

Covariance matrix data format

Guglielmo De Nardo
University of Napoli and INFN

Current status in software

- waveform fit performed in basf2 module ECLDigitizer
 - energy deposit from GEANT from Datastore (ECLHit)
 - Pre-calculated fixed numbers hardcoded in module common for all crystals
 - Waveform shape
 - few parameters needed by fit algorithm
 - covariance matrix
- Alex Boblov determined covariance matrix for all crystals

New Digitizer

- Plan is to update ECLDigitizer module to read fit functions + algorithm parameters + covariance matrix from a condition database
 - Condition database is not going to be available in the short term
- Intermediate solution would be to store data in a single root file
 - Output file of Alex's calibration procedure
 - Read by ECLDigitizer
 - Data format (data object) as near as possible (same?) to what is going to be in the condition database

Single object format

- Data needed for a single fit (single crystal) may be aggregated in a single data structure

Covariance matrix:	136 floats
Function shape:	10 floats
Algorithm pars:	10 integers

Single object size would be $156 \times 4 \text{ bytes} = 624 \text{ bytes}$

Maximum size = 1 data structure x crystal = 5.45 MB

Size can be reduced by sharing same objects among groups of crystals

1 k_a	multipliers power of 2 for fg31,fg41
2 k_b	multipliers power of 2 for fg32
3 k_c	multipliers power of 2 for fg33,fg43
4 y0_start	start point for pedestal calculation
5 chi_thresh	chi2 threshold for quality bit
6 k1_chi	multipliers power of 2 for f
7 k2_chi	multipliers power of 2 for chi2 calculation
8 hit_thresh	hardware threshold(to start digitization)
9 low_amp_thresh	threshold to calculate time
10 skip_thresh	threshold to send data to collector

Root file format

- The most flexible choice would be to store data in a TTree with 2 branches
 - One branch holds a single data object
 - Other branch holds an array of crystal-id identifying the group of crystals sharing the same object
- If we change the logic of grouping the crystal together we do not need to change the format and the code that reads the constants

Related changes on transient side

- Accordingly, during basf2 execution we have to keep in memory (in the DataStore)
 - An array of data objects
 - A lookup table (array of indexes) for crystal → data object association
- Reading the rootfiles and objects creation in the DataStore can be done at EclDigitizer initialization (or by an ad-hoc module)
- Having it in the datastore makes other basf2 modules access it for testing and possible additional studies