Beam Monitor in SHOE

Update status of the bm_calibration branch

Yunsheng Dong

Current state

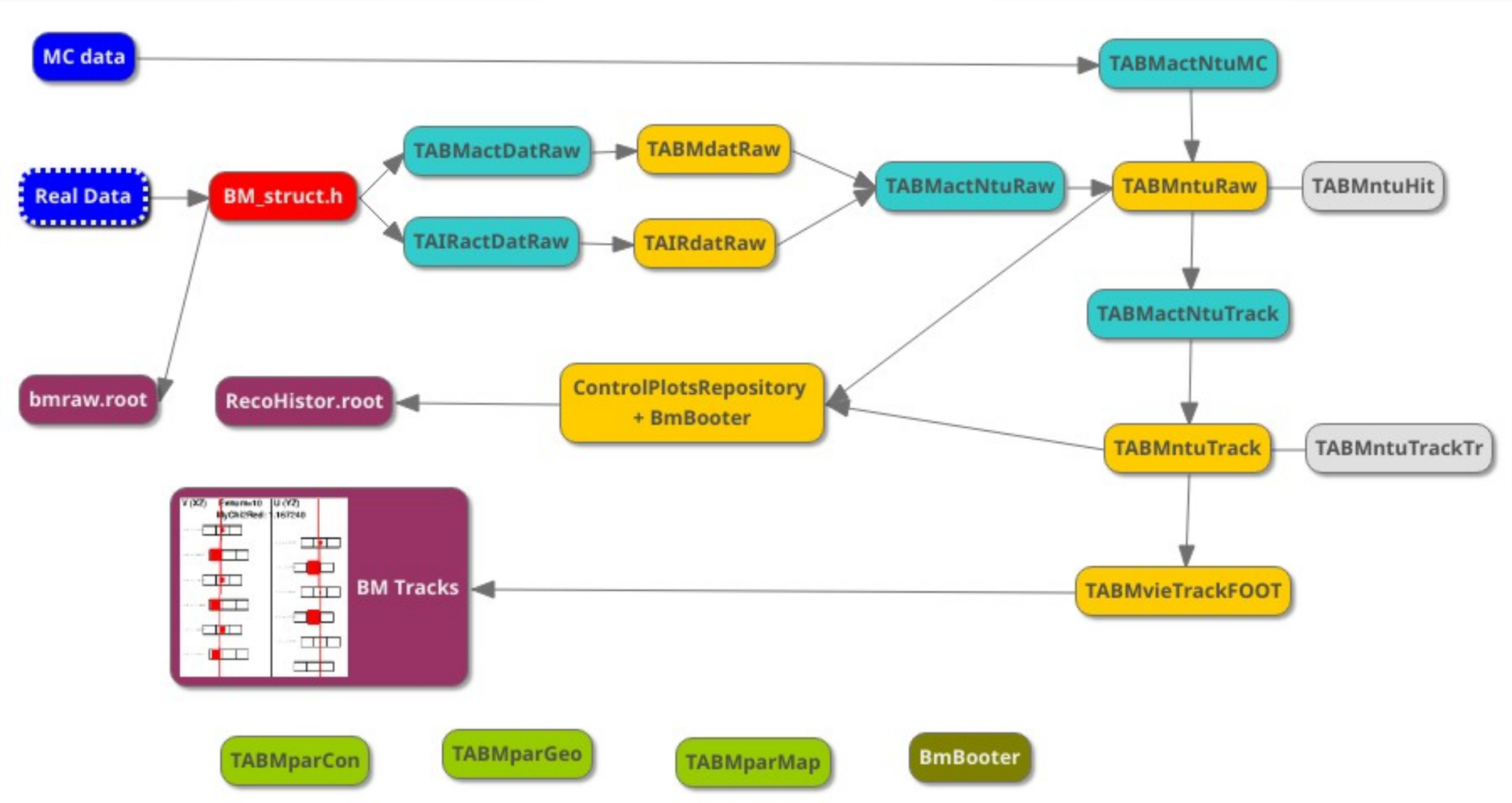
- Beam Monitor branch: bm_calibration
  (from the master branch of the last software meeting held in Bologna ~ 2/2018)
- The detector relevant libraries are mainly in TABMbase/*
- At present, is used to perform both MC and real data analysis

BmBooter.hxx

```
//initialize
void Initialize( TString instr_in, Bool_t isdata_in);
void evaluateT0();//evaluate the T0 from datafile

//process
void Process();
void FillDataBeamMonitor();
void Projecttracktr(); //to save the tracktr2dprojects matrix
void ResidualDistance();//to save the residual_distance matrix

//read data event
Bool_t read_event(Bool_t);//read an event from the datafile and charge bmstruct, it returns true if it read the event,
false if the file is end
Bool_t drop_event();//read an event from datafile and discharge it, it return false if the file is end
void monitorQDC(vector<Int_t>& adc792_words);
void clear_bmstruct(Bool_t forced);
void PrintBMstruct();// to print the content of bmstruct

//finalize
void Finalize();
void PrintSTrel(); //print the st relations in RecoHistos
void PrintEFFpp(); //print the efficiency evaluation with the pivot-probe method
void PrintProjections();// print the projected fitted tracks saved in tracktr2dprojects
void PrintResDist();//print the residual_distance matrix
void Allign_estimate();  //estimate the bm allignment with the residual methods and print the results
void evaluate_cell_occupy(); //fill the cell_occupy matrix
void efficiency_pivot_probe();//evaluation of the efficiency with the eff_pp matrix (pivot-probe method), made with the
cell_occupy matrix
void efficiency_paoloni();//evaluation of the efficiency with the "Paoloni" method
void efficiency_fittedplane();//evaluation of the efficiency with the "Paoloni" method on fitted tracks
```

- The Beam Monitor does not deal with the fragments tracking, so it can be treated as a separated detector.
- The configuration libraries (TABMpar*) are initialized in Booter.cxx
- The management of all the Beam Monitor classes is given by BmBooter (similar to Booter).

Scheme


MC data
TABMactNtuMC
Real Data
BM_struct.h
TABMactDatRaw
TABMdatRaw
TAIRactDatRaw
TAIRdatRaw
TABMactNtuRaw
TABMntuRaw
TABMntuHit
TABMactNtuTrack
bmraw.root
RecoHistor.root
ControlPlotsRepository
+ BmBooter
TABMntuTrack
TABMntuTrackTr
BM Tracks
TABMvieTrackFOOT
TABMparCon
TABMparGeo
TABMparMap
BmBooter

To do list

Tracking algorithms:

- Genfit is not fast as desired, it still has bugs and it is not easy to check what is happening. Moreover the Beam Monitor is not affected by the magnetic field, so a simple geometrical fit should be more suitable and fast than a Kalman fit.
- FIRST least square fitting restored (still under study).
- Other tracking algorithms are considered.
  (Alexopoulos et. al. Implementation of the Legendre Transform for track segment reconstruction in drift tube chambers, *Nuclear Instruments and method in physics, 592,3, 2008* https://doi.org/10.1016/j.nima.2008.04.038)
- The fastest (and reliable) tracking software can be implemented also in the acquisition software to provide an online track reconstruction during the data acquisition?

Beam Monitor space-time rel. calibration test @ Trento in December:

- The test will be performed with two external planes of stripped silicon detectors.
- Detector alignment algorithm (is there already something implemented somewhere in shoe?).
- Development of a macro to read the two datafiles and evaluate the st rel.
  (outside shoe ...probably)

To do list

Data Analysis for the test @ Trento in September:

- Evaluation of the efficiency, fit parameters etc.etc. is still ongoing.
  (some data will be presented during the collaboration meeting in December)

TABMactNtuMC:

- Modify the MC input data adding fake hits, cutting real hits etc., to obtain an input sample similar to a real data sample.
- This will help to study the tracking algorithm performances.

```
//The BM acquisition software structure
typedef struct BM_struct {

  Int_t evnum;                  //number of event (from global header)
  Int_t words;                  //total number of words
  Int_t tdcev;                  //total number of tdc events
  Int_t tdc_hitnum[MAXEVTDC];   //total number of hits of the tdc for each tdc events
  Int_t tdc_evnum[MAXEVTDC];    //tdc event number for each tdc events
  Int_t tdc_id[MAXHITTDC];      //measurement the tdc channel number
  Int_t tdc_meas[MAXHITTDC];    //measurement value (10^-8 sec.)
  Int_t tdc_sync[MAXHITTDC];    //time of the sync channel -10000=not set, (10^-8 sec.)
  Int_t sca830_meas[SCA830MAX]; //scaler measurement for each channel
  Int_t sca830_counts[SCA830MAX]; //scaler counts for each channel
  Int_t adc792_meas[ADC792MAX]; //ADC measurement for each channel
  Int_t adc792_over[ADC792MAX]; //ADC overflow cannel, -1000=not set, 1=overflow, 0=ok
  Int_t tot_status;             //global error flag: 0=ok
  Int_t tdc_status;             //tdc error flag: -10000=ok, 0=not set, 1=tdc wrong ev_num, 2=channel out of range, 3=no
  tdc data
  Int_t sca_status;             //scaler830 error flag: 0=ok
  Int_t adc_status;             //adc792 error flag: 0=ok, 1=wrong words num,
  Int_t time_evtoev;            //time between the previous and this event (microsec)
  Int_t time_read;              //time occurred to read the data (microsec, this time do not consider the vme acess time
  ~+10msec)
  Int_t time_acq;               //time of the event acquisition

} BM_struct;
```

BM_struct.h:

- It’s a struct → dummy way to read the events.
- Will be converted to a real class, the TAGdaqEvent that will contain all the subdetectors data.

Conclusion

- Time constraints: we have to prepare the December calibration test and the following data analysis.
- The implementation of the TABM libraries in the new shoe version can not start before January (not mandatory).
- The code status is at a good point, it should not need relevent changes or development for the March test.