# $B^0 \to K^* \mu^- \mu^+$ MC Filter

Marcin Chrząszcz
mchrzasz@cern.ch
Thomas Blake
Nicola Serra

University of Zurich[UZH]

EWP WG meeting, CERN
October 9, 2016

# A glimpse in the Run1 analysis

⇛ In the Run1 we have asked for a filtered MC to correct for detector acceptance.
⇛ Asked for $5.5M$ events (after stripping in DST), which means we generated around $110M$ events.
⇛ After our full selection we ended up with with only $1.4M$ events.

**Warning!**

The stripping line has a PID cut inside: $\mathrm{PID}_\mu > -3$.
This essentially means we model that efficiency from MC.

# Run2 options

1. Repeat what we did in Run1 and keep the PID cuts.
2. Filter on stripping removing the PID cut.
3. Filter on MC truth:
   - 4 charge tracks on $\mathrm{StdAllNoPiDPions/Kaons/Muons}$
   - And truth matched the decay channel: $\mathrm{mcMatch('[B0 => K*(892)0mu+mu-]CC')}$

**Why MCTruth?**

$\Rightarrow$ We are using a very old stripping line that for sure can be (and should be) optimized for the final analysis of Run2!
$\Rightarrow$ Producing an MCTRUTH match sample would allow the sample to be reused for future analysis even if the stripping line will change!

# Retentions

⇛ To study the solution I have used 2012 Physics MC.

⇛ I have taken 17.250 simulated events.

⇛ Here is the results:

| Type | Filter retention | Events in the ntuple | Truth Matched |
|---|---|---|---|
| Strip | 3447 (20 %) | 4975 | 1648 |
| Strip no PID$_\mu$ | 3504 (20.3 %) | 5176 | 1660 |
| MCTruth | 5009 (29 %) | 4456 | 1660 |

⇛ Now I have cross check this running the same algorithms on stripped and non stripped MC always getting the same numbers.

⇛ For speed purpose I have put a cut on the $m_{K^*} < 1300$ MeV (can be adjusted if needed).

⇛ Other option to consider is to remove `ISMUON` form stripping to get all efficiencies from `PIDCalib`.

# Plans

$\Rrightarrow$ With Tom we feel that it would be best to ask for $200M$ generated events.

$\Rrightarrow$ Also we noticed that we have $50M$ events of some old MC10 (Stripping 12) MC, which we propose to delete.

$\Rrightarrow$ For PPG: The $R(D^*)$ have already got green light for more then $1000M$ generated events, so we getting the $200M$ should not be a problem.

$\Rrightarrow$ To discuss: Do we want a flat $m(K\pi)$ sample or we can keep the $K^*$?

# Plans 2

$\Rrightarrow$ Besides the normal $B \to K^*\mu\mu$ PHSP we should ask for other MC channels.

$\Rrightarrow$ I proposed to scale the old numbers by factor: $\frac{5}{3}$.

| Decay | DecFile event type | N. of events | N. of events Run2 |
|---|---|---|---|
| $B \to K^* J/\psi$ (physics) | 11144001 | 2M | 3.5M |
| $B \to K^* J/\psi$ (PHSP?) | xxxxxxxx | 0 | 3.5M |
| $B \to K^*\mu\mu$ (physics) | 11114001 | 1M | 1.5M |
| $\Lambda_b \to \Lambda(1530)\mu\mu$ | 15114000 | 1M | 1.5M |
| $\Lambda_b \to pK\mu\mu$ | 15114011 | 2M | 3.5M |
| $B_s^0 \to \phi\mu\mu$ | 13114002 | 0.6M | 1M |
| $B_u \to K\mu\mu$ | 12113001 | 1M | 1.5M |

$\Rrightarrow$ This would be unfiltered production and this MC will be needed for other analysis as well.

$\Rrightarrow$ Do we want to simulate a flat $q^2$ in the $B \to K^*\mu\mu$?

$\Rrightarrow$ Do we want to have a flat $K\pi$ mass distribution in the simulation?

# MC model

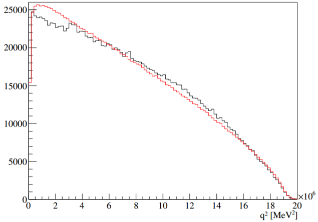
# Acceptance correction

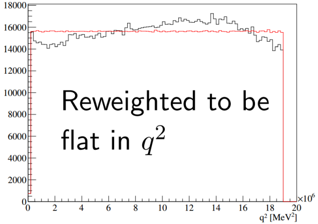
⇛ The decay of $B^0 \to K^{*}\mu^{-}\mu^{+}$ is described by 3 helicity angles and the invariant mass squared of two leptons ($q^2$).

⇛ In order to model the detector acceptance we have used a large MC sample of PHSP simulated events.

⇛ There is a caveat: the $q^2$ distribution.

⇛ We had to reweight it to make it flat.



# Can we optimize it?

⇛ It would be nice if we could generate not only the flat angle distributions but also a flat $q^2$.

⇛ There exists already a model for it: `FLATQ2`.

⇛ It basically reweighs the distribution by $1/p_T^{\text{had}}$.

⇛ The problem is that it was design to generate the flat distribution of decays $B \to X\ell\nu$ :

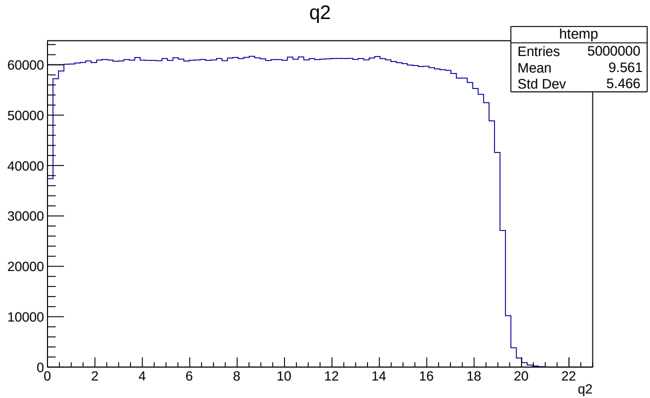
```
void EvtFlatQ2::init(){

  // check that there are 0 arguments
  checkNArg(0);
  checkNDaug(3);

  //We expect B->X l nu events
  checkSpinParent(EvtSpinType::SCALAR);
  checkSpinDaughter(1,EvtSpinType::DIRAC);
  checkSpinDaughter(2,EvtSpinType::NEUTRINO);

}
```

⇛ Will not work in current version for $B \to K^{*}\mu\mu$.

# Modifying the FLATQ2 1

$\Rrightarrow$ I wrote a mirror model that requires that the two leptons are `DIRAC`, and called it `FLATQ2EWP`.

$\Rrightarrow$ And improves the situation a lot:

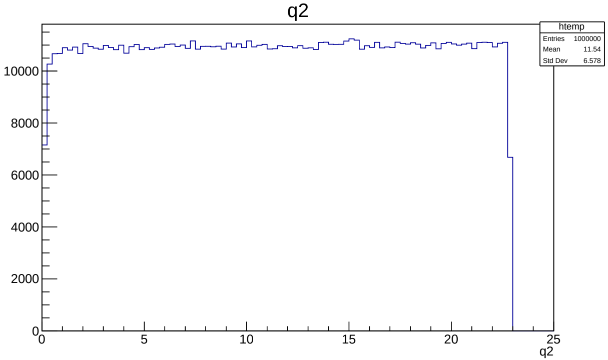



$\Rrightarrow$ So much flatter but the and and the begging still not flat.

$\Rrightarrow$ End of the spectrum is due to $K^*$ width? $\twoheadrightarrow$ Lets test it with $B$ –

# Modifying the FLATQ2 1

⇛ FLATQ2EWP use to simulate the $B \to K\mu\mu$:

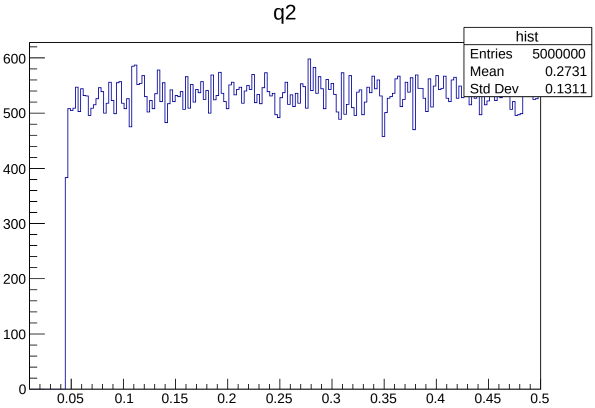



⇛ Oki so end of the spectrum is understood and not much can be done there.

⇛ Now the low $q^2$: Can this be just Phase space suppression: $\sqrt{\lambda} = \sqrt{1 - 4m_\ell^2/q^2}$

# Modifying the FLATQ2 2

⇛ `FLATQ2EWP` with phase space suppression factor.




⇛ Now it's perfect.

# Update since last week

⇛ Discussion was made via: JIRA

⇛ It was suggested my Michal to incorporate the new model into the current one to save the code.

⇛ Thanks to John for merging the two codes:

```
void EvtFlatQ2::init(){

  // check that there are 3 daughters
  checkNDaug(3);

  // We expect B -> X lepton lepton events
  checkSpinParent(EvtSpinType::SCALAR);

  EvtSpinType::spintype d1type = EvtPDL::getSpinType(getDaug(1));
  EvtSpinType::spintype d2type = EvtPDL::getSpinType(getDaug(2));

  if (!(d1type == EvtSpinType::DIRAC || d1type == EvtSpinType::NEUTRINO)) {
      EvtGenReport(EVTGEN_ERROR,"EvtGen") << "EvtFlatQ2 expects 2nd daughter to "
                                          << "be a lepton" <<std::endl;
      EvtGenReport(EVTGEN_ERROR,"EvtGen") << "Will terminate execution!"<<std::endl;
      ::abort();
  }

  if (!(d2type == EvtSpinType::DIRAC || d2type == EvtSpinType::NEUTRINO)) {
      EvtGenReport(EVTGEN_ERROR,"EvtGen") << "EvtFlatQ2 expects 3rd daughter to "
                                          << "be a lepton" <<std::endl;
      EvtGenReport(EVTGEN_ERROR,"EvtGen") << "Will terminate execution!"<<std::endl;
      ::abort();
  }

  // Specify if we want to use the phase space factor
  _usePhsp = false;
  if (getNArg() > 0) {
      if (getArg(0) != 0) {_usePhsp = true;}
  }

  EvtGenReport(EVTGEN_INFO,"EvtGen") <<"EvtFlatQ2 usePhsp = "<<int(_usePhsp)<<std::endl;

}
```

# Update since last week

⇛ Discussion was made via: JIRA

⇛ It was suggested my Michal to incorporate the new model into the current one to save the code.

⇛ Thanks to John for merging the two codes:

```
void EvtFlatQ2::decay( EvtParticle *p){

  p->initializePhaseSpace(getNDaug(),getDaugs());

  EvtVector4R p4Xu = p->getDaug(0)->getP4();

  EvtVector4R p4ell1 = p->getDaug(1)->getP4();
  EvtVector4R p4ell2 = p->getDaug(2)->getP4();

  double pXu_x2 = p4Xu.get(1)*p4Xu.get(1);
  double pXu_y2 = p4Xu.get(2)*p4Xu.get(2);
  double pXu_z2 = p4Xu.get(3)*p4Xu.get(3);
  double pXu = sqrt(pXu_x2+pXu_y2+pXu_z2);
  double prob(0.0);
  if (fabs(pXu) > 0.0) {prob = 1/pXu;}

  // Include the phase space factor if requested
  if (_usePhsp) {

    double Lambda = lambda((p4ell1+p4ell2).mass(), p4ell1.mass());
    if (Lambda > 0.0) {prob=prob/sqrt(Lambda);}

  }

  if (pXu > 0.01) {setProb(prob);}

  return;
```

# FLATQ2 Conclusion

$\Rrightarrow$ The new model was tested by me and John.
$\Rrightarrow$ Changes won't have any influence on the existing DEFILES as the flag is by default switched off.
$\Rrightarrow$ The commit was merge to master by Gloria today.
$\Rrightarrow$ We thank all people involved action

$\Rrightarrow$ The whole things took $<$week and is already available for production!
$\Rrightarrow$ There is also other model XLL, see Biplab slides more suitable for $B \to K\pi\mu\mu$.

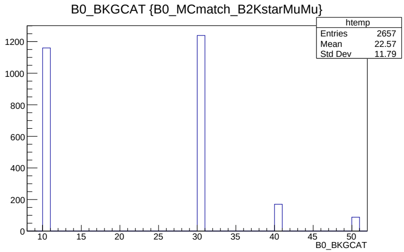
$\Rrightarrow$ For the acceptance correction we propose to generate $B \to K^*\mu\mu$ with FLATQ2 and filter it on truth matching.

# Backup

This is not related to MC requests.

# MCmatching studies.

⇛ Let's look how the candidates that have been matched by: $\mathrm{mcMatch('[B0 => K*(892)0mu+mu-]CC')}$ look like:

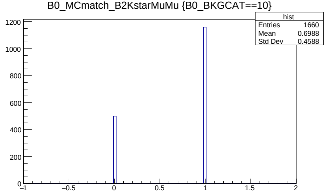



⇛ BKGCAT==10 is the pure signal. The mcMatch is not changing anything in that number of entries.

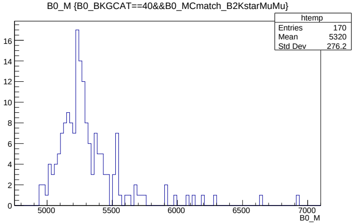
⇛ BKGCAT==30 is the K=$K \leftrightarrows \pi$ swaps. This goes away with some PID selection

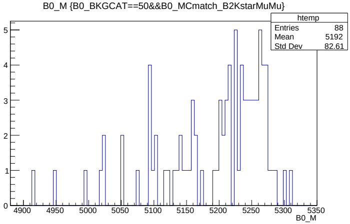
# MCmatching studies.

⇛ Now all `BKGCAT==10` have true mcMatch:




⇛ How does `BKKCAT==50,40` (missID +FSR, FSR)look like:




⇛ We need to consider which `BKGCAT` we should use for the analysis