**CMS TN/94-174**
**5 December, 1994**

# Simulation of the CMS Muon Trigger System

**C.-E. Wulz**
HEPHY, Österreichische Akademie der Wissenschaften, Vienna, Austria

**Á. Csilling**
Department of Atomic Physics, Roland Eötvös University, Budapest, Hungary

**J. Gajewski[1)], M. Konecki[1)] and J. Królikowski[1)]**
Institute of Experimental Physics, Warsaw University, Poland

**M. Górski**
Institute for Nuclear Studies, Warsaw, Poland

**G. Wrochna[2)]**
CERN, Geneva, Switzerland

**Abstract**

**Program MTRIG for the RPC Muon Trigger simulation is described. The digitisation procedure, the noise generation and the Pattern Comparator Trigger (PACT) algorithm are briefly explained. The note contains the flow chart of the program, description of the control cards, the structure of BOS banks, and the list of available simulated data.**

## 1 Introduction

This paper is intended to provide some technical information about the software related to the muon trigger study. Description of the trigger system itself and more information about algorithms one can find in [1, 2, 3, 4]. The CMS software in general is described in [5]. Some ideas about what kind of questions should be answered by the muon trigger simulation and how to unify simulation of different subsystems are presented in [6].

The full chain of the muon trigger simulation is shown in Fig. 1. It consist of four basic steps:

| | | |
|---|---|---|
| **generation** | **ISAJET, PYTHIA** | produces particles at the vertex, |
| **simulation** | **CMSIM / GEANT** | passes particles through the detector, |
| **digitization** | **MTRIG (IDIGFL>0)** | gives detector answer, |
| **analysis** | **MTRIG (IPATFL>0)** | prepares trigger algorithm, |
| | **MTRIG (IRATEFL>0)** | calculates efficiency curves and rates |
| | **MTRIG (IUSERFL>0)** | any user defined analysis. |

In this paper we concentrate mainly on the program **MTRIG**, however, for completeness, we also describe the structure of all related banks and give a list of available data.

[1)] Supported by Polish Committee for Scientific Research under grants KBN-PB-2 0422 91 01, KBN-SPUB-206/93 and KBN-SPUB/P3/201/94.
[2)] On leave of absence from Institute of Experimental Physics, Warsaw University, Poland.

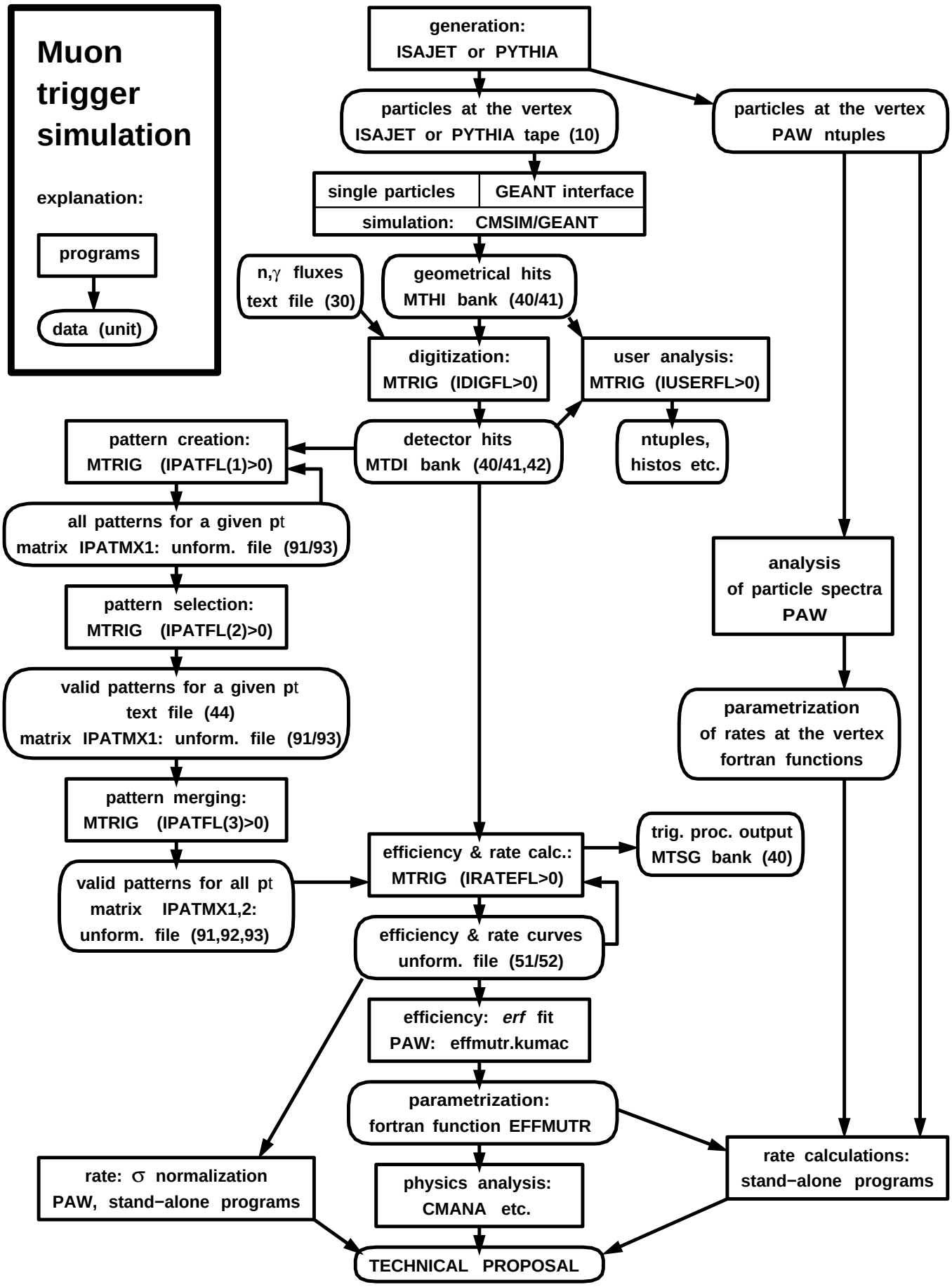


Figure 1: Muon trigger analysis chain

## 2 Bank structure

In order to describes the structure of banks used in the trigger study let us keep the following convention:

- For two banks **xxxx** and **yyyy** the following relations are possible:

  **xxxx = yyyy** parallel
  **xxxx – yyyy** one to one
  **xxxx** $<$ **yyyy** one to many
  **xxxx** $>$ **yyyy** many to one

- For the relation **xxxx** $<$ **yyyy** three indices are needed:

  **Nyyyy** – number of rows in **yyyy** related to this **xxxx**
  **Iyyyy** – first row in **yyyy** related to this **xxxx**
  **Jxxxx** – next row in **yyyy** related to the same **xxxx**

- For **xxxx** – **yyyy** or **xxxx** $>$ **yyyy** only one is enough:

  **Kyyyy** – corresponding row in **yyyy**

Using this convention the following banks are designed:

| **MTBX – $\mu$ trig. bunch crossing (b.x.) info** | | | |
|---|---|---|---|
| column position | type | name | function |
| 1 | I | NMTEV | number of pp interactions in this b.x. |
| 2 | I | IMTEV | first pp interactions (<MTEV) |

| **MTEV – $\mu$ trig. event (single pp interaction) info** | | | |
|---|---|---|---|
| column position | type | name | function |
| 1 | I | KMTBX | pointer to the b.x. (>MTBX) |
| 2 | I | JMTBX | next row related to the same b.x. (MTBX<) |
| 3 | I | NMTVX | number of particles at the vertex |
| 4 | I | IMTVX | first particle (<MTVX) |

| **MTVX – $\mu$ trig. bank for particle at the vertex** | | | |
|---|---|---|---|
| column position | type | name | function |
| 1 | I | IPVEX | particle type (GEANT convention) |
| 2 | F | ETAVEX | $\eta$ of particle at the vertex |
| 3 | F | PTVEX | $p_t$ of particle at the vertex |
| 4 | F | PHIVEX | $\varphi$ of particle at the vertex (radians) |
| 5 | I | KMTEV | pointer to the event (>MTEV) |
| 6 | I | JMTEV | next row related to the same event (MTEV<) |
| 7 | I | NMTHI | number of hits by this particle |
| 8 | I | IMTHI | first hit (<MTHI) |

| **MTHI – geometrical hit in $\mu$ trig. station - OLD VERSION** | | | |
|---|---|---|---|
| column position | type | name | function |
| 1 | I | IPHIT | particle type (GEANT convention) |
| 2 | I | MSTHIT | $\mu$ station type (MB/MS/MF = $-1/0/1$) |
| 3 | I | MSNHIT | $\mu$ station number 1-5 ($+/-$ = in/out) |
| 4 | F | THIT | time of flight from the vertex |
| 5 | F | RHIT | hit $r$ coordinate |
| 6 | F | PHIHIT | hit $\varphi$ coordinate (radians) |
| 7 | F | ZHIT | hit $z$ coordinate |
| 8-10 | F | PHIT(1-3) | $p_x, p_y, p_z$ of the hitting particle |
| 11 | I | KMTVX | pointer to the particle (>MTVX) |
| 12 | I | JMTVX | next row related to the same particle (MTVX<) |

| **MTHI – geometrical hit in $\mu$ trig. station** | | | |
|---|---|---|---|
| column position | type | name | function |
| 1 | I | IPHIT | particle type (GEANT convention) |
| 2 | I | MSTHIT | $\mu$ station type (MB/MS/MF = $-1/0/1$) |
| 3 | I | MSNHIT | $\mu$ station number 1-5 ($+/-$ = in/out) |
| 4 | F | THIT | time of flight from the vertex |
| 5 | F | RHIT | hit $r$ coordinate |
| 6 | F | PHIHIT | hit $\varphi$ coordinate (radians) |
| 7 | F | ZHIT | hit $z$ coordinate |
| 8 | F | ETAHIT | $\eta$ of the hit |
| 9-11 | F | PHIT(1-3) | $p_x, p_y, p_z$ of the hitting particle |
| 12 | I | KMTVX | pointer to the particle (>MTVX) |
| 13 | I | JMTVX | next row related to the same particle (MTVX<) |
| 14 | I | NMTDI | no. of hit strips in MTDI for this hit |
| 15 | I | IMTDI | pointer to the 1st hit in MTDI (<MTDI) |

| **MTDI – digitized hit in $\mu$ trig. station** | | | |
|---|---|---|---|
| column position | type | name | function |
| 1 | I | IPHISC | $\varphi$ sector (1-12) |
| 2 | I | IPHISG | $\varphi$ segment in the sector (1-15) |
| 3 | I | ISTRIP | strip no. in a segment (1-6) |
| 4 | I | IETASC | $\eta$ sector (1-4) |
| 5 | I | IETASG | $\eta$ segment $\pm$ ($-20$ - $+20$) |
| 6 | I | MSTHIT | $\mu$ station type (MB/MS/MF=$-1/0/1$) |
| 7 | I | MSNHIT | $\mu$ station number 1-5 (in/out = $+/-$) |
| 8 | F | THIT | TOF= THIT(MTHI)+propagation time to Trig.Proc. |
| 9 | F | QSTRIP | charge on a strip (in pC) (presently 1.0) |
| 10 | I | KMTHI | pointer to the hit in MTHI (>MTHI) |
| 11 | I | JMTHI | next dig.hit for the same geom.hit (MTHI<) |

| **MTSG – segment trigger answer** | | | |
|---|---|---|---|
| column position | type | name | function |
| 1 | I | IPHISC | $\varphi$ sector (1-12) |
| 2 | I | IPHISG | $\varphi$ segment in the sector (1-15) |
| 3 | I | IETASC | $\eta$ sector (1-4) |
| 4 | I | IETASG | $\eta$ segment $\pm$ ($-20$ - $+20$) |
| 5 | I | KMTDI1 | pointer to the strip in st.1 (>MTDI) |
| 6 | I | KMTDI2 | pointer to the strip in st.2 (>MTDI) |
| 7 | I | KMTDI3 | pointer to the strip in st.3 (>MTDI) |
| 8 | I | KMTDI4 | pointer to the strip in st.4 (>MTDI) |
| 9 | I | IPTCUT | index of the highest $p_t^{cut}$ observed |

The GENE, GPHY, GCUT and CHED banks [5] are also kept, but CHED is filled once per run (not per event). Other standard CMSIM banks (like CKIN, CSEC) are not needed and therefore not written to tape. Bank MTBX is not created and first two columns of MTEV are not filled at the simulation time. They are reserved for superposition of several pp interaction into a whole bunch crossing. Bank MTDI is created during digitization, bank MTSG – during trigger analysis.

# 3 Available simulated data

In order to develop trigger algorithms a large number of single muon events have been simulated. The calculations have been done for the CMS version 9, presented in the *CMS Status Report* '94 [7]. The simulated data are grouped in files corresponding to given $p_t$ values and $\eta$ intervals. Each file contains 100 000 events. The data are in form of BOS banks, described above. The MTHI bank is in the old format, but the routine **crmtne** provides the conversion to the new one. The data sets already prepared are listed in Tab. 1. Question marks indicate data sets practically useless because of very low probability that a particle reach any muon station.

| $p_t$ | | $\eta$ range | | | | |
|---|---|---|---|---|---|---|
| | | a | b | c | d | IBM 800 MB |
| index | GeV | 0.0-0.8 | 0.8-1.5 | 1.5-2.0 | 2.0-2.5 | cartridge |
| 1 | 0.5 | | | | ? 1 (52) | |
| 2 | 1.0 | | | ? 2 (32) | ? 3 (53) | |
| 3 | 1.5 | | | ? 4 (33) | 5 (54) | I14202 |
| 4 | 2.0 | | 6 (14) | 7 (34) | 8 (55) | |
| 5 | 2.5 | | 9 (15) | 10 (35) | 11 (56) | |
| 6 | 3.0 | 12 (1) | 13 (16) | 14 (36) | 15 (57) | |
| 7 | 3.5 | 1 (2) | 2 (17) | 3 (37) | 4 (58) | |
| | 3.6 | 5 (3) | | | | I14203 |
| | 3.8 | 6 (4) | | | | |
| 8 | 4.0 | 7 (5) | 8 (18) | 9 (38) | 10 (59) | |
| | 4.2 | 11 (6) | | | | |
| 9 | 4.5 | 12 (7) | 13 (19) | 14 (39) | 15 (60) | |
| 10 | 5.0 | 1 (8) | 2 (20) | 3 (40) | 4 (61) | |
| 11 | 5.5 | 5 (9) | 6 (21) | 7 (41) | 8 (62) | I14204 |
| 12 | 6.0 | 9 (10) | 10 (22) | 11 (42) | 12 (63) | |
| 13 | 7.0 | 1 (11) | 2 (23) | 3 (43) | 4 (64) | |
| 14 | 10.0 | 5 (12) | 6 (24) | 7 (44) | 8 (65) | I14205 |
| 15 | 15.0 | 9 (111) | 10 (25) | 11 (45) | 12 (66) | |
| 16 | 20.0 | 1 | 2 (26) | 3 (46) | 4 (67) | |
| 17 | 30.0 | 5 (13) | 6 (27) | 7 (47) | 8 (68) | I14206 |
| 18 | 50.0 | 9 | 10 (28) | 11 (48) | 12 (69) | |
| 19 | 70.0 | 1 | 2 (29) | 3 (49) | 4 (70) | |
| 20 | 100.0 | 5 | 6 (30) | 7 (50) | 8 (71) | I14207 |
| 21 | 150.0 | 9 | 10 (31) | 11 (51) | 12 (72) | |

Table 1: Generated single muon data.

The files are stored on IBM cartridges, and on the exabyte **SA0030** at the CERN tape vault. Entries in the table indicate file numbers on the IBM cartridges listed in the last column. Numbers in parenthesis correspond to file numbers on the exabyte type. Each file is named **cmXXXXY.bos**, where **XXXX** stands for $p_t$ in 0.1 GeV units, and **Y**=a,b,c,d indicates one of the four rapidity regions (see table). For example the file **cm0025c.bos** contains muon generated with $p_t$ = 2.5 GeV and 1.5< $|\eta|$ <2.0 .

# 4 The MTRIG program

## 4.1 Organization of the program

The flow of the program is symbolically shown in Fig. 2.

| | | | |
|---|---|---|---|
| **mtffgo** | | | FF cards handling |
| **mtinchk** | | | check consistency of input cards |
| **mtpatin** → | **mtpator** | IPATFL(2)>0 and LEVELOR>0 | pattern handling initialization |
| **mtratin** | IRATEFL>0 | | initialization of eff. and rate calculations |
| **mtdini** | | | geometry definition for digitization |
| **mtinih** | IUSERFL≤0 | | book histograms |
| IF IPATFL(1)>0 or IRATEFL> 2 | | | |
| **begrun** | | | read/write header banks |
| DO | | | loop over bunch crossings |
| DO | | | loop over pp collisions |
| **readbos** | | | read an event |
| **crmtne** | | | convert old format MTHI into new one |
| **crebank** | | | create MTDI bank |
| **mtdiev** → | | | digitization |
| | **mteta** | | find segment/sector in $\eta$ |
| | **mtime** | | calculate flight+propagation time |
| | **mtclsz** | | generate cluster size |
| | **mtphi** | | find segment/sector in $\varphi$ |
| **mtnoise** → | INOISFL>0 | | n,$\gamma$ and electronics noise generation |
| ENDDO | | | |
| **mtpatev** → | IPATFL(1)>0 | | search for patterns, loop over MTDI |
| **fillpat** | | | store the pattern |
| **mtratev** → | | | apply low and high $p_t$ algorithms |
| | **mtpatev** → | IRATFL>0 | search for patterns, loop over MTDI |
| | | **comppat** | compare the pattern with a predefined set |
| **mtfillh** | IUSERFL≤0 | | fill histograms |
| **writebos** | | | write BOS banks |
| **cleanbos** | | | release the memory |
| ENDDO | | | |
| ENDIF | | | |
| **mtpaten** | | | select/merge valid patterns |
| **mteffen** | IRATEFL=1,2 | | finish eff. and rate estimations |
| **mtraten** | IRATEFL>2 | | finish eff. and rate calculations |
| **mtputh** → | | | end of run histograms |
| | **histdo** | | print histograms |
| | **hrput** | | store histograms |

Figure 2: MTRIG flow chart.

The trigger simulation is presently designed to handle the following tasks:

| action | switch | input | output | script |
|---|---|---|---|---|
| digitization of hits in RPCs | IDIGFL=1,2 | MTHI (41) | MTDI (40) | mtpat1 |
| creation of hit patterns | IPATFL(1)>0 | MTHI (41), IPATMX1 (91) | text (60), IPATMX1 (93) | mtpat1 |
| selection of valid patterns | IPATFL(2)>0 | IPATMX1 (91) | IPATMX1 (93), text (60) | mtpat23 |
| merging of valid patterns | IPATFL(3)>0 | IPATMX1,2 (91,92) | IPATMX1 (93) | mtpat23 |
| efficiency and rate estimations | IRATEFL=1 | IPATMX1,2 (91,92), NEFF (51) | NEFF (52) | mtrat1 |
| efficiency calculations | IRATEFL=2 | MTHI (41), NEFF (51) IPATMX1,2 (91,92) | NEFF (52) text (60), MTSG (40) | mtrat2 |
| rate calculations | IRATEFL>2 | MTHI (41,42), NTACC1,2 (51) IPATMX1,2 (91,92) | NTACC1,2 (52) text (60), MTSG (40) | mtrat3 |
| any user defined analysis | IUSERFL>0 | | | |

1. **Input / output**

   Input data set is on unit 41, the name of this set should be given in the data cards. It is assumed that it is a BOS file containing muon trigger banks MTEV, MTVX, MTHI, (possibly MTBX,) and general banks GENE, GPHY, GCUT and CHED as described in the CMSIM manual (CMS TN/93-63). The file can contain results of the simulation of single particles as well as full pp interactions. At the moment the BOS banks are written with fortran format. It is foreseen to use EPIO as soon as it works properly on UNIX.

   In addition to the standard output there are the following optional output files:

   - Output data set unit 40, the name of which has to be given in the data cards.

     This file contains old banks, copied from the input file, and all new banks created in the program.
   - PAW file for histograms, the name of which has to be given in the data cards.

2. **Digitization of the hits in the trigger chambers.**

   This is done by the MTDIEV routine, the enabling flag is IDIGFL:

| IDIGFL | action |
|---|---|
| 0 | no digitization |
| 1 | 'MTDI' created with cluster size = 1 strip |
| 2 | full digitization – cluster size generated by the MTCLSZ routine |

   The results are placed in the 'MTDI' bank described in the previous section.

   **Digitization geometry**

   To start this task the initialization of the trigger chambers' geometry is done in **MTDINI**. The definitions of the sectors, segments and strips in $\varphi$, as well as sectors and segments in $\eta$ are, at present, given in data statements in **MTDINI**, and in the utility routines in patch **UTIL**. They conform to the foreseen structure of the trigger processor organization described in [4] and shown in Fig. 3.

   The early conversion to the data base structure is clearly recommended before more complicated trigger geometry is introduced.

   Present version assumes uniform angular strip width of 1/3 deg. It means that the cluster size in terms of strips is largest at low radii of the hit, and smallest at high radii. The table below gives a few examples.

| muon station | radius | strip width |
|---|---|---|
| MF1, $\|\eta\|$=2.5 | 100 cm | 0.6 cm |
| MS1 | 385 cm | 2.2 cm |
| MS4 | 695 cm | 4.0 cm |

   **Efficiency of RPC**

   Only a (random) fraction of hits is digitized, according to RPC efficiency for MIPs, neutrons and gammas defined by control card **RPC**. In the present version there are no geometrical inefficiencies.

   **Noise generation**

   Expected neutron and gamma fluxes [8] are read from unit 30 and interpolated for each $\eta$ interval. RPC efficiency as defined by control card **RPC** is then applied in order to calculate average number of hits. Electronics noise and intrinsic RPC noise are simulated by random generation of hits with constant frequency per strip and per cm$^2$ respectively as defined by control card **RPC**. Actual number of hits is generated with Poisson distribution and then hits are randomly distributed in $\varphi$.

   **Cluster size generation**

   For $|\eta| \leq 2$ – where the rates allow the use of the RPC's – the cluster size, for **IDIGFL=2**, is generated according to the parametrization of the Warsaw RPC test data from June '93. The cluster size is then expressed in number of strips.

   For $|\eta| > 2$ – where the high rates will force us to use a different technique – the cluster size is presently set to 1 strip irrespective of the value of **IDIGFL**.

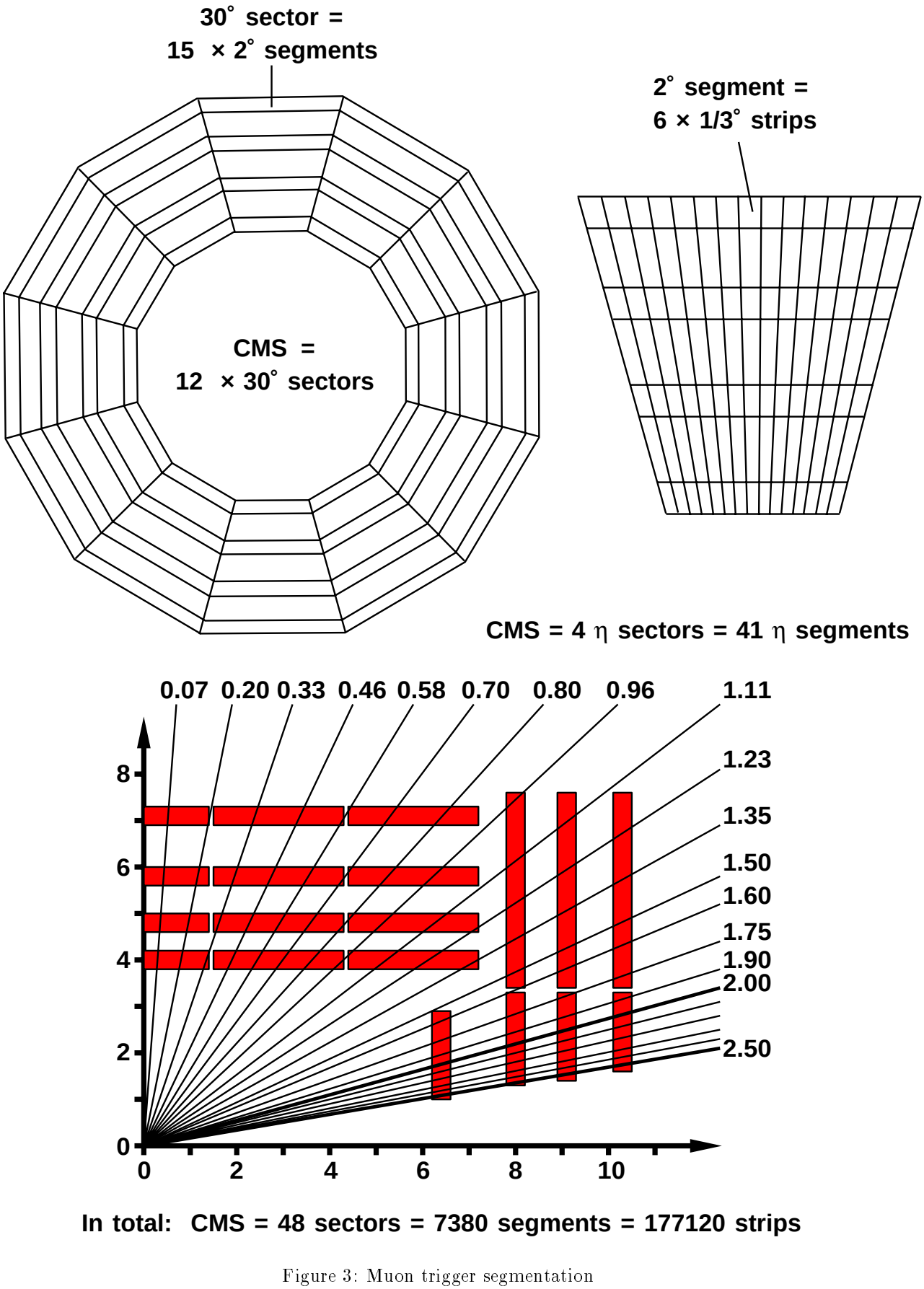


Figure 3: Muon trigger segmentation

3. **Creation of hit patterns.**

   This action can be selected by **IPATFL(1)>0**. Every combination of hits in four triggering planes is stored in a matrix **IPATMX1** having $\Delta_{12}$, $\Delta_{23}$, $\Delta_{34}$ and $\eta$ coded as indices. Here $\Delta_{ij} = [\varphi_j] - [\varphi_i]$ is a difference of hit positions in stations $i$ and $j$. Thus each element of the matrix corresponds to a given pattern and contains a number of events containing this pattern. In case of **IPATFL(2)>0** the next step – selection of valid patterns – is performed immediately without storing the pattern matrix on disk. If **IPATFL(2)>0** the matrix is stored as unformatted fortran file on unit 93.

   One can collect statistics running the program with **IPATFL=1** several times with many input data sets. In that case the pattern matrix provided on input unit 91 will be updated.

4. **Selection of valid patterns.**

   A set of routines called **MTPATxx** could be selected by **IPATFL(2)>0**. IF **LEVELOR=1** the symmetrization procedure is performed, i.e. every negative particle pattern gets its mirror image. IF **LEVELOR>1** than every **LEVELOR** adjacent strips are OR-ed together. The patterns are sorted according to their frequency. Mattrix elements of **IPATMX1** corresponding to the first 98% (variable **PATCUT** in **SEQ,HIPAT**) of patterns are filled with a momentum index of the given input set. Others are set to zero. The mattrix is again stored as unformatted fortran file on unit 93 or passed to the next step – merging of patterns with **IPATFL(3)>0**.

   Optionally, the valid patterns are written onto unit 60 with the formatted write statement (routine **MTPATEN**). The file contains the following information:

   (a) The difference between hit strip numbers in stations 1 and 2,

   (b) Ditto, for stations 2 and 3,

   (c) Ditto, for stations 3 and 4,

   (d) Number of such patterns found in the data set,

   (e) $\chi^2$ w.r.t the average pattern for this data set,

   (f) Serial number of the pattern written,

   (g) Rapidity interval number (1-21).

   The format is `(3I6,I10,F12.4,I10,I4)`.

   The pattern finding works on the 'MTDI' bank; if the data set does not contain it, the digitization should be switched on in order to create it. Cluster size generation should be suppressed **(IDIGFL=1)** and the input file should contain single muons. A given file should correspond to a single $p_t^{cut}$ value, and an arbitrary $\eta$ interval.

5. **Merging of valid patterns.**

   This option could be selected with **IPATFL(3)>0**. The program takes the **IPATMX1** mattrix from the previous step or reads it from unit 91. A second matrix is read form unit 92 into **IPATMX2**. Valid patterns from both matrices are compared. Results are stored in **IPATMX1**. The value of an **IPATMX2** element is the highest $p_t^{cut}$ index (see Tab. 1) for which the given pattern is valid. At the end the matrix is stored on an unformatted fortran file 93, to be used in the next steps.

   If one wants to study low and high $p_t$ algorithms (**LOWMOM**=1 and 0 respectively) two different matrices should be prepared.

6. **Efficiency and rate estimations.**

   Fast estimation of efficiencies and rates could be selected with **IRATEFL=1**. It is based on single muon patterns created with IPATEFL(1)>0 and stored in **IPATMX1** on unit 91. Valid patterns prepared in the previously described steps are loaded from unit 92 into IPATMAX2. Number of events accepted for every $p_t^{cut}$, $p_t$ and $|\eta|$ interval are stored in **NEFF** array onto unit 52. One can supply this file as unit 51 for the next run with another data set in order to merge different $p_t$ and increase statistics.

   Rate estimation uses parametrized spectrum of prompt muons. The program calculates "sharpness" of every efficiency curve, i.e. a fraction of triggered events to be accepted for offline analysis.

7. **Efficiency and rate calculations.**

   Detailed efficiency and rate calculations are based on full event simulation. This option could be selected with **IRATEFL>1**. Input files 91 and 92 containing the valid pattern matrices as described above have to be given. The file 91 corresponds to the low $p_t$ algorithm, making use of two muon stations, having two triggering planes each. This algorithm is applied for $|\eta| < 1.5$ and $p_t < 8$ GeV. The file 92, based on four muon stations (one triggering plane each) is used in the remaining phase space.

   The data set will be read (from unit 41) and hit patterns in the 'MTDI' bank will be compared with valid patterns in the mattrix. If the MTDI bank does not exist on the input file it can be created by setting **IDIGFL=1 or 2**. The results are written onto an unformatted file (unit 52) and, optionally, onto a text file (unit 60) in form of four columns: $p_t^{cut}$ index, $p_t^{cut}$ value [GeV], fraction of events accepted with the given $p_t^{cut}$, and fraction of events having at least two tracks above the given $p_t^{cut}$. Interpretation of these numbers depends on the input data.

   One can collect statistics running the program several times with many input data sets. In that case the output (unit 52) of the previous run should be supplied as unit 51.

   **Efficiency (IRATEFL=2).**

   In order to calculate trigger efficiency the input file should contain muons with fixed $p_t$ and $\eta$ range. Then the third column of the output file gives an efficiency for these $p_t$ and $\eta$ range as a function of the $p_t^{cut}$. The fourth column is irrelevant. One can obtain a full set of efficiency curves by running the program several times with input files covering an interesting region. A UNIX script **mtrat2.job** has been provided in order to automatize this task.

   **Rates (IRATEFL>2).**

   In this case the input file should contain full events and a single pass of the program is enough. The third and the fourth output column contains fractions of events accepted by the single and double muon trigger respectively, as a function of the $p_t^{cut}$. Trigger segmentation is taken into account, i.e. two muons within the same segment are seen as only one. In order to get trigger rates one can normalize these numbers according to the number of generated events and the expected cross section.

## 4.2 Description of the data cards

The data cards are of two types:

1. Free format steering cards, to define flow, program options, debug flags,
2. Fixed format cards defining the input/output file names.

**The free format data cards:**

| KEY | N | I | VAR | Short description | COMMON |
|---|---|---|---|---|---|
| RUN | 2 | I | **event selection:** | | /DEBFLG/ |
| | | | MINEVT | number of events to be skipped at the beginning | |
| | | | MAXEVT | number of events to be processed afterwards | |
| DEBU | 10 | I | **debugging flags:** | | /DEBFLG/ |
| | | | IDEBFL(1-4) | debug main/digitization/patterns/rates | |
| | | | IDEBFL(5-10) | not yet defined | |
| OUT | 10 | I | **optional output definitions, 0/1= off/on** | | /DEBFLG/ |
| | | | IOUTFL(1) | banks – write BOS banks onto fortran file 40 | |
| | | | IOUTFL(2) | ntuples – write ntuples onto RZ file | |
| | | | IOUTFL(3) | hbook – write histograms onto RZ file | |
| | | | IOUTFL(4) | histdo – write histograms onto standard output 6 | |
| | | | IOUTFL(5) | patterns – write sorted valid patterns onto text file 60 | |
| | | | IOUTFL(6-10) | not yet defined | |
| FLOW | 6 | I | **flow control:** | | /DEBFLG/ |
| | | | IUSERFL | user defined analysis | |
| | | | IDIGFL | 0: no digitization | |
| | | | | 1: digitization with cluster size of 1 strip | |
| | | | | 2: digitization with cluster size generated in the MTCLSZ | |
| | | | IPATFL(1) | create hit patterns | |
| | | | IPATFL(2) | select valid patterns | |
| | | | IPATFL(3) | merge valid patterns | |
| | | | IRATEFL | 1: estimate efficiency or rates | |
| | | | | 2: calculate efficiency | |
| | | | | >2: calculate rates | |
| | | | ITIMEFL | timing study | |
| | | | INOISFL | neutron, gamma and electronics noise generation | |
| EFF | 2 | I | **how to define patterns and calculate efficiency:** | | /EFFLAG/ |
| | | | LOWMOM | 0: chooses high momentum option for valid patterns | |
| | | | | 1: chooses low momentum option for valid patterns | |
| | | | | 1 shuold be used for $p_t < 8$ GeV and $\|\eta\| < 1.5$ | |
| | | | LCONCHI | presently not used | |
| | | | LEVELOR | 0: no action | |
| | | | | 1: symmetrization | |
| | | | | >1: symmetrization and OR of LEVELOR adjacent strips. | |
| CUTS | 3 | M | **kinematical cuts:** | | /PTCUTS/ |
| | | I | INDEXPT | $p_t$ index | |
| | | R | AETAMIN | lower $\|\eta\|$ limit | |
| | | R | AETAMAX | upper $\|\eta\|$ limit | |
| RPC | 4 | R | **RPC efficiency and noise:** | | /RPCEFF/ |
| | | | EFFMIP | RPC efficiency for MIPs | |
| | | | EFFNEU | RPC efficiency for neutrons | |
| | | | EFFGAM | RPC efficiency for photons | |
| | | | RPCNOIS | RPC intrinsic noise [Hz/cm$^2$] | |
| | | | ELENOIS | electronic noise [Hz/channel] | |
| | | | RPCGATE | time gate [ns] | |
| COLL | 9 | R | **Collider information:** | | /COLLIS/ |
| | | | XLUMIN | Luminosity [$10^{34}$ cm$^{-2}$ s$^{-1}$] | |
| | | | CROTIM | bunch crossing time distans [ns] | |
| | | | XSECTN | pp inelastic cross section [mb] | |
| | | | others | not used | |
| END | | | **end of the data cards** | | |

**The fixed format (A4,6X,A60)** cards define the names of the various input and output files (x - blank):

- **F30**xxxxxxx**name** – neutron and gamma fluxes (text),
- **F40**xxxxxxx**name** – output data set (BOS fortran),
- **F41**xxxxxxx**name** – input data set, single muons or physics events (BOS fortran),
- **F42**xxxxxxx**name** – input data set, pileup events (BOS fortran),
- **F51**xxxxxxx**name** – input file of efficiencies, and rates for IRATEFL>0 (unformatted fortran),
- **F52**xxxxxxx**name** – output file for sorted valid patterns, efficiencies, and rates (unformatted fortran),
- **F60**xxxxxxx**name** – output file for sorted valid patterns, efficiencies, and rates (text),
- **F91**xxxxxxx**name** – input file to read the pattern matrix IPATMX1 (unformatted fortran),
- **F92**xxxxxxx**name** – input file to read the pattern matrix IPATMX2 (unformatted fortran),
- **F93**xxxxxxx**name** – output file to store the pattern matrix IPATMX1 (unformatted fortran),
- **HIFI**xxxxxx**name** – PAW file name (RZ file),
- **ENDF** – end of file name data cards.

For each of the output files chosen with the **OUT** and **FLOW** data cards the user is asked to provide an appropriate file name cards. It is checked by the **mtinchk** routine.

## 4.3 Histograms

A number of histograms is defined in the routine **mtinih**:

780x particles at the vertex and background rates

781x geometrical hits

782x digitized hits

783x digitization cross checks

784x timing study

785x pattern creation

786x pattern selection

787x and 79xx efficiencies and rates

## 4.4 How to run the program?

The current version of the program is placed at

**csf.cern.ch : /u1/zh/wrochna/MTRIG.**

All the source is stored in **mtrig005.cmz** file. In order to run the program the files **stor003.cmz** and **bos.cmz** are also needed. You can load them automatically into CMZ using **mt.kumac**. One can use provided script **mtrig.com** for compilation and linking. It can be done inside CMZ session by executing **mt#mtrig** macro. Finally, at subdirectory **run** there are scripts containing also input cards necessary to run the program with various options and automatize submitting the jobs for many input files.

### 4.5 Future development

The current version of MTRIG has several items which needs to be improved:

- Rate estimation is based on prompt muons only; K, $\pi$ decays needs to be included;
- In case of two particles crossing one strips two **MTDI** rows are created.
- **MTNOISE** does not generate clusters.
- One cannot run **INOISFL>0** without **IDIGFL>0** i.e. without reading BOS banks with hits.
- Timing calculations are now disabled, because they need to be updated.
- There is no text(60) file in case of **IPATFL(1)>0**.
- In case of many pp collisions per bunch crossing the bank **MTBX** is not created and respective links in **MTEV** and **MTVX** are not defined. The **MTHI** bank is removed after each pp collision, thus the links from/to **MTDI** are valid only during digitization of this pp collision.

The present version of MTRIG uses its own definition of the RPC geometry. It was very useful in the development phase when we needed to optimise the granularity and the algorithm. Now the design is more stable and it is possible to make use of the "official" CMSIM/GEANT geometry. Next version will have this version implemented and digitisation will be moved from MTRIG to CMSIM. At the same time a number of new features will be implemented:

- Nonprojective geometry in $\eta$.
- Proper treatment of dead areas ("3 out of 4" algorithm).
- Segment numbers will be assign according to hit position in MS4 (now MS1 is used).
- Sectors in $\varphi$ will be replaced by "rings" in $\eta$.
- Cones treated by segment processors will be expressed in absolute $\varphi$ value rather then in $\Delta\varphi$.

## References


[1] *CMS Letter of Intent*, **CERN/LHCC 92-3**.

[2] M. Konecki, Warsaw University M.Sc. thesis, 1922, unpublished,
M. Konecki, J. Królikowski, G. Wrochna, *Simulation study of the single muon, RPC based trigger for CMS*, **CMS TN/92-39**.

[3] G. Wrochna, *RPC Based Muon Trigger for the CMS Detector at LHC*, talk given at the RPC Workshop, Roma, Feb.'93, **CMS TN/93-80**.

[4] H. Czyrkowski et al., *RPC Based CMS Muon Trigger – Progress Report*, **CMS TN/93-111**.

[5] C. Charlot et al., *CMSIM-CMANA, CMS Simulation Facilities*, **CMS TN/93-63**.

[6] G. Wrochna, *Proposal of Unified Criteria for Muon Trigger Studies*, **CMS TN/94-200**.

[7] **CERN/LHCC 94-20**, *CMS Status Report and Milestones*, May 1994.

[8] M. Huhtinen and P. A. Aarnio, *Radiation problems at LHC experiments, I: Neutral particle background*, CMS technical note **CMS TN/94-135** and University of Helsinki preprint **HU-SEFT R 1994-01**;
M. Huhtinen and G. Wrochna, *Estimation of the RPC Muon Trigger Rates Due to Neutral Particles*, **CMS TN/94-138**.