

Trigger Validation in COMPASS

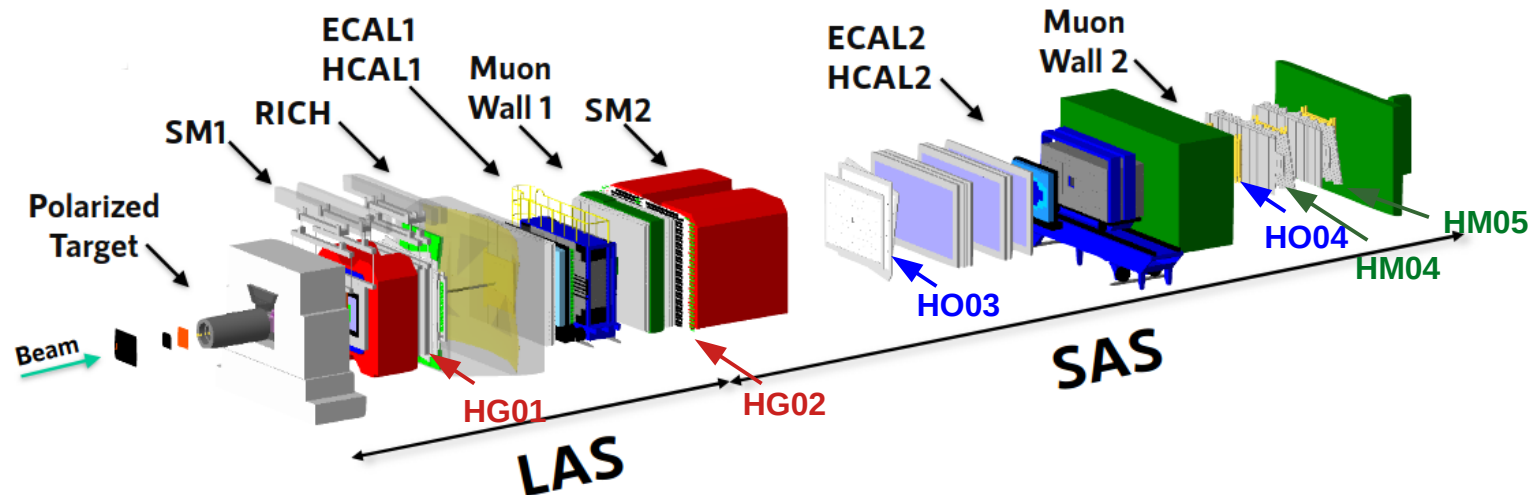
In the Drell-Yan data of COMPASS, there are 3 dimuon triggers implement:

- **LAS-LAS: trigger bit 8** – both muons at large angle
- **LAS-OT: trigger bit 2** – one muon at large, another at small angle (in Outer system)
- **LAS-MT: trigger bit 0** – one muon at large, another at small angle (in Middle system)

The LAS system consists of 2 large hodoscopes: **HG01** (Z=585 cm) and **HG02** (Z=1610 cm)

The Outer system consists of 2 hodoscopes: **HO03** (Z=2150 cm) and **HO04** (Z=3970 cm)

The Middle system consists of 2 small hodoscopes: **HM04** (Z=4023 cm) and **HM05** (Z=4784 cm)



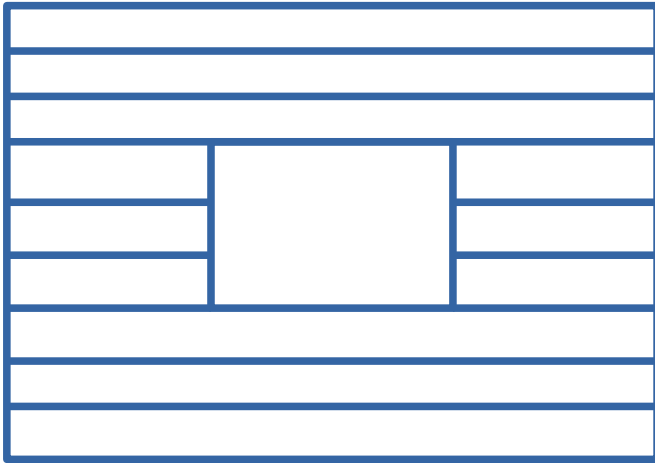
Geometry of the hodoscopes

The hodoscopes are made of plastic scintillator horizontal slabs.

Some are rectangular, symmetric and with a rectangular dead zone in the middle: HG01, HO03

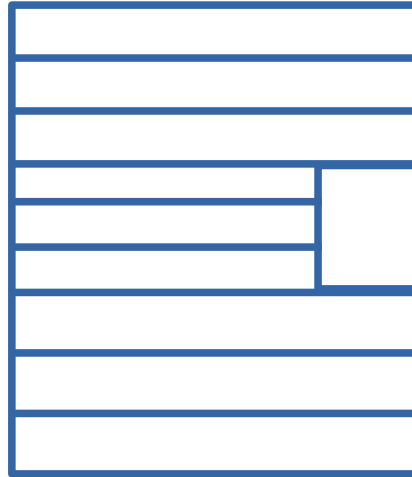
Some are rectangular, asymmetric, with central wrt vertical dead zone: HG02Y1, HG02Y2, HO04Y1, HO04Y2

Some are simply rectangular, with top and bottom parts: HM04, HM05



HG01 (32 slabs)

HO03 (18 slabs)



HG02Y1, HG02Y2 (32 slabs)

HO04Y1, HO04Y2 (16 slabs)

(HG02Y2 and HO04Y2 are
Inverted images of these)



HM04 (24 slabs)

HM05 (24 slabs)

In the Drell-Yan analyses, LAS-MT trigger is not used, because polluted with muons from Hadron beam decays.

Why trigger validation is needed

In the experiment, the dimuon trigger ensures that the recorded event is interesting, since it contains two or more particles candidate to be muons (since detected after walls).

Whenever we make an event selection for analysis based in a dimuon trigger, we want to be sure that the selected muons where the ones making the trigger system fire.

Selecting the dimuon trigger bit(s) in the analysis is not enough. We want to check if this muon track effectively could have crossed one of the trigger systems that fired. This is the validation.

Example: if the LAS-OT trigger fired (bit 8), we check that one of the muon tracks, extrapolated to the HG01 and HG02 positions, is within the active area of these hodoscopes; while the other Extrapolated to positions of HO03 and HO04 is within the active area of these other hodoscopes.

Technicalitie: hodoscopes positions

```
// find the exact positions of the hodoscopes:

const PaSetup& setup = PaSetup::Ref();

int ipl1 = setup.iDetector("HG01Y1__");
assert(ipl1 >= 0);
const PaDetect& d1 = setup.Detector(ipl1);

int ipl2 = setup.iDetector("H004Y1_m");
assert(ipl2 >= 0);
const PaDetect& d2 = setup.Detector(ipl2);

int ipl3 = setup.iDetector("H004Y2_m");
assert(ipl3 >= 0);
const PaDetect& d3 = setup.Detector(ipl3);

int ipl6 = setup.iDetector("HG02Y1__");
assert(ipl6 >= 0);
const PaDetect& d6 = setup.Detector(ipl6);

int ipl7 = setup.iDetector("HG02Y2__");
assert(ipl7 >= 0);
const PaDetect& d7 = setup.Detector(ipl7);

int ipl8 = setup.iDetector("H003Y1_m");
assert(ipl8 >= 0);
const PaDetect& d8 = setup.Detector(ipl8);
```

Since this is generic, same for all events, you want to put this at the very beginning, outside any cycles.

Technicalities: LAS-LAS validation

```
const PaTrack& trMup = e.vTrack(p.iTrack()); // defined for mu+ track out of p.PID()==5
const PaTrack& trMum = e.vTrack(pp.iTrack()); // defined for mu- track out of pp.PID()==6

PaTPar extr2mup,extr2mum,extr3mup,extr3mum,extr4mup,extr4mum,extr5mup,extr6mup,extr7mup,extr5mum,extr6mum,extr7mum;
bool ok3,ok4,ok5,ok6,ok7,ok8,ok9,ok10,ok11,ok12,ok13,ok14;
ok3=ok4=ok5=ok6=ok7=ok8=ok9=ok10=ok11=ok12=ok13=ok14=0;
ok3=trMup.Extrapolate(d1.Z(),extr2mup); //HG01
ok4=trMum.Extrapolate(d1.Z(),extr2mum);
ok5=trMup.Extrapolate(d7.Z(),extr3mup); //HG02Y2
ok6=trMup.Extrapolate(d6.Z(),extr4mup); //HG02Y1
ok7=trMum.Extrapolate(d7.Z(),extr3mum);
ok8=trMum.Extrapolate(d6.Z(),extr4mum);
ok9=trMup.Extrapolate(d8.Z(),extr5mup); //H003
ok10=trMup.Extrapolate(d2.Z(),extr6mup); //H004Y1
ok11=trMup.Extrapolate(d3.Z(),extr7mup); //H004Y2
ok12=trMum.Extrapolate(d8.Z(),extr5mum);
ok13=trMum.Extrapolate(d3.Z(),extr6mum);
ok14=trMum.Extrapolate(d2.Z(),extr7mum);

// LAS-LAS:
if (trigger&256) {

if (ok3 && ok4 && d1.InActive(extr2mup(1),extr2mup(2)) && d1.InActive(extr2mum(1),extr2mum(2)) &&
    ((ok5 && d7.InActive(extr3mup(1),extr3mup(2))) || (ok6 && d6.InActive(extr4mup(1),extr4mup(2)))) &&
    ((ok7 && d7.InActive(extr3mum(1),extr3mum(2))) || (ok8 && d6.InActive(extr4mum(1),extr4mum(2)))) ) trigval2LAS = 1;

} //end trigger&256
```

Technicalities: continue with LAS-OT validation

```
// LAS-OT:
if (trigger&4) {

    int trigval1,trigval2,mupinLAS,mupinOT;
    mupinLAS=mupinOT=0;
    trigval1=trigval2=-1;

//mup:
    if (ok3 && d1.InActive(extr2mup(1),extr2mup(2)) &&
        ((ok5 && d7.InActive(extr3mup(1),extr3mup(2))) || (ok6 && d6.InActive(extr4mup(1),extr4mup(2))))) {
        trigval1=1;
        mupinLAS=1;
    } //mup in LAS

    if (ok9 && d8.InActive(extr5mup(1),extr5mup(2)) ) { //mup in H003
        if ((ok10 && d2.InActive(extr6mup(1),extr6mup(2))) || //mup in H004Y1
            (ok11 && d3.InActive(extr7mup(1),extr7mup(2))) ) //mup in H004Y2
            {
                trigval1=1;
                mupinOT=1;
            }
    } //mup in OT

//mum:
    if (trigval1==1 && mupinOT==1) { //mup in OT then check if mum in LAS
        if (ok4 && d1.InActive(extr2mum(1),extr2mum(2)) &&
            ((ok7 && d7.InActive(extr3mum(1),extr3mum(2))) || (ok8 && d6.InActive(extr4mum(1),extr4mum(2))))) trigval2=1;
    }

    if (trigval1==1 && mupinLAS==1) { //mup in LAS then check if mum in OT
        if (ok12 && d8.InActive(extr5mum(1),extr5mum(2)) && //mum in H003
            ((ok14 && d2.InActive(extr7mum(1),extr7mum(2))) || //mum in H004Y1
             (ok13 && d3.InActive(extr6mum(1),extr6mum(2))) ) ) //mum in H004Y2
            trigval2=1;
        }
    }
    if (trigval1 == 1 && trigval2 == 1) trigval0TLAS=1;
} //end trigger&4
```

How to use

```
static int trigval2LAS;  
static int trigvalOTLAS;
```

These are obtained in the way described before and are the flags you want to store in your trees for dimuon trigger validation.

In the macros for analysis, ask for condition:

```
if ((trigger&256 && trigval2LAS==1) || (trigger&4 && trigvalOTLAS==1))
```