February 27, 2002
E. Perez (eperez@hep.saclay.cea.fr)

Simulation of the Performances of Jet Trigger Algorithms

1 Parameters of the Algorithm

The algorithm is defined by the following parameters:

- the size in (η, ϕ) of the regions of interest (RoI's). The corresponding RCP parameters are `eta_size_roi` and `phi_size_roi`, given in units of 0.2 for both directions.
- the steps in (η, ϕ) by which the candidate RoI's are sliding. RCP parameters are `eta_step_roi` and `phi_step_roi`. They should be ≥ 1. One sets `eta_step_roi` and `phi_step_roi` to 1 to define RoI's sliding by steps of 0.2 in η and ϕ. The case of non-overlapping RoI's is obtained with `eta_step_roi = eta_size_roi` and `phi_step_roi = phi_size_roi`.
- the minimal distance in (η, ϕ) between two local maxima RoI's is defined by the RCP parameters `eta_sep` and `phi_sep` (in units of 0.2). These can be zero, positive, or negative. Eventually, two local maxima may overlap when `eta_sep` or `phi_sep` is set to a negative value. To ensure that two local maxima RoI's do not overlap `eta_sep` and `phi_sep` should be ≥ 0. Two example of local maxima RoI's are shown in Fig. 1.

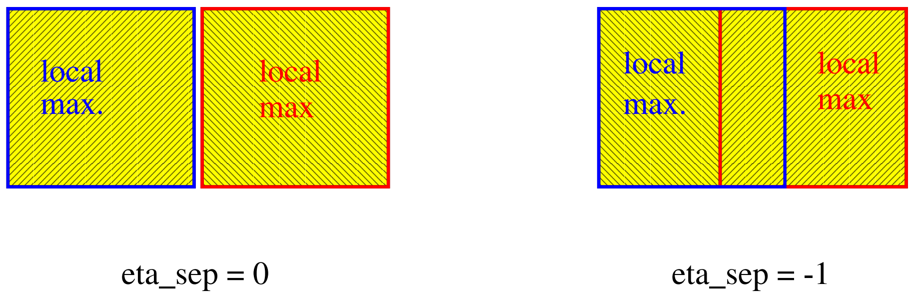

Figure 1: Two example definitions of the local maxima RoI's.

- the parameters `jet_eta_right`, `jet_phi_up`, `jet_eta_left` and `jet_phi_down` control how the (maxima) RoI's are extended to define trigger jets, as shown in Fig. 2. These are given in units of 0.2 and should be ≥ 0.
- only trigger jets with a transverse momentum above a given cut ($P_T >$ `jet_pt_cut`), and in a given range in rapidity ($\mid \eta \mid <$ `jet_eta_cut`) are taken into account.

The algorithm returns the energy and position of the trigger jets found (η and ϕ correspond to the geometric centre of the jet area). Moreover a calculation of the missing energy as seen by the trigger is also performed, by just summing the P_x and P_y of the input 0.2 x 0.2 trigger towers.

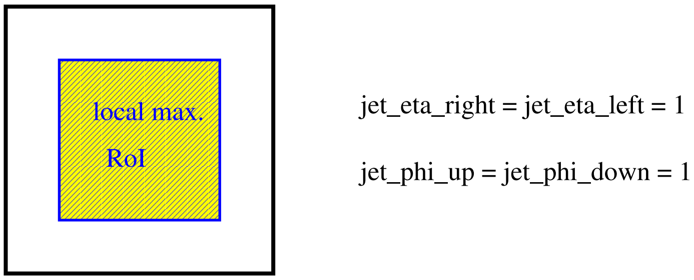

Figure 2: Definition of trigger jets.

2 The simulated trigger towers

The "building blocks" of the RoI's and trigger jets are 0.2 x 0.2 trigger towers which can be chosen to be:

- the standard TT as calculated by `tsim_l1cal` (RCP parameter `TTbase` = 1).
- or (`TTbase` = 2) TT's recalculated from the precision readout cells (excluding ICD, massless gaps and coarse hadronic). That's quite similar to what is done in `tsim_l1cal`, but allows the user to choose the amount of smearing to be applied to the precision readout energies (RCP parameter `Smear`), as well as the rounding/truncation procedure for the TT energies: the parameters `round_em`, `round_had` and `round_tot` define whether energies are rounded or truncated, and `em_step`, `had_step` and `tot_step` define the digitization steps. For example, with `round_em = 1` and `em_step = 0.25`, an EM energy of 5.2 GeV will be digitized to 5.25 GeV, while the result would be 5.0 GeV with `round_em = 0`.

For `TTbase` = 1: TT energies are obtained by smearing the precision readout cells by 10% (`tsim_l1cal/L1CalData.cpp`). In release versions before `p11.01`, the definition of TT's in `tsim_l1cal` corresponds to `round_em = round_tot = 0`, `em_step = 0.25` and `tot_step = 0.5`. In most recent versions, it corresponds to `em_step = 0.25` and `tot_step = 0.25`.
In release versions older than `p10`, the TT's do not contain the signals coming from ICD and massless gaps. If the sliding algorithm code is used with a version $\geq$ `p10`, ICD and MG energies are taken into account in the calculation of the trigger $E_{T,miss}$, but not in the trigger jet algorithms.

For `TTbase` = 2: The ICD and massless gaps energies are not taken into account, neither in the definition of jets, neither in the calculation of missing energy.

Note: if the package is run several times with different steerings, the TT's are recalculated each time: for a given event, the TT's for steering A and steering B will not be the same due to the energy smearing. *(to be modified soon to facilitate comparisons on an event by event basis.)*

3 Output information

Used with the ntuple manager, the package can create various blocks in the root-tuple:

- the blocks CalTT, RoI, MaxRoI and TTjets, which contain information on trigger towers, RoI's, maxima RoI's, trigger jets. Each block contains: the number of corresponding objects Ntt; for each object, the total, electromagnetic and hadronic transverse energies (et[i], etem[i], ethad[i]), and its position (eta[i] and phi[i]). eta[i] and phi[i] are the η and ϕ of the centre of the area.
- the block MET contains the total P_x and P_y calculated by suming energy deposits in all TT's (Px and Py), as well as the sums obtained when suming on electromagnetic TT's only (PxEM and PyEM).

The RCP parameters iTT, iRoI, iMaxRoI, iTTjets and iMET determine which information will be written to the root-tuple. The root-tuple blocks are prefixed by a string which labels the algorithm (algo in the RCP file). This allows to produce one root-tuple with the output from several steerings for the sliding window algorithm.

4 Code installation

The package is not yet in CVS. The sources can be found on d0mino in

```
/home/perez/R2B_TRIGGER/simul_trigger/trigger_calo
```

To install the code:

```
setenv STUFF /home/perez/R2B_TRIGGER/simul_trigger/trigger_calo
setup d0cvs
newrel -t t02.05.00 simul_trigger
cd simul_trigger
ctnewpkg -l trigger_calo
cd include
ln -s ../trigger_calo/trigger_calo/ trigger_calo
cd ..
cd trigger_calo
cp $STUFF/LIBDEPS .
cp $STUFF/SUBDIRS .
cp $STUFF/src/* src/.
cp $STUFF/trigger_calo/* trigger_calo/.
mkdir bin
mkdir rcp
cp $STUFF/rcp/*.rcp rcp/.
cp $STUFF/bin/* bin/.
```

This has been tested on d0mino using release t02.05.00. Note that with older test versions, or with a production version, you may have to modify TriggerCalo.cpp to be consistent with the L1CalData constructor in tsim_l1cal/L1CalData.cpp : e.g. replace in TriggerCalo :: TriggerCalo (Context * context) :

```
_l1_caldata = new L1CalData(smear,current_hardware,tot_size);
```

by:

```
_l1_caldata = new L1CalData(smear);
```

5 First checks & control plots

Several algorithms have been run on a sample of QCD events, together with the offline Run2 cone algorithm (JCCB) with a radius $R = 0.5$. The position and energy of the highest p_T jet found at the trigger level are compared with those of the highest p_T reco'd jet in Fig. 3 and Fig. 4. The trigger algorithms used TT's from `tsim_l1cal` and are labelled `TT_x_y_z` with:
`x = eta_size_roi = phi_size_roi`,
`y = eta_sep = phi_sep`,
`z = jet_eta_right = jet_eta_left = jet_phi_down = jet_phi_up`.
`jet_pt_cut` and `jet_eta_cut` were set to 5 GeV and 2.5, respectively.

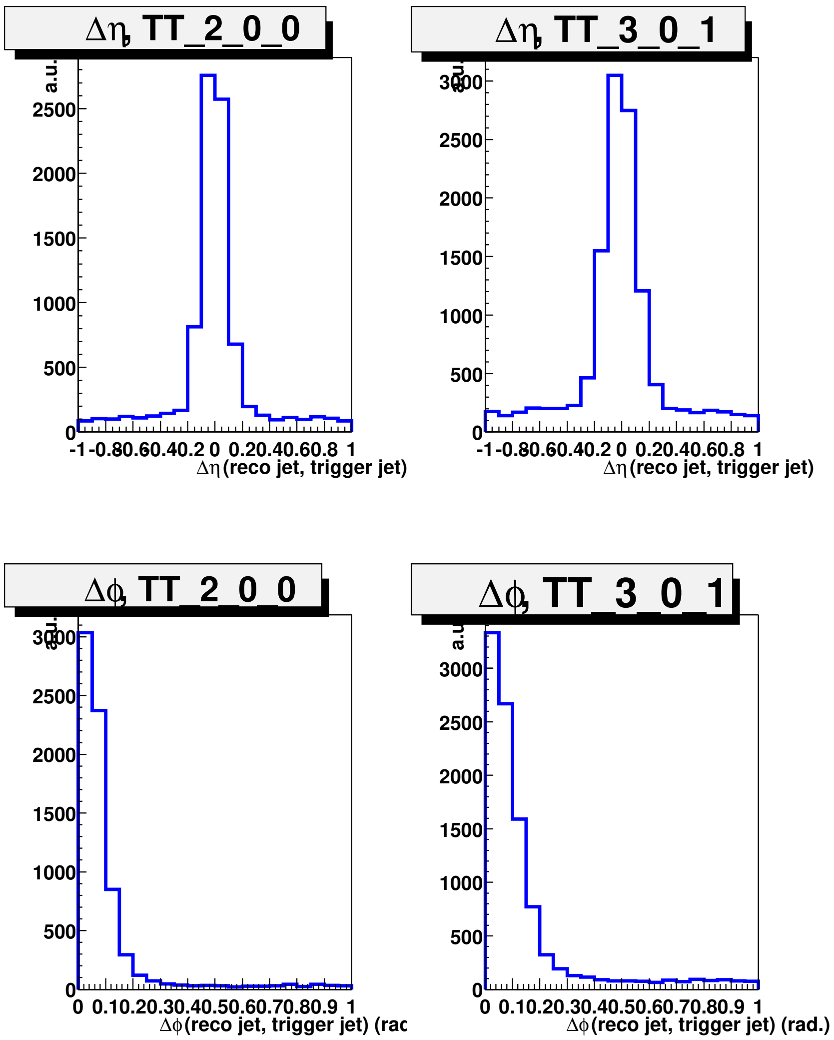

Figure 3: η and ϕ of the highest p_T trigger jet, w.r.t. that of the highest reco'd jet.

In Fig. 5, the number of jets with $p_T > p_{T,cut}$ found in a sample of QCD events is compared for algorithms TT_3_0_1 (which requires to know the energies of 7×7 TT's to treat one TT) and TT_3_minus1_1 (which requires 5×5 TT's to treat one TT). The number of reco'd jets with $p_T > p_{T,cut}$ is also shown (illustrative only, since the TT jets do not carry 100% of the reconstructed energy).

Appendix: Example RCP file

```
string PackageName = "TriggerCalo"


int debug = 0

int TTbase = 2  // TTbase =1: use TT from tsim_l1cal
                // TTbase =2: rebuild TT's from CalCells

RCP tsiml1cal = <trigger_calo tsim_l1cal>

// String to define algorithm
string algo="TT_3_0_1_"

// ----- Definition of RoI's :
int eta_size_roi = 3
int phi_size_roi = 3

int eta_step_roi = 1    // defines the overlaps
int phi_step_roi = 1


// ----- Declustering of RoI's  (definition of local maxima) :
int eta_sep = 0
int phi_sep = 0

// ----- Definition of trigger jets:
int jet_eta_right = 1
int jet_phi_up    = 1
int jet_eta_left  = 1
int jet_phi_down  = 1

// ----- Cuts on eta and Pt for trigger jets:
float jet_eta_cut = 2.5
float jet_pt_cut = 5.


// ----- Variables in the root-tuple
int iTT = 1       // 0.2 x 0.2 Trigger Towers
int iRoI = 0      // RoI's
int iMaxRoI = 0   // local maxima
int iTTjets = 1   // trigger jets
int iMET = 1      // missing Et from 0.2 x 0.2 TT's
```

```
// -----------------------------------------------
//
// ----- If TTbase =2:
// ----- Rebuild Calo TT's from CalCells:

int debug_Cal_Dig = 0

int Xcal_dig    =1 // Calo digi : 0 none, 1 CC and EC, 2 CC only,3 EC only
float Emin_Cal_dig= 0.0
float Emin_Cal_Tower = 0.0
float Smear = 0.1   // smearing of CalCells energies;
                   // if negative, no smearing

   // Rounding (1) or truncation (0) of TT energies:
int round_em = 1
int round_had = 1
int round_tot = 0

  // energy step in GeV for the rounding / truncation
float em_step = 0.25
float had_step = 0.25
float tot_step = 0.5

  // .... old stuff ....
int Calo_weight=0 //if =1 , apply the layer weights below

// -----------------------------------------------
```

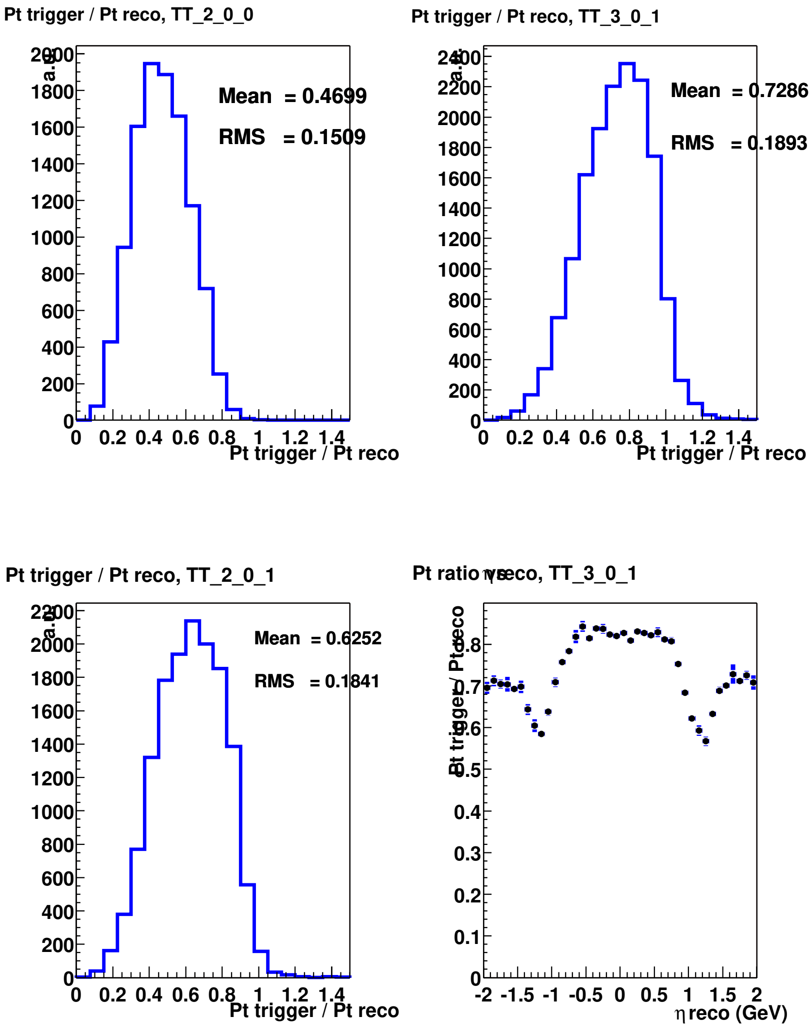

Figure 4: p_T of the highest p_T trigger jet, w.r.t. that of the highest reco'd jet.

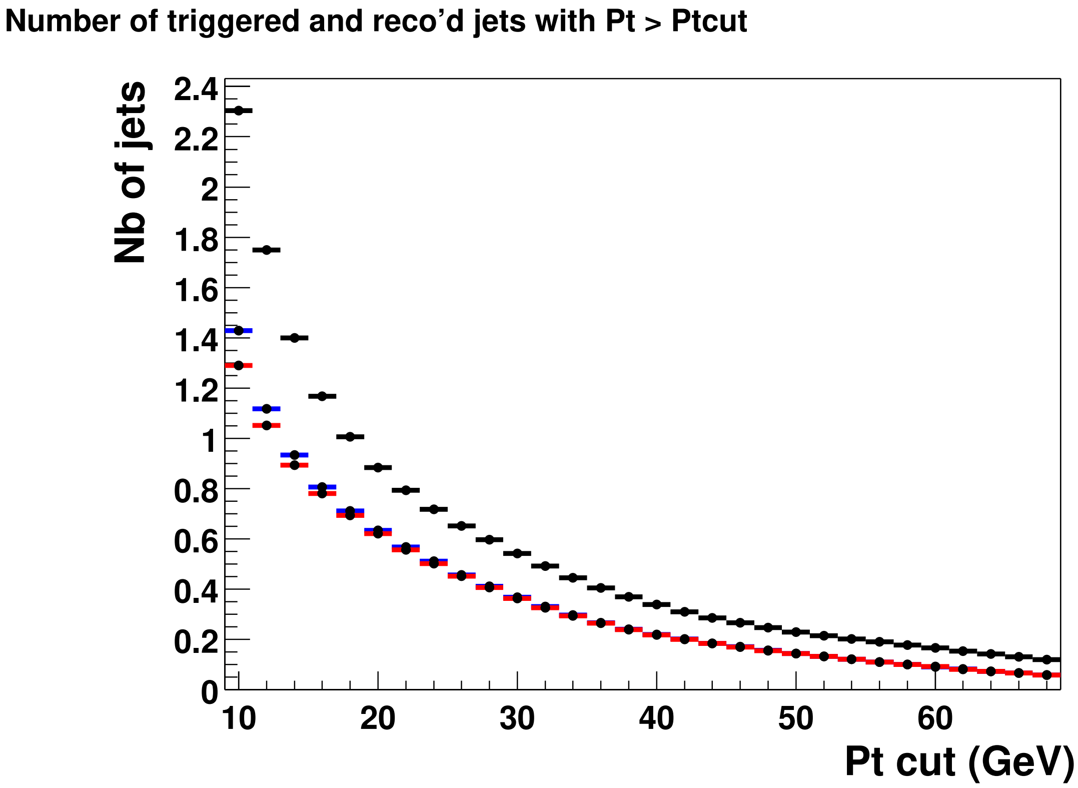

Figure 5: Number of jets above $p_{T,cut}$ found with TT-3-0-1 (lower histogram) and with TT-3-minus1-1 (middle histogram). The number of reco'd jets is shown as the upper histogram.