EMC Fast Simulation Plans

Chih-hsiang Cheng
Caltech
SuperB FastSim Meeting, April 17, 2008

What do we want from EMC fast sim?

- I assume we don't want to build the whole EMC reco objects and go through the reco sequences.
- For each generated particle that reaches the EMC, we want to produce the responses fast and as detail/accurate as possible.
- For `ChargedTracks`, we want to attach the EMC responses to each (Bta)Candidate.
- For neutrals, we want to create the `CalorNeutral` list (do we want `CalorClusterNeutral`?)

Detector model

- Barrel: cylinder. Babar barrel.
- Forward endcap: a section of a cone. New crystals, but the overall geometry is basically the same as Babar endcap.
- Possible backward endcap: a disk or a section of a cone. New, unknown geometry yet.
- Geometry is simple enough to be specified with a few numbers (can be stored in text files in detector model package); no detail of crystals.
- Rolf Andreassen has started the code for Dirc detector model. I started using it as the template.
  - His model interface takes `PdtPid::PidType` for particle species, which only includes charged tracks. EMC needs neutrals too!

What do we want to generate?

- Do we want to build some sort of EmcCand (AbsRecoCalo) object, or is the info through BtaCalQual enough?

```
  void setPID( PdtPid::PidNeutralType x ) { _PID=x; }
  void setNBumps( unsigned x ) { _nBumps=x; }
  void setNCry( unsigned x ) { _nCry=x; }
  void setLat(  double x ) { _lat=x; }
  void setS9S25(  double x ) { _s9s25=x; }
  void setS1S9(  double x ) { _s1s9=x; }
  void setAbsZern42(  double x ) { _absZern42=x; }
  void setAbsZern31(  double x ) { _absZern31=x; }
  void setAbsZern20(  double x ) { _absZern20=x; }
  void setStatus(  int i ) { _status=i; }
  void setSecondMomentTP(  double x ) { _secondMomentTP=x; }
  void setERaw(  double x ) { _eRaw=x; }
  void setECal(  double x ) { _eCal=x; }
  void setCdPhi(  const Consistency& x ) { _cdPhi=x; }
  void setCentroid(  const HepPoint& x ) { _centroid=x; }
  void setCentroidX(  double x ) { _centroid.setX(x); }
  void setCentroidY(  double y ) { _centroid.setY(y); }
  void setCentroidZ(  double z ) { _centroid.setZ(z); }
  void setCovMat(  const HepSymMatrix& x ) { _covMat=x; }
  void setCovMatElem( unsigned i, unsigned j, double x ) { _covMat[i][j]
=x; }
```

How to simulate?

- Create a dictionary (lookup table) based on Babar full simulation for barrel and Geant4 simulation of new endcap (Perugia group?)
    - In Babar we can run BBbar and ccbar generic MC. For each long-lived particle (from MCtruth), we predict the entrance point to the EMC (for neutrals, based on true momentum, for charged particles, based on fit trajectory if available, or helix based on true momentum).

      Find the BtaCandidate that has the EmcCand closest to the entrance point.

      Record the particle species, momentum, location and angle at the entrance, and most of the CalQual's to an ntuple.

How to simulate? (cont.)

- For each bin of species/momentum/entrance angle, we store the distribution of each recorded variables, and a correlation matrix to a database.
  - Simplified version would be storing the mean and RMS, but many have very non-gaussian distributions. Probably need to store histograms?
- When generating, for each species/momentum/entrance angle, we throw random numbers and sample from those distributions, according to the correlation matrix.
- Change neutral's energy and momentum according to the sampling, and set CalQuals for each particle.

Next steps

- Write a beta application to collect EMC responses in a TTree.
- Design proper binnings and store the distributions to a root file.
- Design the dictionary that looks up the root file and does random sampling.
- Implement the detector model to be used in PravdaMC.
- ...