Table of Contents

Ghosts (since 2012)......1

how to get a new ghostprobability on not-so-new datasets in DV......1

wording (i keep "new" and "old" at their meanings from 2012, to avoid messing up the 2012 documentation)......1

which ghost probability is the "correct" ghost probability......1

howto get the run2 25ns tuning on run1 data......1

howto get the run2 50ns tuning on run1 data......2

howto get the new/very new ghostprob on run1 data/MC (if not present already)......3

when to recompute the ghostprobability?......3

and on MDST?......4

documentation......4

how to get the "new" ghostprobability on "old" datasets (older than DV v33r1)......4

the story......4

Ghosts (since 2012)

how to get a new ghostprobability on not-so-new datasets in DV

wording (i keep "new" and "old" at their meanings from 2012, to avoid messing up the 2012 documentation)

old before Reco13e

new since Reco13e, tuned for the ghost problem we had in early Reco13, heavily discussed in summer 2012. (longtracks only, downstream tracks unchanged wrt **old**)

very new not even in Reco14, adds ghost probability for downstream tracks. (longtracks unchanged wrt **new**)

run2 50ns the run2 tuning as used in the early measurement period

run2 25ns the run2 tuning, picking up the "new" multiple scattering model

which ghost probability is the "correct" ghost probability

https://groups.cern.ch/group/lhcb-davinci/Lists/Archive/Flat.aspx?RootFolder=%2Fgroup%2Flhcb%2Ddavinci%2FLis

howto get the run2 25ns tuning on run1 data

use at least DaVinci v37r0

and getpack Tr/TrackTools v5r11 (or later. at the time of writing it's unknown which version will be too new in the future)

cp /afs/cern.ch/user/p/pseyfert/public/GhostProbRun2/25ns/* Tr/TrackTools/src

and compile Tr/TrackTools

on MDST, it was necessary to provide the RootInTES by hand to various tracking tools. Not sure if recent DaVinci versions cover this automatically. Try first w/o all the "MyStream" lines (commented out).

MDST should always work (except for turbo.mdst). full DST in stripping 21 and stripping22 are bug affected if you do **not** request the full RAW event (fixed for stripping23, don't request the full RAW event if you don't need it, please).

```
from STTools import STOfflineConf
STOfflineConf.DefaultConf().configureTools()
from Configurables import RefitParticleTracks
refitter = RefitParticleTracks()
refitter.DoFit = True
refitter.DoProbability = True
refitter.UpdateProbability = True
refitter.ReplaceTracks = True
refitter.Inputs = ["Phys/B2JpsiX_B2JpsiKKLine/Particles"]   ### EDIT ME!
DaVinci().UserAlgorithms = [refitter,decaytreetuple]           ### pick YOUR ntuple algorithm in
MyStream = "Dimuon"                                            ### EDIT ME
isStripping23 = True


from Configurables import TrackInitFit, TrackMasterFitter
from TrackFitter.ConfiguredFitters import ConfiguredMasterFitter
from Configurables import TrackBestTrackCreator, TrackMasterFitter, TrackStateInitTool

def giveitafit(thething):
   thething.addTool(TrackInitFit,"TrackInitFit")

   thething.TrackInitFit.addTool( TrackMasterFitter, name = "Fit" )
   thething.TrackInitFit.addTool( TrackStateInitTool, name = "Init")
   thething.TrackInitFit.Init.UseFastMomentumEstimate = True
   #thething.TrackInitFit.Init.OutputLevel = 1
   #thething.TrackInitFit.OutputLevel = 1
   ConfiguredMasterFitter( getattr(thething.TrackInitFit, "Fit"), SimplifiedGeometry = True, Lite
   thething.TrackInitFit.Init.VeloFitterName = "FastVeloFitLHCbIDs"
   if DaVinci().InputType == "MDST" or isStripping23:
      thething.TrackInitFit.RootInTES = MyStream
      thething.TrackInitFit.Fit.RootInTES = MyStream
      thething.TrackInitFit.Init.RootInTES = MyStream
      from Configurables import FastVeloFitLHCbIDs
      thething.TrackInitFit.Init.addTool(FastVeloFitLHCbIDs,name="FastVeloFitLHCbIDs")
      thething.TrackInitFit.Init.FastVeloFitLHCbIDs.RootInTES = MyStream


giveitafit(refitter)

from Configurables import OTRawBankDecoder, PatSeedFit, Tf__OTHitCreator, Tf__DefaultVeloPhiHitMa

ToolSvc().addTool(OTRawBankDecoder())
ToolSvc().addTool(PatSeedFit())
ToolSvc().addTool(Tf__OTHitCreator("OTHitCreator") )
ToolSvc().addTool(Tf__DefaultVeloPhiHitManager("DefaultVeloPhiHitManager"))
ToolSvc().addTool(Tf__DefaultVeloRHitManager("DefaultVeloRHitManager"))
if "MDST" == DaVinci().InputType or isStripping23:
   ToolSvc().OTRawBankDecoder.RootInTES = '/Event/' + MyStream
   ToolSvc().PatSeedFit.RootInTES = '/Event/' + MyStream
   ToolSvc().OTHitCreator.RootInTES = '/Event/' + MyStream
   ToolSvc().DefaultVeloPhiHitManager.RootInTES = '/Event/' + MyStream
   ToolSvc().DefaultVeloRHitManager.RootInTES = '/Event/' + MyStream
```

howto get the run2 50ns tuning on run1 data

use DaVinci v36r7p7

and getpack Tr/TrackTools v5r10

and getpack Phys/DaVinciTrackRefitting v2r1

cp /afs/cern.ch/user/p/pseyfert/public/GhostProbRun2/50ns/* Tr/TrackTools/src

and compile Tr/TrackTools

```
from STTools import STOfflineConf
STOfflineConf.DefaultConf().configureTools()  #only needed in case fitting is done
from Configurables import RefitParticleTracks
refitter = RefitParticleTracks()
refitter.DoFit = True
refitter.DoProbability = True
refitter.UpdateProbability = True
refitter.ReplaceTracks = True
refitter.Inputs = ["Phys/B2JpsiX_B2JpsiKKLine/Particles"]
DaVinci().UserAlgorithms = [refitter,decaytreetuple]


from Configurables import TrackInitFit, TrackMasterFitter
from TrackFitter.ConfiguredFitters import ConfiguredMasterFitter
from Configurables import TrackBestTrackCreator, TrackMasterFitter, TrackStateInitTool
```

```
def giveitafit(thething):

   thething.addTool(TrackInitFit,"TrackInitFit")
   #oldghostinfo.TrackInitFit.addTool(TrackMasterFitter,"Fit")
   #ConfiguredMasterFitter( oldghostinfo.TrackInitFit.Fit )

   thething.TrackInitFit.addTool( TrackMasterFitter, name = "Fit" )
   thething.TrackInitFit.addTool( TrackStateInitTool, name = "Init")
   thething.TrackInitFit.Init.UseFastMomentumEstimate = True
   #thething.TrackInitFit.Init.OutputLevel = 1
   #thething.TrackInitFit.OutputLevel = 1
   ConfiguredMasterFitter( getattr(thething.TrackInitFit, "Fit"), SimplifiedGeometry = True, Lite
   thething.TrackInitFit.Init.VeloFitterName = "FastVeloFitLHCbIDs"

giveitafit(refitter)
```

howto get the new/very new ghostprob on run1 data/MC (if not present already)

DaVinci v33r2 is required

```
from STTools import STOfflineConf
STOfflineConf.DefaultConf().configureTools()  #only needed in case fitting is done
from Configurables import RefitParticleTracks
refitter = RefitParticleTracks()
refitter.DoFit = True # not needed after Reco13e, ... well, to be on the safe side, maybe switch
refitter.DoProbability = True
refitter.UpdateProbability = True
refitter.ReplaceTracks = False  # to keep the old chi^2 values not to worry about stripping vs. n
refitter.Inputs = ["Phys/B2JpsiX_B2JpsiKKLine/Particles"]
DaVinci().UserAlgorithms = [refitter,decaytreetuple]
```

In case you run over (M)DSTs which were reconstructed with something older than Reco13e, you need to fit tracks, enabled with the `DoFit` property. (In fact after some discussions I realised that it might be worth switching it on even on Reco13e and later.) imho the `TrackInitFit` should be "correctly" configured out of the box. However you can configure it as follows:

```
from Configurables import TrackInitFit
refitter.addTool(TrackInitFit)
refitter.TrackInitFit.OutputLevel = 2   ## this is to make sure, that really your configured fitt
ConfiguredMasterFitter( refitter.TrackInitFit.Fit )
```

when to recompute the ghostprobability?

On some samples (see above) the "new" ghost probability is not available. Sometimes you want to have it anyhow, but it can be a bad idea to blindly recompute it, therefore an overview:

- Data 2011 (Reco12) standard: please don't play around here. You have cut away events in the stripping already. the new ghostprob invokes a track fit. that fit would make some bkg events pass your selection. for a decent study you must get hold of these ... well but you don't and therefore have something messy. especially for comparison with 2012 data it's the wrong thing to do. If you just want to know how your 2011 signal candidates will look like this year (and don't care for additional new signal and additional background): go ahead.
- Data 2011 (Reco12) standard, but disabling the fit (option not provieded): that avoids the problem above, but needs some hacking of the ghostprobability tool to make it work. Furthermore the input variables are not as they should be. Therefore I discourage this as well.
- Data 2011 (Reco12) minimal_invasive: I tested it successfully and think it makes sense to use this. Still you need to use a Reco13e-ish DaVinci version (FIXME: insert version number). However in my analysis It didn't seem to be a large benefit.
- signal MC (MC11a) standard, for comparison with 2012 data: I guess you're safe here since you look at signal only. But please do the selection again and don't take the stored candidates from the stripping

unless you really know what you're doing.
- signal MC (MC11a) for comparison with 2011/Reco12 data: do what you did to data as well. I recommend the minimal_invasive version.
- signal MC (MC 2012): I guess you're safe here

"standard" and "minimal_invasive" refer to preDVv33r1. The translation is as follows:

standard:

```
refitter.ReplaceTracks=True
```

minimal_invasive:

```
refitter.ReplaceTracks=False
```

and on MDST?

Is supported from DV v33r1 onwards. Thanks to Matt, Chris, and Marco for their help in debugging, developing and releasing the necessary ingredients.

documentation

Johann wrote an internal note about his work. http://cdsweb.cern.ch/record/1478372

Scripts for developement have been collected in Tr/TrackingScripts/GhostProbability (this is an Erasmus package)

how to get the "new" ghostprobability on "old" datasets (older than DV v33r1)

old = before Reco13e, new = since Reco13e.

this section is deprecated don't expect to find the necessary files, don't expect them to work, don't expect support.

everything you need is explained in:

```
/afs/cern.ch/work/p/pseyfert/public/GhostProbability_Production_TestVersion/README
```

this file is being updated frequently to account for new findings / features / options. The name still contains "TestVersion" since the number of happy testers for who it worked out of the box is smaller that my personal quality standards require to get rid of this comment ;).

FIXME: I'm working on a way to make this file readable via www such that I can put a clickable link here for convenience.

the story

In 2012 data we observe a high number of ghosts. The most striking discoveries were those reported by Anton (in the following "busy event ghosts") and Ricci ("SPD ghosts", NB: this name is misleading these are not ghosts in the SPD, but ghosts which have been discovered using the SPD)

- links to talks about this (FIXME is this list complete?)
    - SPD ghost discovery busy event ghosts
    - followup on busy event ghosts T&A 1 FIXME: is this list complete?

  - followup on SPD discovered ghosts T&A 1 T&A 2 T&A 3
  - Report to PPTSM link
  - Report to LHCb week https://indico.cern.ch/contributionDisplay?contribId=4&confId=203532
- they are **visible in the ghost rate** in the OT, above and below the IT (roughly OT modules 8 and 9). The ghost fraction in the affected region is about 50% (strongly depending on the track selection) and constant as a function of y.
- however those which actually harm you in physics don't have that striking pattern... (private discussions w/ Paul Se., Sascha S., Olivier C., probably others as well)

The current understanding is the following:

- These ghosts have always been there! Just nobody ever saw them. (In fact they caused trouble in the RICH PID in 2010, which lead to the introduction of a chi^2 cut at 5 for the RICH PID). They must have been there because you can make them appear / disappear by changing the OTProjector. Since the TrackEventCloneKiller, which defined Rec/Track/Best up to Reco12, used unfitted tracks, it must have had these ghosts as input.
- They became visible due to the new OTProjector in Reco13. (see above, we can switch them off)
- Old strategies (link 1 link 2) to get rid of them have been revived and received further developement and checks
  - first presentation 2012
  - Status report in T&A meeting, decision to include in endofyear reprocessing 2012
  - Summer Student report 2012
  - Tracking&Alignment meeting on what has been committed for Reco13e
  - Feedback in T&A meeting
- This is available since Reco13e
- a beta test is available at /afs/cern.ch/work/p/pseyfert/public/GhostProbability_BetaTester (READ THE FINE MANUAL! -- read the README file)

FIXME after discussions with Sascha and Olivier I'm not happy with the beginning following paragraph anymore.

The physical source for these ghosts is believed to be the high occupancy in the OT close to the IT. Hits get "transported" up and down until they make good combinations with hits in other layers. This is an obvious problem when a tracking detector is operated at 30% occupancy (see OT online monitoring, to verify that these are "normal" conditions through 2011/12). Usually gas detectors can deal with such high occupancies due to the drift time measurement. Unfortunately our TDC cannot read out multiple hits per straw, which makes the track chi^2 a function of pile-up, x (and thereby eta), and whether a track goes through IT or OT. (compare this to the analyses at the GPDs at the LHC, where they have to split up their samples in disk/barrel and pile-up). The OTProjector since Reco13 "hides" these problems from the analysts and has been tested on hadronic and muonic signal, and on "muonic" background (i.e. ghosts which fulfill a muon selection. of cause they're not muons because muon and ghost exclude each other). (due to manpower limitations no hadronic ghost sample has been tested. in hindsight that was a problem.)

-- PaulSeyfert - 11-Aug-2012

This topic: LHCb > Ghostsinreco13
Topic revision: r26 - 2018-05-22 - PaulSeyfert