Description of Current SBSTimingHodoscope Implementation in SBS-Offline

University of Glasgow: R. Montgomery (Rachel.Montgomery@glasgow.ac.uk), D. Hamilton, R. Marinaro, G. Penman

Last updated: 26/08/2021

GitHub for SBS-Offline

The SBS-offline code is under development for timing hodoscope (TH) purposes and can be found online in the repository at:
https://github.com/rachel-m/SBS-offline.git
To clone it and navigate to the right dev branch:

- git clone https://github.com/rachel-m/SBS-offline.git
- git checkout BB_TH_dev

GitHub for the DB and Replay files

For development, example DB files and replay scripts can be copied from:
https://github.com/rachel-m/HodoSW.git
To clone it:

- git clone https://github.com/rachel-m/HodoSW.git

The DB files are in the directory DB and the replay scripts are in the replay directory. Please note that there are two DB files for the hodoscope: db_bb.hodoadc.dat and db_bb.hodotdc.dat. The ADC one is only relevant during calibrations and is called in the replay_BBhodo.C script alongside the ADC DB file, since in the case where we want to replay ADC and TDC spectra we have to replay two TH detectors – one for the TDC data and one for ADC data.

Overall Structure of SBS-Offline for the Timing Hodoscope

In general for the timing hodoscope (TH) the left and right sides are defined when looking at the hodoscope from the front, with the e' travelling into the TH. The left side is on the left of the e' and the right side is on the right of the e'. In the code, the left side is labelled 0 and the right side is labelled 1.

In all code the bar numbering starts from 0 and increases sequentially to 89. Bar number 0 is located at the bottom of the TH stack, closest to the Hall floor, bar 89 is at the top of the stack, closest to the Hall roof. There are actually only 89 bars instrumented (not the originally planned 90 bars). The code includes the 90^{th} bar (labelled 89 in the code) only for legacy and convenience, but be aware that all entries for this bar are null, even though code and documentation may discuss 90 bars and 180 pmts.

With this detector design, we have, in accordance with the row column layer structure of SBSGeneric detector, 2 rows, 90 columns and 1 layer. Each row corresponds to a side of the TH (0 for left, 1 for right). Each column corresponds to a bar (0 for bottom and 89 for top).

Two CAEN v1190 TDCs are used for the TH. It is always assumed that the first module instruments the left PMTs of the TH, and the filling of the channels is done sequentially from 0 to 89 for the 90 left PMTs. The reference time is connected to channel 127 of the module. All channels between 90 and 126 are empty. The same mapping scheme is assumed for the second TDC module, only it is connected to the right PMTs.

Note that for calibration purposes two ADC modules are used to instrument a subset of 64 PMTs of the TH at a time. This corresponds to 32 bars, with left/right readout simultaneously. It is always assumed that the first 32 ADC channels instrument the left PMTs (this corresponds to the first ADC module) and the second 32 ADC channels instrument the right PMTs (this corresponds to the second ADC module).

Database files: There are two database files – one describing the TDC setup and one for the ADC setup. Currently they are named db_bb.hodotdc.dat and db_bb.hodoadc.dat respectively. A lot of the parameters are explained in the SBSGenericDetector documentation (https://github.com/MarkKJones/SBSBBShowerDoc).

SBSTimingHodoscopePMT: This class inherits from SBSElement and expands upon it, so that you can add hodoscope relevant parameters from the database files to each element (ie each PMT), like time walk correction parameters, bar number, side of bar (left or right). There are 180 SBSTimingHodoscopePMTs when running SBS-offline for the normal TH operation. When including the ADC info for calibration, there will be 64 SBSTimingHodoscopePMTs.

SBSTimingHodoscopeBar: This class calls SBSTimingHodoscopePMT. The implementation of a bar collects of the pointers to each of a left and a right SBSTimingHodoscopePMT. The bar number in the stack is also set.

SBSTimingHodoscope: This is the main class for the timing hodoscope (TH) implementation. It inherits from SBSGenericDetector (which is comprised of SBSElements), and relevant SBSGenericDetector documentation should be referred to for the specifics of how that class works (https://github.com/MarkKJones/SBSBBShowerDoc). Currently the extra parameters which are specific to the hodoscope are the tdcbaroffset, adcbaroffset, timewalk0map and timewalk1map which are explained further below. The functions of the class are explained below.

SBSTimingHodoscope:ReadDatabase: The ReadDatabase function first calls the SBSGenericDetector ReadDatabase function. Then it reads in additional specific parameters from the TH database files. At the moment this includes the following.

- timewalk0map. This is a non-optional map from the TH database files. It is an array of the gradient parameter from the linear time walk calibration result for each PMT.
- timewalk1map. This is a non-optional map from the TH database files. It is an array of the intercept parameter from the linear time walk calibration result for each PMT.
- tdcbaroffset. This is an optional entry. This should always be zero in real life. It is only inserted in case at any point one could only instrument a certain subset of the TH with TDCs only (eg in the event of TDC module failure).
- adcbaroffset. This is an optional entry. This should be set in the ADC DB file to reflect the first bar instrumented by the ADC readout (again bar numbering from 0 at the bottom of the stack). This allows us to output the correct ADCBarID in the output tree while only instrumenting a subset of bars at a time.

At the end of ReadDatabase, the ConstructHodoscope function is called. This function looks through the fElementGrid set by the SBSGenericDetector ReadDatabase, to sort each the relevant SBSElement pointers into the appropriate SBSTimingHodoscopePMT and SBSTimingHososcopeBar objects. In this way the TH is comprised of 90 SBSTimingHososcopeBar objects, which are a collection of pointers to 180 SBSTimingHodoscopePMT objects, which themselves contain the pointers to the original SBSElements.

SBSTimingHodoscope:DefineVariables: This is the function which defines the outputs from the SBSTimingHodoscope class which will be written to the replayed root tree. The different variables are explained in the code by their names. Add anything here you want to be available in the output.

SBSTimingHodoscope:FindGoodHit: At the moment this does nothing, but this would be a good place to add an algorithm to determine the best hodoscope hits in a ranked order.

SBSTimingHodoscope:CoarseProcess: Currently this function is finding bars with a good hit on each PMT. First the SBSGenericDetector::CoarseProcess is called. It finds one good hit for each SBSElement. For TDC hits this is defined in the multihit TDC to be the hit closest to the bb.hododc.tdcGoodTimeCut value in the TDC DB file. The same find good hit process is performed for the reference hits. In SBSTimingHodoscope if both PMTs of a bar have a good hit, the relevant information for the bar (which includes only **one** TDC value per PMT) is added to the relevant fGoodBar vectors to be recorded to the output root file. Time walk is also applied before pushing back info to the fGoodBar vectors. There are branches for the fGoodBar times with and without the time walk correction – so that a check of its implementation can be made after replaying. Currently there is no check for a good reference hit – this is assumed to always be true (since good hits in the generic detector are already reference time corrected before being passed to SBSTimingHodoscope). We can add this check for safety if deemed important.

SBSTimingHodoscope:FineProcess: At the moment this does nothing, but this would be a good place to add a process using other subsystem detectors to determine the best hodoscope hits.

SBSTimingHodoscope:TimeWalk: This uses the two timewalk parameters per TDC channel to perform a linear time walk correction using the LE and TOT times. This linear correction has been found from literature review of the NINO chip in similar applications.

SBSTimingHodoscope:ConstructHodoscope: This is called in ReadDatabase and sorts the generic detector elements into the appropriate TH ones.

SBSTimingHodoscope:ClearEvent and the destructor: The standard type of function to clear vectors which are re-filled and written out per event, and a standard destructor.