Si Tracker Calibration studies on MC

The software framework

Why bother ?

Results and comments

Goals

- Use proton-proton Monte Carlo event
- Simulate the electronics effect (pedestals,noise, cm) as output from the FED to be realistic
- Use the ApvAnalysis framework for subtract/correct
- Use available calibration algorithms
- Evaluate the algorithms in proton-proton environment reliability, speed, memory required

The Software Framework

In the DataHandlingInterface package

SiStripDigitizerFromApvAnalysis : public SiStripDigitizer (T.Boccali)

It accesses MC data (no pedestals, no noise, no CM) for a given module

Digitized SimHits are passed to the DataHandlingApvAnalysisFactory's event reader and can be analysed in the ApvAnalysis framework

Reconstructs zero-suppressed digis within the ApvAnalysis framework

Pushes the digis in the Readout of the module

The Software Framework

SiStripGeneralDigitizerFromApvAnalysis : public SiStripDigitizer (N.M.)

Same basic functionalities as in Tommaso's example

+

- **The SiFedFullNoiseAdder is set in the SiStripGeneralDigitizerFromApvAnalysis constructor so that the SimHits are added with**
  - **gaussian noise (all strips not only those containing signals)**
  - **pedestals**

- **A GeneralDHApvAnalysisFactory is used, which inherits from VirtualApvAnalysisFactory (ApvAnalysisConcrete package)**

It provides the interface with the all concrete classes in ApvAnalysisConcrete and with the GeneralApvAnalysis class

so that calibration on MC data can be performed and any of the available ped,noise,cm calculators can be chosen via .orcarc file.

The Framework

The CalibrateApvAnalysis, also in the DataHandlingInterface package, takes care of calibration, check that the calibration is complete on each module
before sending data to zero-suppression

The GeneralDHApvAnalysisFactory contains also a method which allows to set pedestals and noise to a fixed constant
Useful for debugging purposes
Ped and Noise Calculators are set to Ready so when the Calibration is done only the Common Mode is calculated

The code is under CVS since before Xmas and documented 

.orcarc cards

The necessary lines are

TkSiStripDigitizer: DigitizerName = SiStripGeneralDigitizerFromApvAnalysis

ApvAnalysis:FactoryName = GeneralDHApvAnalysisFactory

GeneralDHApvAnalysisFactory:Calib = 1 Default. The calibration is fully done

GeneralDHApvAnalysisFactory:Calib = 0 Ped and Noise are Fixed

It has NO impact with other Factories

SiStripFedDigitizer:Pedestal = 400.

SiStripFedDigitizer:rmsCM = 100.

+

All the lines connected with GeneralApvAnalysis, in particular

GeneralApvAnalysisFactory:pedestalCalculator = calculatorName
GeneralApvAnalysisFactory:noiseCalculator = calculatorName
GeneralApvAnalysisFactory:commonModeCalculator = calculatorName
GeneralApvAnalysisFactory:maskAlgorithm = calculatorName
…… and related cuts

Why bother .....?

- Start to understand how to calibrate in 2007
  Common mode calculation and subtraction on FED FPGA +
  pedestals and noise to be measured online and stored in FED LUT

- Calibration during LHC inter-fill time for sure ..... no signals

- Almost likely need for calibration also during physics runs.
  FED can output virgin data (not zero-suppressed) at low rate.

  Private gaps would be great to use, since no signals **BUT** is the calibration fast enough ? Private orbits give longer time but collisions are there.

Why bother ……..?

- A number of calibration algorithms already developed and used in beam-test data analysis **BUT** …beam-test showed to be a very 'mild' environment (low occupancy, low/flat common mode noise etc.)
  Also ……on the beam, modules were not in the **REAL system configuration**

- What happens if occupancy is high (physics runs), CM is high (it can happen) ? First idea only from MC p-p events

- Present MC studies useful, of course NOT exhaustive and not final

  ✓ e.g. the CM is assumed to be flat (within a chip) but TOB rod system test recently showed that linear CM could be present. In the specific case the cause was understood and will be absent in the final system still …room for surprises …… (…… knocking on wood ……)

  ✓ e.g. whether re-calibrate periodically during physics or not depends on the stability of calibration constants. Experience has to come from the system

Why bother ..... ?

Bottom line .....

Whatever knowledge on this subject is acquired in the lab from growing-up system should be feed-back to the MC

"Preliminary" conclusions
From my previous talk

- In very benign conditions all the available algorithms give sensible results
- In slightly less benign conditions median CM (TT6 or R2) definitely better than the mean CM

  As for the pedestals more investigation is needed for large CM conditions. The present evidence seems to favor the median calculation.

- Updating frequency of pedestals and noise will depend on how the real data really look. The important thing is choosing the kind of calculation.
- The different implementations now available have pros and cons The final tools will result being the combination of bits and pieces from all the algorithm we have now.

Future work plans

From my previous talk

- **Understand better pedestals and noise** algorithms in High CM conditions and choose between the various options.

- **Superimpose pile-up events**
  It should have been 'today' work, unfortunately I kept getting a ☹ segmentation fault when using pile-up events
  Investigating ….

- **Revise** the determination of the CPU consumption

- A CMS note is in preparation

Work environment

- ORCA_6_3_0
- Use proton-proton collisions
  - a) single muons Pt>100GeV,
  - b) MB events + high lumy pileup
- Simulated pedestals: 400 ADC counts
- Simulated noise: 2500(ENC)*thickness/(0.03)
- Simulated rms CM: 5 ADC, 100 ADC
- Focus on the innermost TIB layers

Algorithms under test/evaluation

Pedestals

- ✓ Accumulating (median)
- ✓ Updating (running average)
- ✓ TT6 (truncated mean followed by updating)

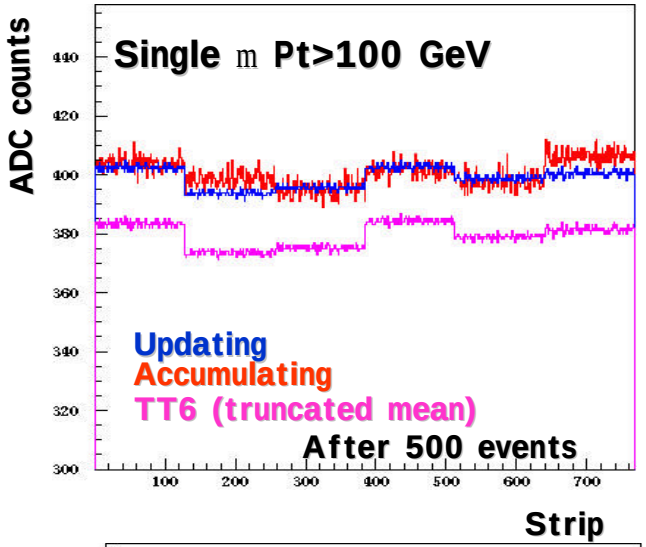
Noise

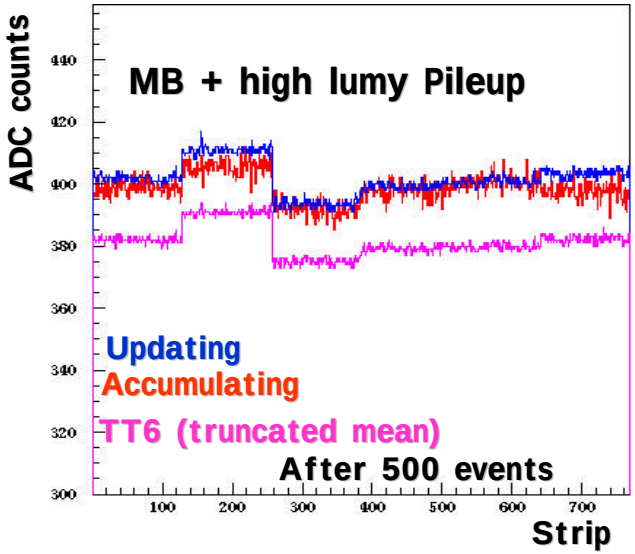
- ✓ Accumulating (median)
- ✓ Updating (running rms)
- ✓ TT6 (mean)

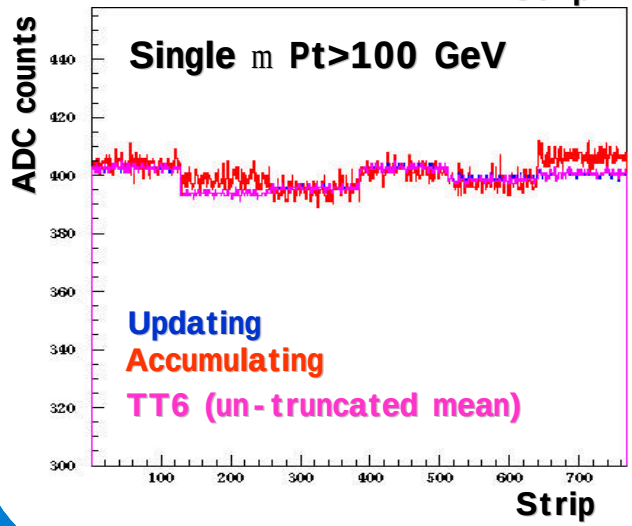
Common Mode Noise

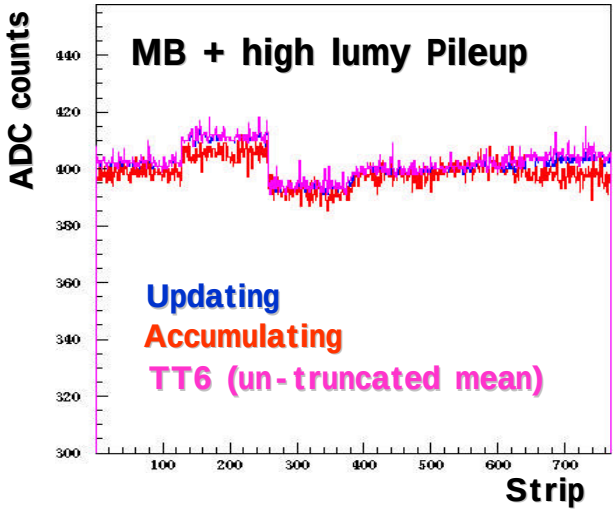
- ✓ MeanCM ® simple average over ped-subtracted data over the group of strips (128 by default) Only strips with no signal are used
- ✓ MedianCM
- ✓ R2CM ® Median
- ✓ MedianTT6
- ✓ MeanTT6

Pedestal calculators

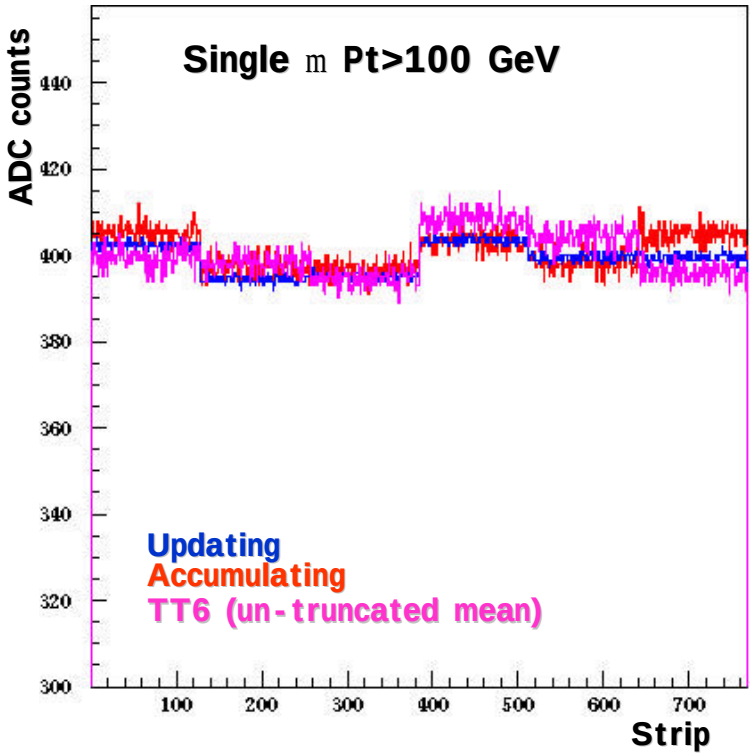


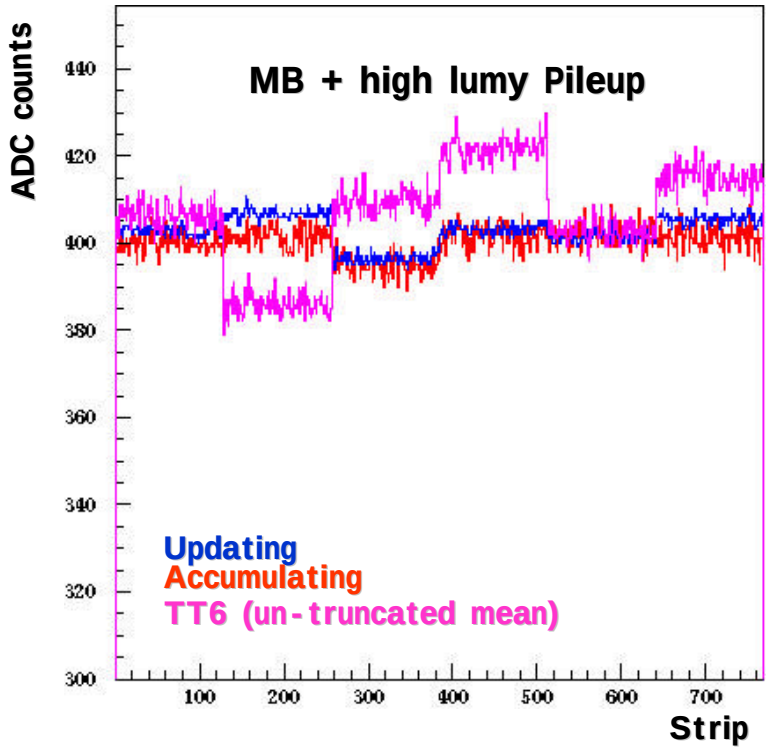






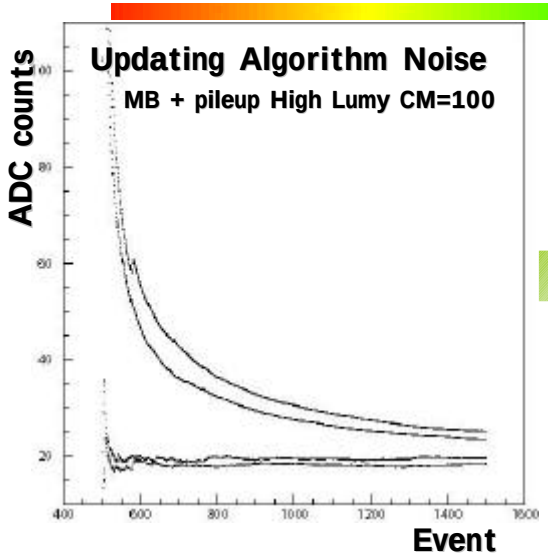
Pedestal calculators

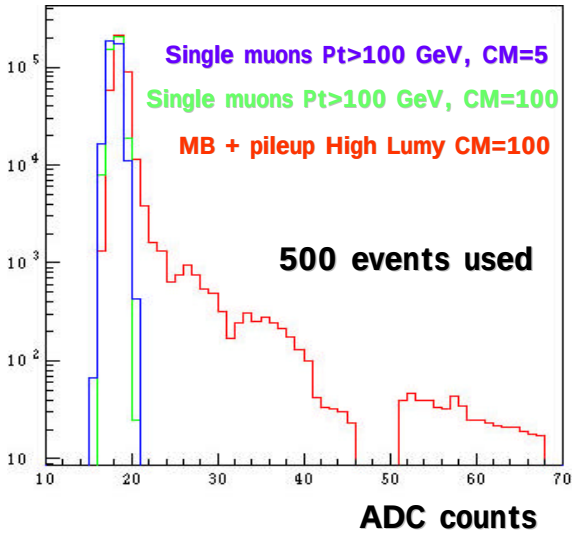




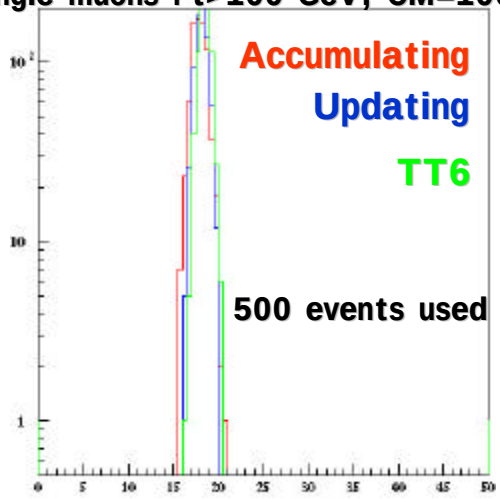
After 1000 events

Noise calculators

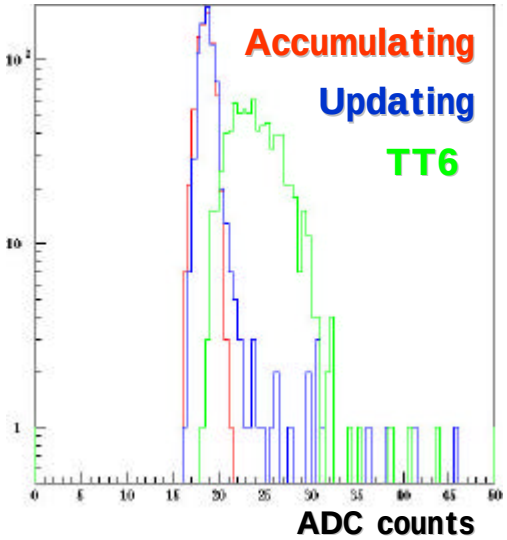




Single muons Pt>100 GeV, CM=100



MB + pileup High Lumy CM=100



Mask Algorithms

TBApvMask ::calculateMask

```
for (int i=0; i<in.size(); i++) {
        if (fabs ( noise[i]-meanrms )  > theTbad*rmsofrms ||
          noise[i] < MinNoise ||
          noise[i] > MaxNoise ||
         pedestal[i] < MinPedestal ||
         pedestal[i] > MaxPedestal) {
            theMask.push_back(dead); theNbad++;
            } else {
          theMask.push_back(ok);
      }
    }
}
```

Too many 'rigid' cuts

It easily fails with large CM

All strips are flagged 'bad'
and excluded by the CM calculation

CM = 0

Mask Algorithms

TT6ApvMask ::calculateMask

```
for (int i=0; i<in.size(); i++){
  if ( defineNoisy ( avVal,rmsVal,in[i] ) ) {
    theMask.push_back(noisy);
  } else if (in[i] < theDeadCut*avVal) {
        theMask.push_back(dead);
  } else {
    theMask.push_back(ok);
  }
}
```

Effective dynamical cuts

Distinction between NOISY and BAD

No drawback on CM calculation

Summary

- Pedestal and noise calculators developed so far are not, in general, adequate enough for conditions different from beam tests
- Pedestals with running mean or median equally satisfactory (Updating or Accumulating) algorithm for the same number of events TT6 first determination with truncated mean leads to a bias
- Noise with running rms (Updating) not adequate in high occupancy regime (MB+pileup). Accumulating and TT6 (median and mean) much better in harsh conditions
- Median common mode definitely preferred to mean common mode in high common mode regime
- CMS Internal note almost finished. Aim to send it out by the TRK week