25's

Gets the current state of the SVT

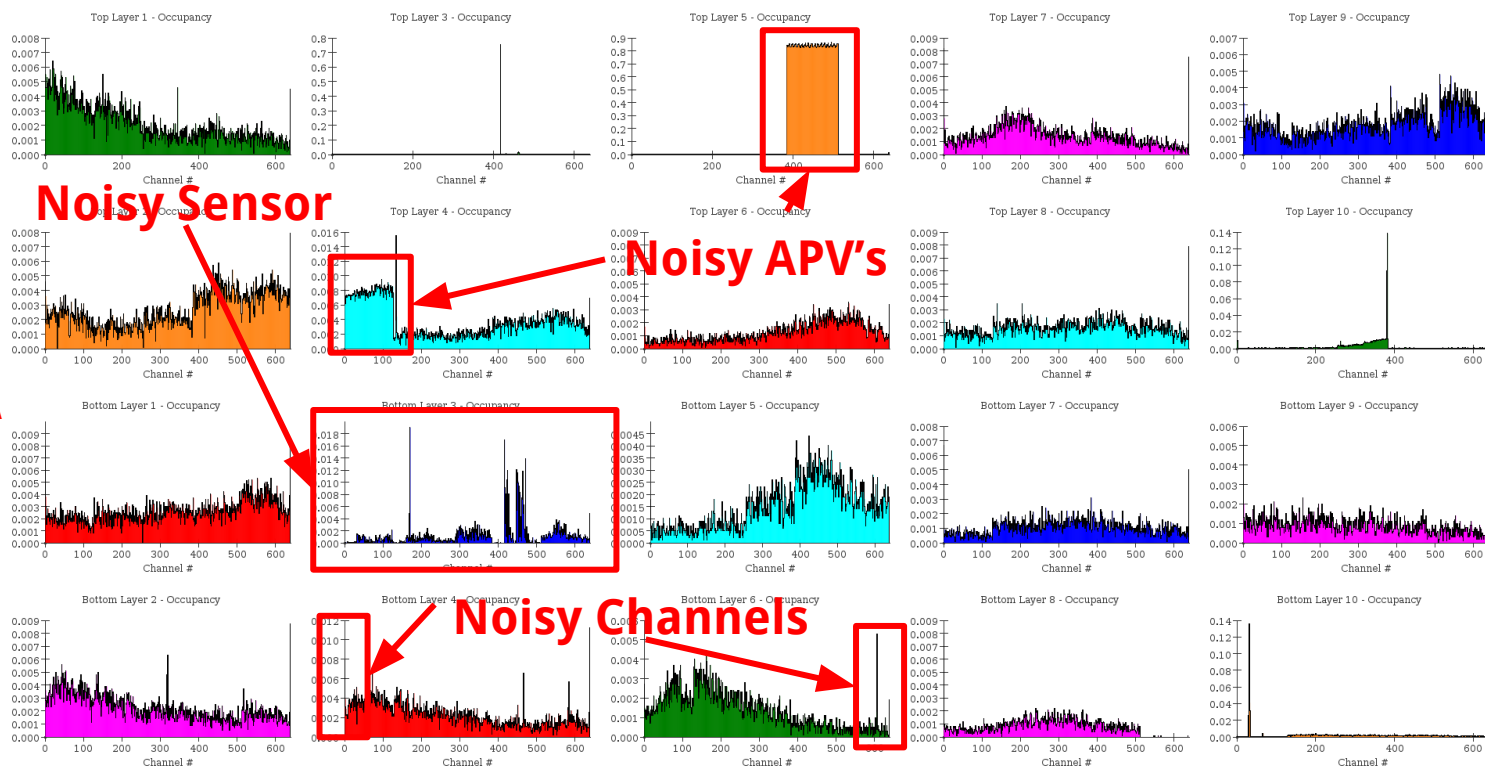
Integration of configuration of the SVT into CODA is still being discussed. However, at the moment the trackerGui works well and has been tried and tested.



Online Monitoring of the SVT

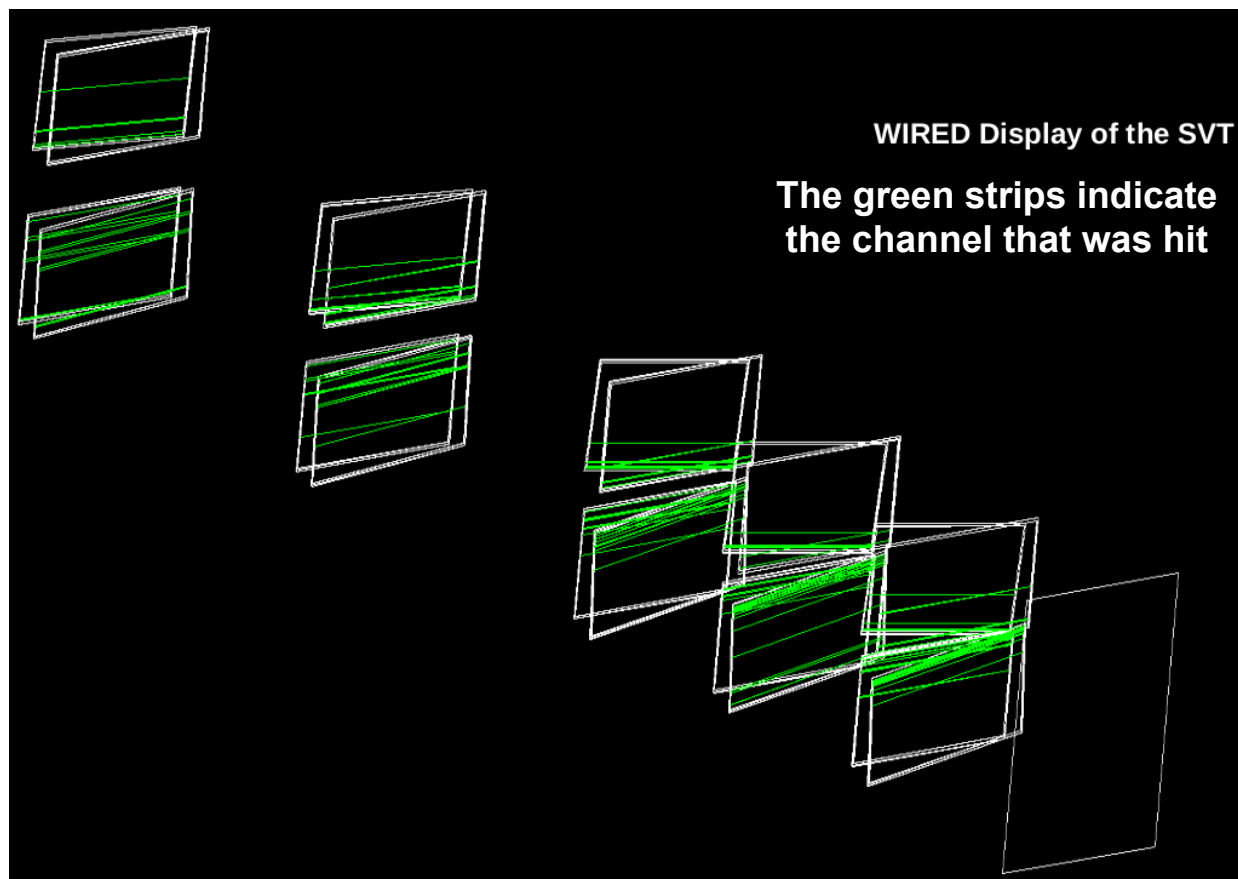
- ❑ A monitoring application which makes use of the lcsim software package currently exist [See Jeremy's talk]
 - ❑ Currently makes use of an AIDA back-end to make plots, however, this will be replaced by JFreeChart soon
 - ❑ Plots are easy to develop ➔ they are just lcsim drivers
- ❑ Several plots currently exist (and more can be added) but the crucial ones include
 - ❑ Occupancy - can be used to spot when pedestals have shifted, noisy channels or if some APV25's are out of sync
 - ❑ Shaper signal samples - will allow shift takers to check if the SVT is timed in correctly
 - ❑ Data rate
 - ❑ DAQ Errors

A "System Monitoring" framework that will allow users to define tests that need to be run on each of the plots still needs to be developed and integrated into the monitoring app. A test can be as simple as checking whether a plot has feature i.e. variations in occupancy that are unexpected.



Event Display

- ❑ The leading candidate for an event display is Wired \Rightarrow requires a lot of development
- ❑ A simple event display can also be created using ROOT with the LCIO C++ API used to access collections created with the reconstruction code
- ❑ The use of the event display wouldn't be limited to looking at events, but can also be used to indicate problems with the SVT etc.



SVT Slow Controls

- ❑ The slow control monitoring requirements for the SVT include
 - ❑ Power to the front end and flange boards
 - ❑ High voltage bias
 - ❑ hybrid power
 - ❑ hybrid and front end board temperatures
 - ❑ SVT position
 - ❑ Chiller temperature, temperature in and out of the cooling manifolds
- ❑ Need to understand what variations in these variables we can tolerate so that meaningful alarms can be set
- ❑ What was in place for the test run is a good starting point with some small modifications ➔ **work to implement these requirements is already underway**

Test Run Hybrid Temperatures

	FPGA 0	FPGA 1	FPGA 2	FPGA 3	FPGA 4	FPGA 5	FPGA 6
Hybrid 0	21	21	22	20	20	21	20
Hybrid 1	21	21	22	20	21	0	21
Hybrid 2	21	20	21	20	21	21	20

SVT DAQ HEARTBEAT ● Dead

Test Run SVT Power GUI

HPS SVT POWER SUPPLIES

WARNING
TO BE USED BY SVT EXPERTS ONLY !

Bias Voltage Setpoints 0

Hybrid	TRIPS	ON/OFF	MEASURED CURRENTS			BIAS VOLTAGE	
			I _{VDD}	A _{VDD}	V ₁₂₅	MEASURED	DEMAND
Hybrid 0	●	OFF ON	0,000	0,000	0,000	0,000	0,000
Hybrid 1	●	OFF ON	0,000	0,000	1,000	0,000	0,000
Hybrid 2	●	OFF ON	0,000	0,000	0,000	0,000	0,000
Hybrid 3	●	OFF ON	0,000	0,000	0,000	0,200	0,000
Hybrid 4	●	OFF ON	0,000	0,000	0,000	0,000	0,000
Hybrid 5	●	OFF ON	0,000	0,000	1,000	0,000	0,000
Hybrid 6	●	OFF ON	0,000	0,000	0,000	0,000	0,000
Hybrid 7	●	OFF ON	0,000	0,000	0,000	0,000	0,000
Hybrid 8	●	OFF ON	0,000	0,000	0,000	0,100	0,000
Hybrid 9	●	OFF ON	0,000	0,000	0,000	0,000	0,000

TOP SVT

Hybrid	TRIPS	ON/OFF	MEASURED CURRENTS			BIAS VOLTAGE	
			I _{VDD}	A _{VDD}	V ₁₂₅	MEASURED	DEMAND
Hybrid 0	●	OFF ON	0,000	0,000	1,000	0,000	0,000
Hybrid 1	●	OFF ON	0,000	0,000	2,000	0,000	0,000
Hybrid 2	●	OFF ON	0,000	0,000	0,000	0,100	0,000
Hybrid 3	●	OFF ON	0,000	0,000	0,000	0,100	0,000
Hybrid 4	●	OFF ON	0,000	0,000	0,000	0,000	0,000
Hybrid 5	●	OFF ON	0,000	0,000	2,000	0,000	0,000
Hybrid 6	●	OFF ON	0,000	0,000	0,000	0,000	0,000
Hybrid 7	●	OFF ON	0,000	0,000	0,000	0,100	0,000
Hybrid 8	●	OFF ON	0,000	0,000	6,000	0,000	0,000
Hybrid 9	●	OFF ON	0,000	0,000	3,000	0,000	0,000

BOTTOM SVT

VOLTAGES MORE PARAMETERS

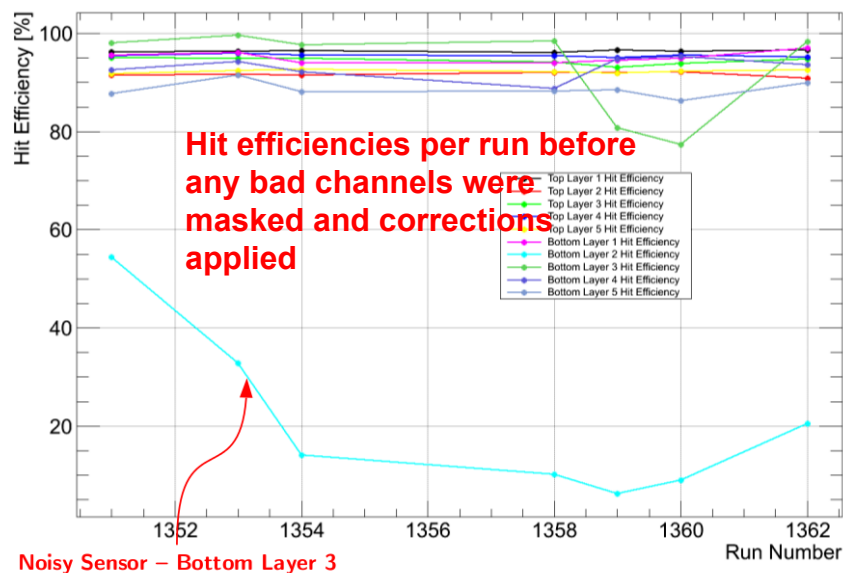
Hybrid 0 Hybrid 0
Hybrid 1 Hybrid 1
Hybrid 2 Hybrid 2
Hybrid 3 Hybrid 3
Hybrid 4 Hybrid 4
Hybrid 5 Hybrid 5
Hybrid 6 Hybrid 6
Hybrid 7 Hybrid 7
Hybrid 8 Hybrid 8
Hybrid 9 Hybrid 9

Power Supply Interlock ●

Offline Monitoring

- ❑ A subset of the data from all runs will be checked by shift workers immediately after being taken
 - ❑ The “system monitoring” capabilities of the monitoring application can be used to run tests on subset from the tests
- ❑ SVT Performance
 - ❑ Calibrations runs will be taken every few runs in order to check for any shifts in the pedestals and changes to the noise \Rightarrow thresholds will be recalculated and loaded when needed
 - ❑ Hit efficiencies - a dip in the efficiencies will likely point to an issue with a sensor
 - ❑ Track residuals, Track χ^2 - Will be used to check if for any issues with the alignment of the SVT
 - ❑ Basic track variables - number of tracks, hits,
 - ❑ Beam spot
 - ❑ Ecal cluster/track matching - This will also give us insight as to whether there are issues with the alignment of the SVT
 - ❑ Cluster charge - will give us insight as to whether pedestal shifts occur during a run

Most plots that will be used in the offline analysis already exists. See Matt's talk for more details.



Summary and Needs

- ❑ The software requirements that will be needed to calibrate, configure, operate and monitor the SVT are well understood
 - ❑ Calibration of the SVT
 - ❑ All calibration code has been written and was successfully tested during the test run
 - ❑ Development of the backend used to load calibration constants to the conditions database is underway [\[Moreno\]](#)
 - ❑ A shifter interface is also being developed [\[Moreno\]](#)
 - ❑ Timing In
 - ❑ Development of an algorithm that will allow to efficiently “time in” the SVT will begin soon [\[Moreno\]](#)
 - ❑ Alignment [\[Hansson\]](#)
 - ❑ Configuration of the SVT
 - ❑ SVT configuration UI is in place. Integration into CODA is being discussed but is not crucial [\[Herbst, Moreno, Hansson\]](#)
 - ❑ Online Monitoring
 - ❑ Main monitoring framework is in place and plots have been developed
 - ❑ Still need to develop “system monitoring” capability [\[Uemura, McCormick, Moreno\]](#)
 - ❑ Offline Monitoring
 - ❑ Several of the offline monitoring plots have already been developed
 - ❑ The tests that will be used to check for the quality of the data still need to be developed [\[Graham, Hansson, Moreno\]](#)
 - ❑ Event Display [\[McCormick\]](#)
 - ❑ Slow Controls [\[Hansson, Uemura\]](#)