Update: Decay Tree Fitter

Ralf Kliemt

Helmholtz-Institut Maiz, GSI Darmstadt

Mar. 2016

HIM
Helmholtz-Institut Mainz

GSI

1. TreeFitter - Concept
2. DecayTreeFitter Code Status
3. Usage
4. Examples

$\psi(2S) \to J/\psi\,\pi^+\pi^-$

$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

$\bar{p}p \to D^{*+}D^{*-} \to D^0\pi^+\bar{D}^0\pi^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

$\bar{p}p \to \eta_c\pi^0\pi^0 \to K_sK^\pm\pi^\mp\pi^0\pi^0$

Decay Fits in Rho until now

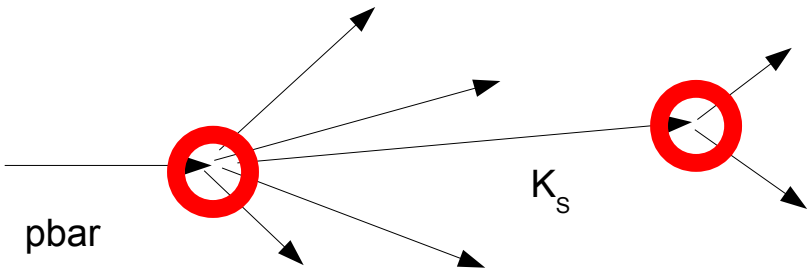
Vertex Fit Corrects final state momenta to one common point along trajectories (PndKinVtxFitter or PndKalmanVtxFitter)

Kinematic Fit Corrects daughter momenta to meet the mass or 4-momentum constraint

Executing fits subsequently and with locking some candidates, a leaf-by-leaf structure is created.

Example

1. Vertex fits for K_S and rest of tracks.
2. Mass constraint fit with vertex fitted K_S daughters
3. Locking K_S daughters
4. 4C fit on rest & K_S



Decay Tree Fitter

Fits the whole decay tree at once. Vertices, known masses, measured tracks & neutrals and beam/target measurement ("4C") are included as constraints. The common approach is the χ^2 fit with Lagrange multipliers.

$\rightarrow$ Very large parameter space and large matrices have to be inverted!

Solution: Kalman Filter approach

- Calculation of χ^2 is linearized
- Each constraint to the fit enters as one separate, scalar term
- Measurements are constraints and are treated similar to, e.g four-momentum conservations
  $\rightarrow$ maximum matrix dimension to be inverted is usually 5 (helices).
- Do not confuse with our track fitting!

Fit Parameters

(3) Primary vertex
(3) Secondary vertices
(3) Final state momenta
(4) Composite's four-momenta

Constraints

(5) Tracks (helix parameters)
(3) Clusters
(4) Initial four-momentum
(4) Internal four-momentum conservations
(1) Mass constraints

Example

$$\psi(2S) \to J/\psi\ \pi^+\pi^-$$
$$\hookrightarrow \mu^+\mu^-$$

23 Parameters:

12 4 Final State Particles
8 2 Composites
3 Primary vertex

32 Constraints:

20 4 Helices
8 2 P4-Conservations
4 Beam-Target

$\to$ 9 Degrees of Freedom

Beam: 4 plus Vertex: $2n - 3 = 5$

Existing Decay Tree Fitter

- BaBar & LHCb have a Tree Fitter, written by W.Hulsbergen
- The author provided us the latest stable code.
- Our goal: Implementation into PandaRoot

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Nuclear Instruments and Methods in Physics Research A 552 (2005) 566–575

NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A

www.elsevier.com/locate/nima

Decay chain fitting with a Kalman filter

Wouter D. Hulsbergen*

University of Maryland, College Park, MD 20742, USA

Received 4 March 2005; received in revised form 21 June 2005; accepted 26 June 2005
Available online 26 July 2005

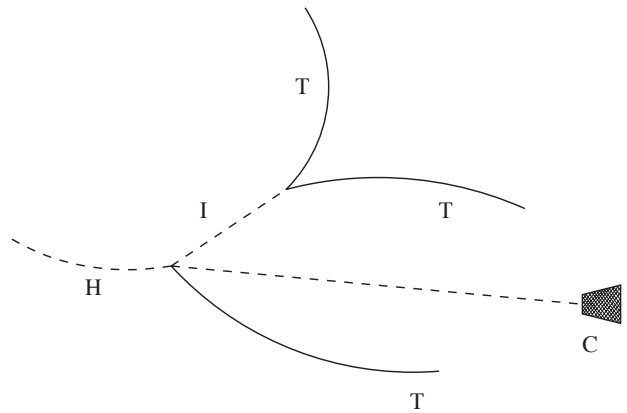


Fig. 1. Schematic picture of a decay tree with three charged particles reconstructed as track segments (T), one photon reconstructed as a calorimeter cluster (C), and two composite particles (I for 'internal' and H for 'head').

Status from last report:

- ✓ Obtain the code & look for showstoppers
- ✓ Matrices & Vectors: CLHEP → ROOT
- ✓ Framework interfaces: Gaudi → FairBase/ROOT & LHCB → PandaRoot
- ✓ Candidate Interfaces via Rho (calculations to be transformed)
- ✓ Move all particle fitters to Rho
- ✓ Intense Debugging & understanding the code

- Running more Tests & Debugging
- Covariance issue to be investigated.
- Neutrals need possibly fixing (may be an fsim thing)
- Treat "stabe" particles (Kshort) properly
- Test delayed Vertices (D's, Lambdas)
- Compare performance with available fitters

Status now:

- ✓ Port & understand the code
- ✓ Development: Beam constraint may have covariance

- Compare performance with available fitters
- Test delayed Vertices (D's)

→ See our SVN trunk from rev. 28854

WIP:

- → Neutrals need possibly fixing (may be an fsim thing)
- → Treat "stabe" particles (Kshort) properly
- → Test delayed Vertices (Lambdas)
- → Eliminate "decay length" parameter from code

Setup for Treefitting

```
PndAnalysis* theAnalysis = new PndAnalysis();

TString work = getenv("VMCWORKDIR");
location = work + "/pgenerators/EvtGen/EvtGen/Private/evt.pdl";
// force-replaces all entries in TDatabasePDG
RhoPdtLoader::ReadPDGTable(location,true);
tmppdg=TDatabasePDG::Instance();

double m0_p = TDatabasePDG::Instance()->GetParticle(2212)->Mass();
// *** the lorentz vector of the initial beam
TLorentzVector ini(0, 0, pbarmom,
                   sqrt(m0_p*m0_p + pbarmom*pbarmom) + m0_p);
double beamres = 1e-4*pbarmom;
RhoError inicov(4);
inicov[0][0]=1e-6*1e-6;
inicov[1][1]=1e-6*1e-6;
inicov[2][2]=beamres*beamres;
inicov[3][3]=beamres*beamres/(m0_p*m0_p + pbarmom*pbarmom);
```

In Eventloop

```
while (theAnalysis->GetEvent() && i++<nevts)
{
  theAnalysis->FillList(...);
  pbarp.Combine(...);
  for (j=0;j<pbarp.GetLength();++j)
  {
    // TREE FITTER
    //last number is verbosity
    PndDecayTreeFitter treefit(pbarp[j],inie,0);
    // 100um, for helix linearization limit
    treefit.SetToleranceZ(0.01);
    treefit.setMassConstraint(pbarp[j]->Daughter(1),m0_pi0);
    treefit.Fit();

    RhoCandidate* pbarpfit=pbarp[j]->GetFit();
    qa.qaFitter("fitter_",&treefit, ntp);
    //export whole decay tree, mc-difference & pulls
    qa.qaComp("pbarpfit_", pbarpfit, ntp, true, true);

    ntp->DumpData();
  }
}
```

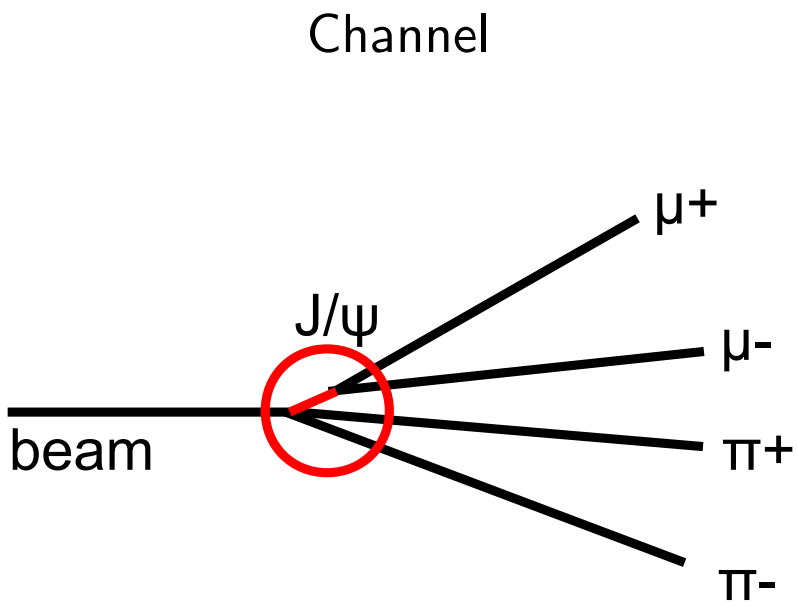
Examples

Simulation: "easy" FastSim

```
1 fastSim->SetUseFlatCov(true);
2 fastSim->AddDetector("SimpleTracker","thtMin=0. thtMax=145. ptmin
      =0.1 pmin=0.0 pRes=0.02 thtRes=0.001 phiRes=0.001 efficiency=1."
      );
3 fastSim->AddDetector("SimpleVtx", "thtMin=0. thtMax=145. ptmin=0.1
      vtxRes=0.005 efficiency=1.");
4 fastSim->AddDetector("EmcFS", "efficiency=1. thtMin=0.05 thtMax=10.0
      Emin=0.01 dist=8.0 aPar=0.02 bPar=0.0274");
5 fastSim->AddDetector("EmcFwCap", "efficiency=1. thtMin=10.0 thtMax
      =22.0 Emin=0.01 dist=2.5");
6 fastSim->AddDetector("EmcBarrel", "efficiency=1. thtMin=22.0 thtMax
      =142.0 Emin=0.01 barrelRadius=0.5");
7 fastSim->AddDetector("EmcBwCap", "efficiency=1. thtMin=142.0 thtMax
      =160.0 Emin=0.01 dist=0.7");
```

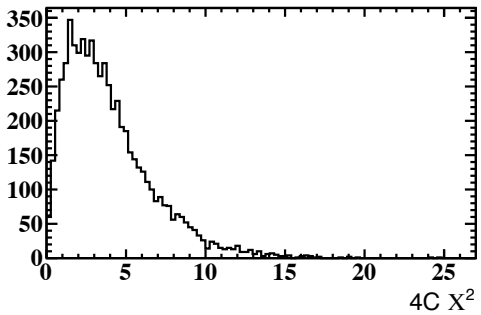
10000 events each with EvtGen

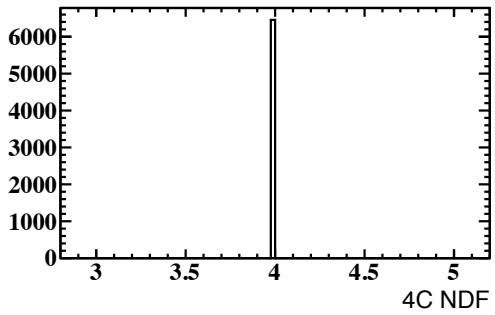
$\psi(2S) \to J/\psi\,\pi^+\pi^-$

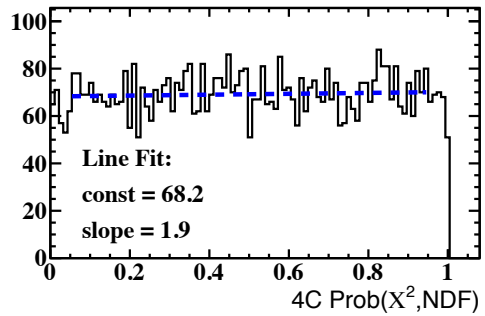


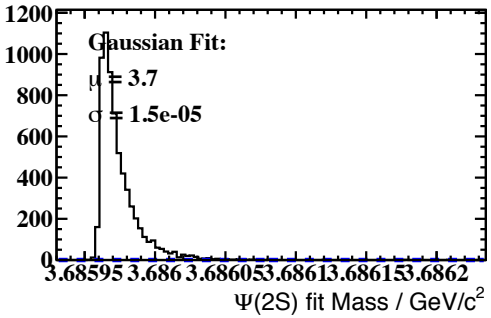
$\psi(2S) \to J/\psi\,\pi^+\pi^-$

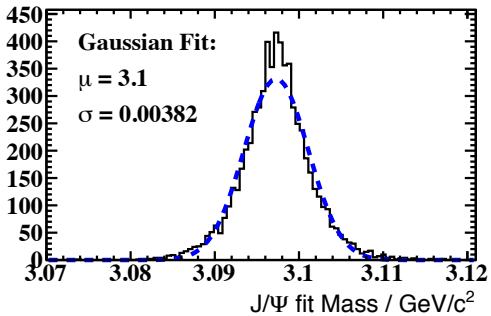
KinFit (4C) Quality

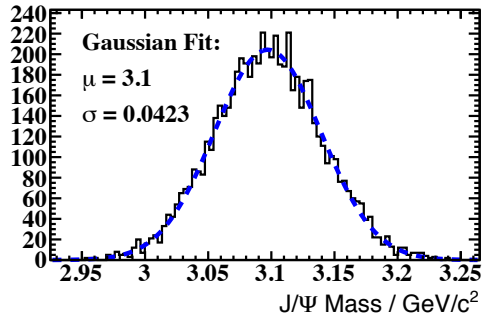






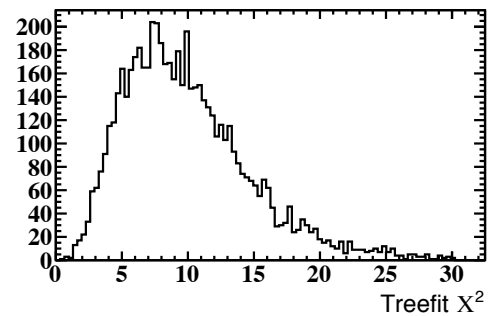


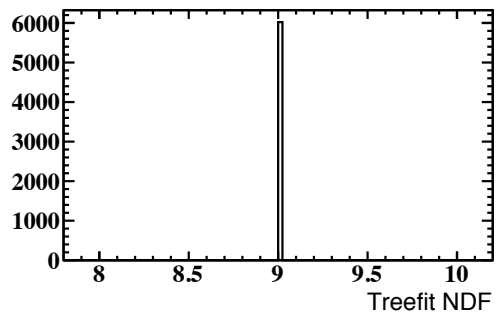


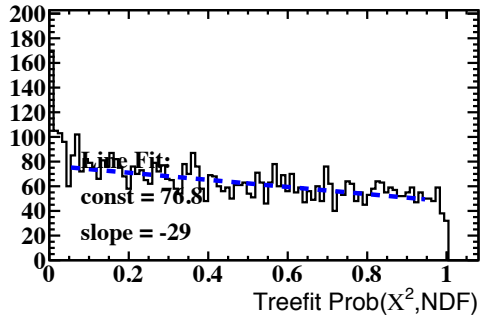


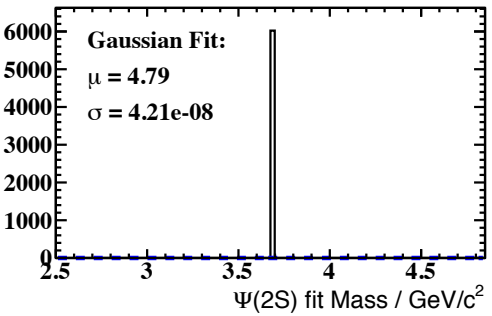
$\psi(2S) \to J/\psi\,\pi^+\pi^-$

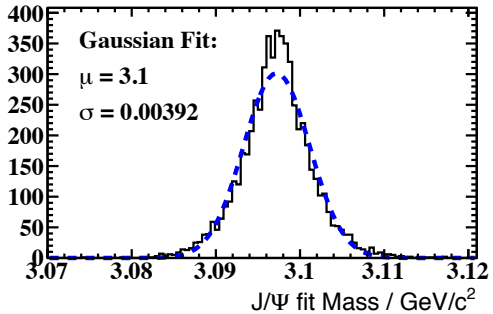
Treefit Quality

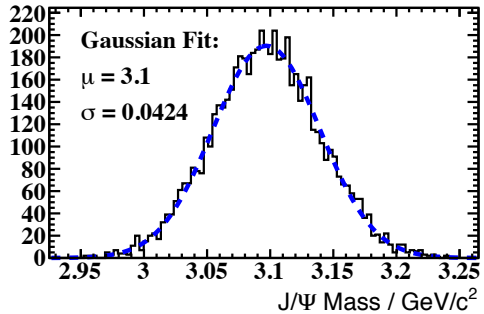






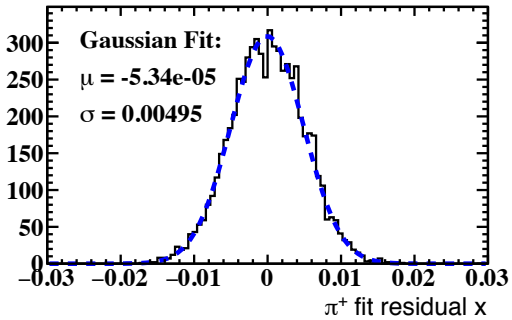


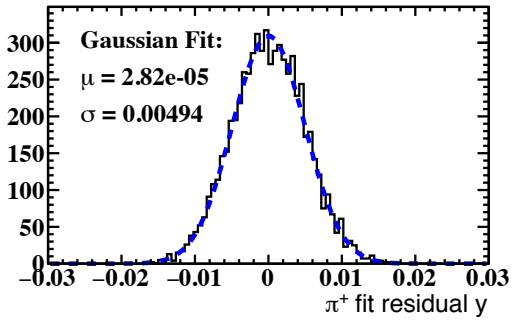


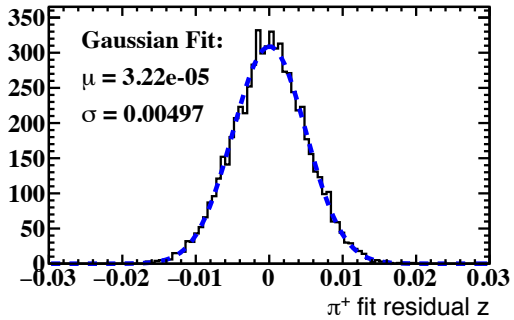


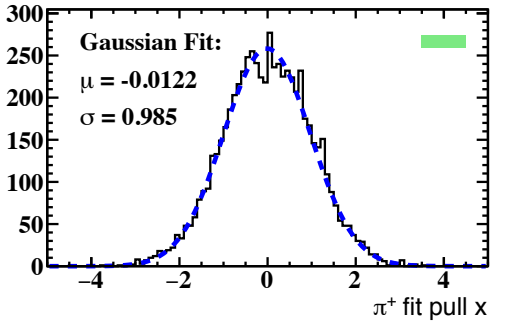
$\psi(2S) \to J/\psi\,\pi^+\pi^-$

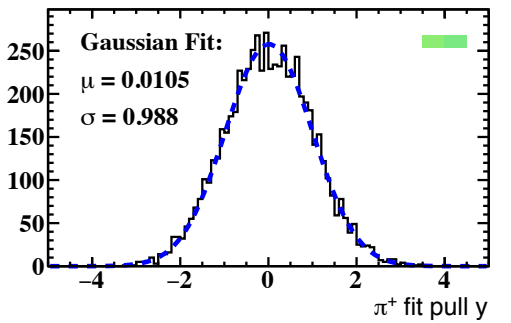
KinVtx Vertex

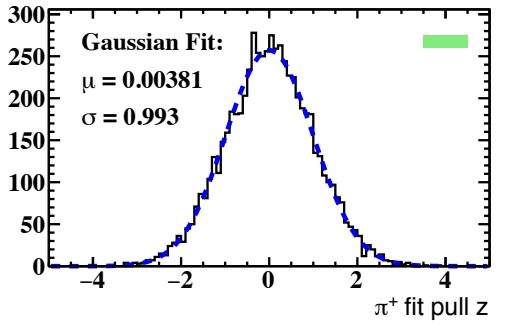






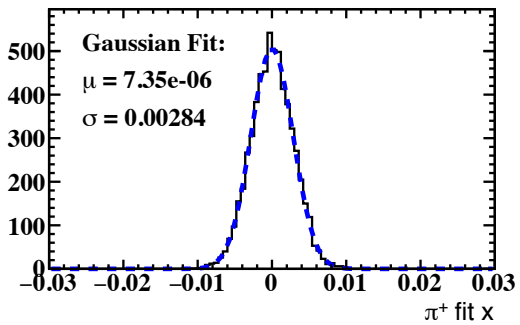


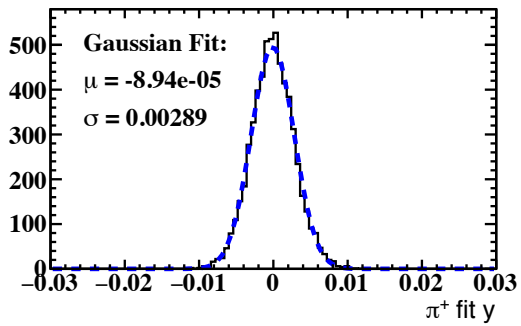


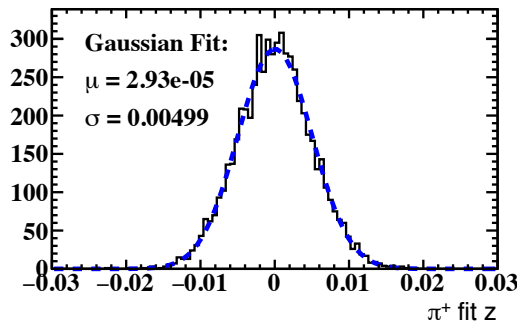


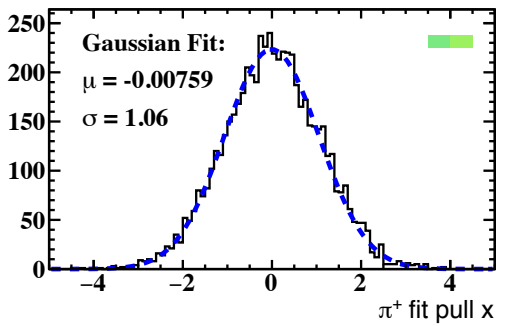
$\psi(2S) \to J/\psi\,\pi^+\pi^-$

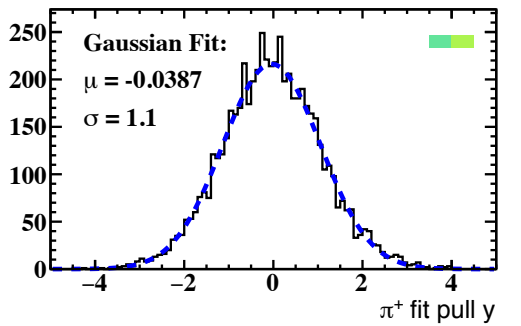
Treefit Vertex

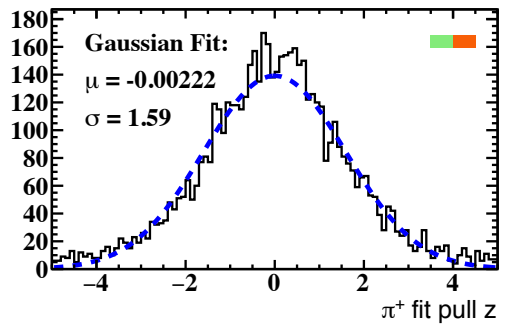






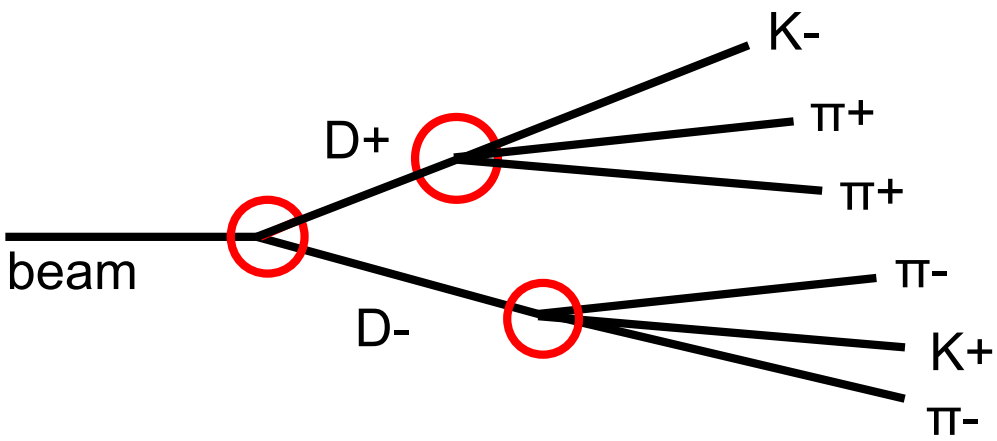






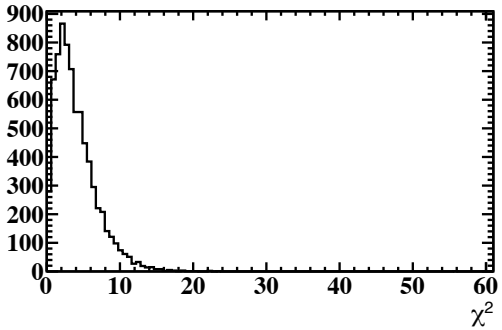
$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

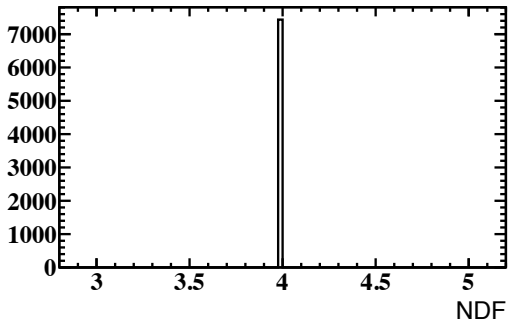
Channel

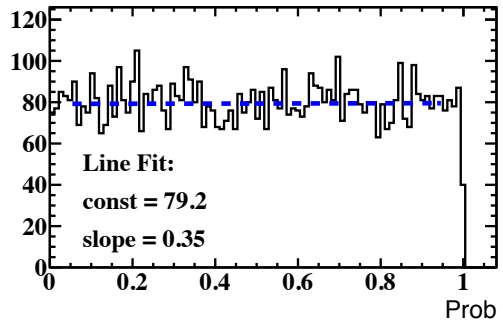


$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

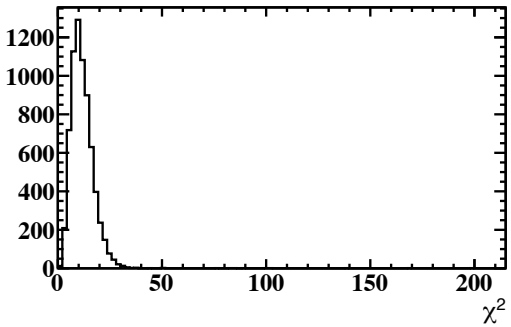
Leaf Fit QA

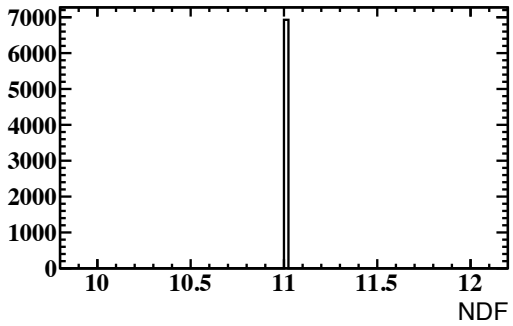


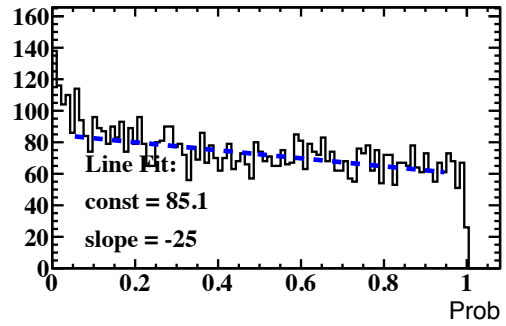




Treefit QA

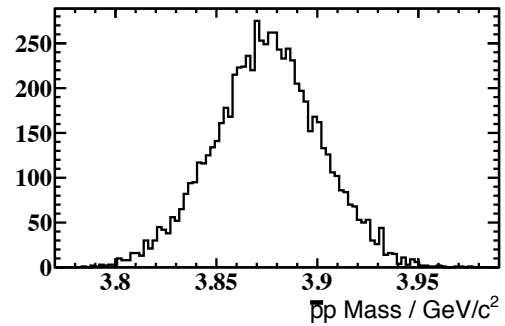


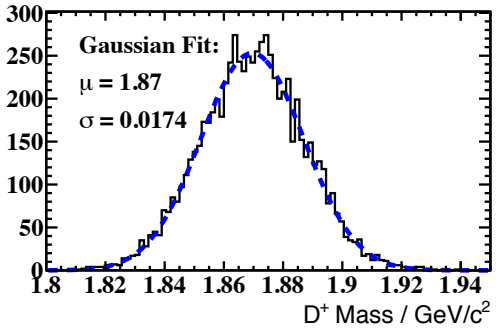


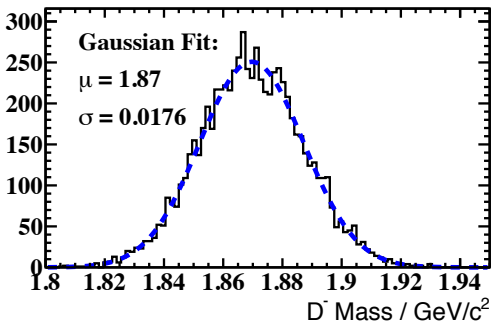


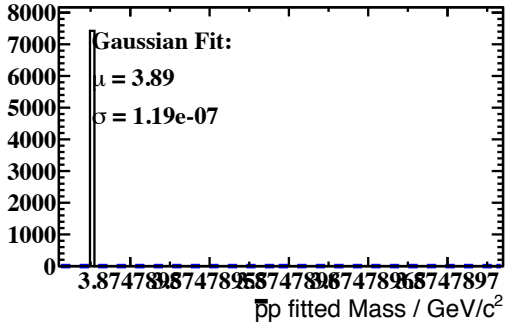
$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

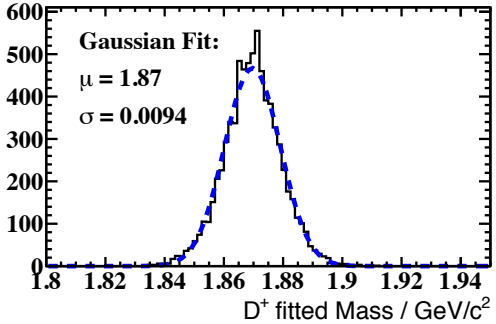
Leaf Fit Masses

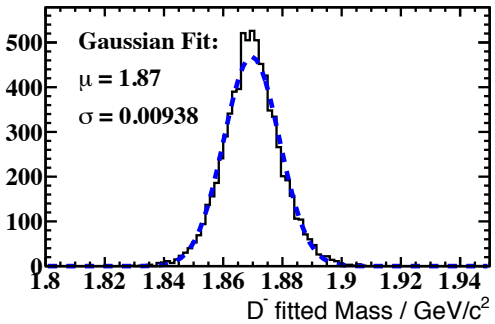






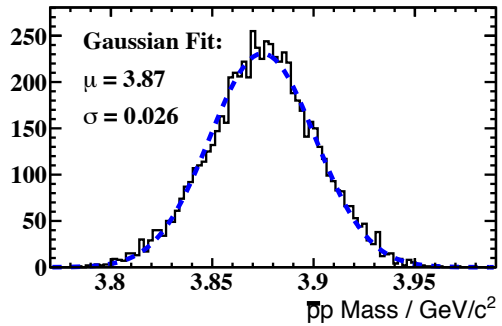


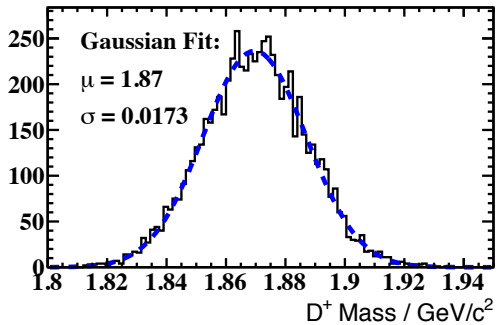


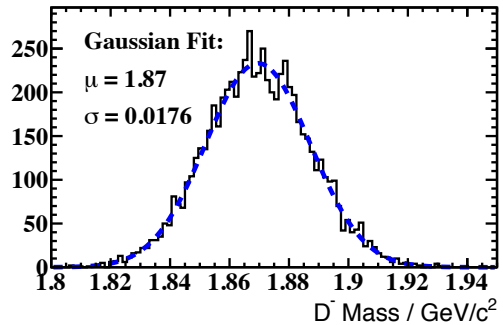


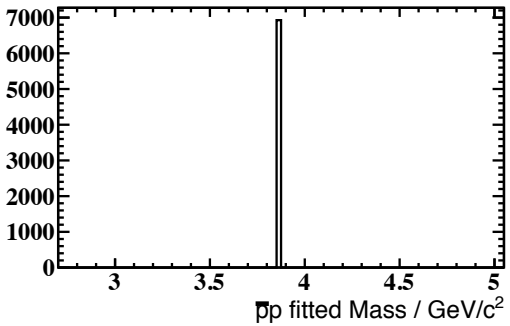
$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

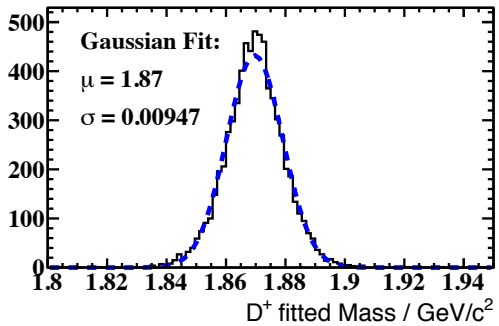
Treefit Masses

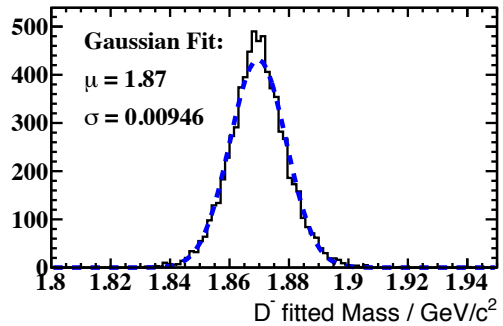






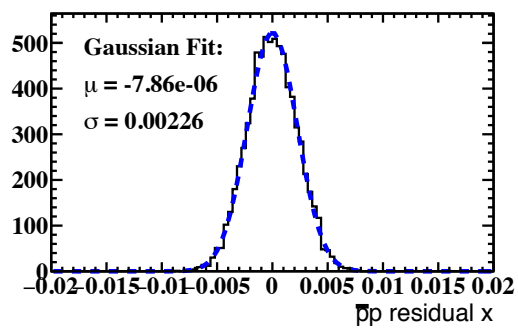


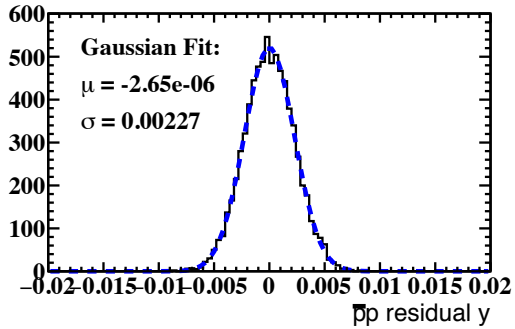


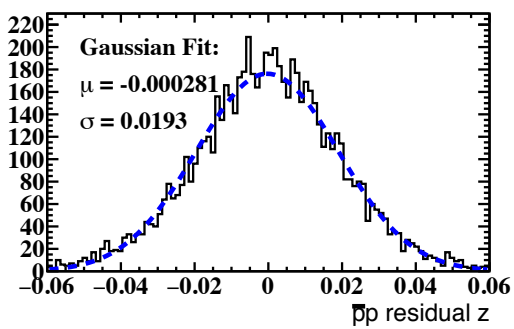


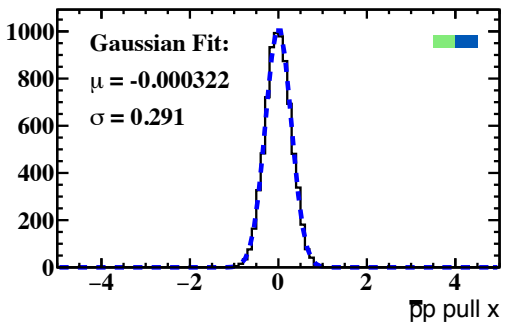
$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

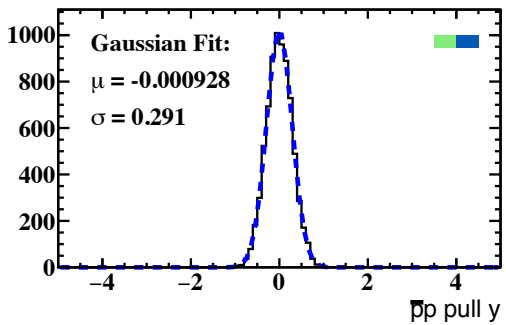
Leaf Fit $\bar{p}p$ Vertex

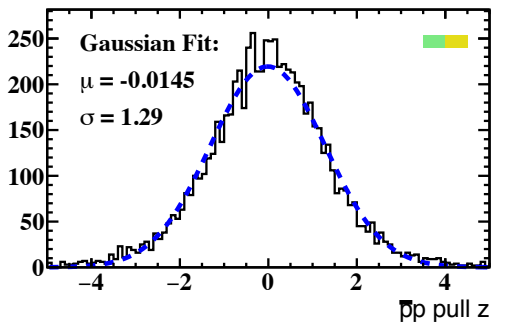






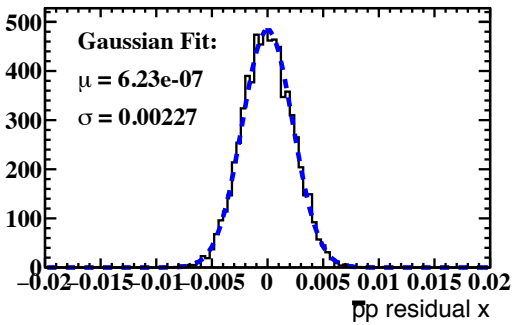


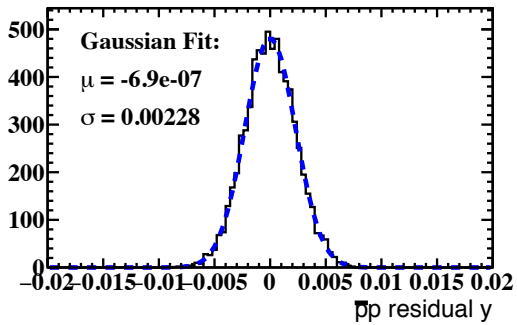


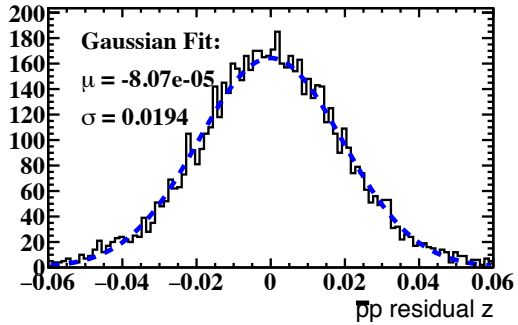


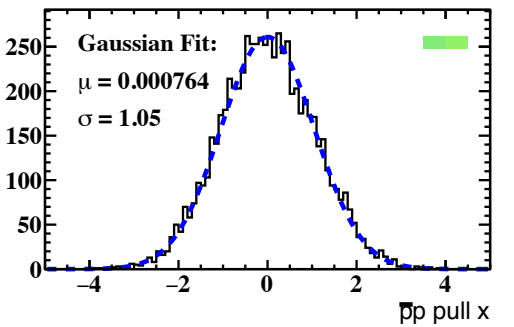
$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

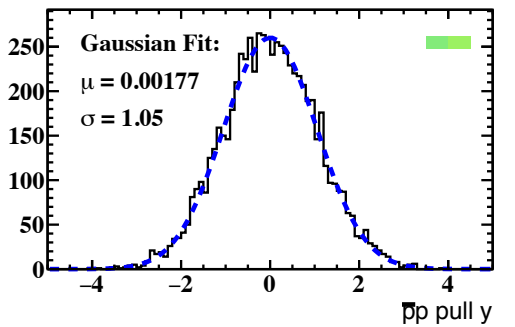
Treefit $\bar{p}p$ Vertex

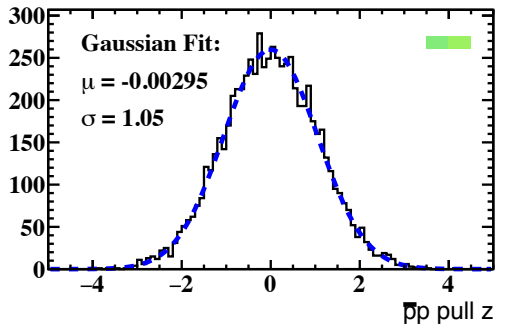






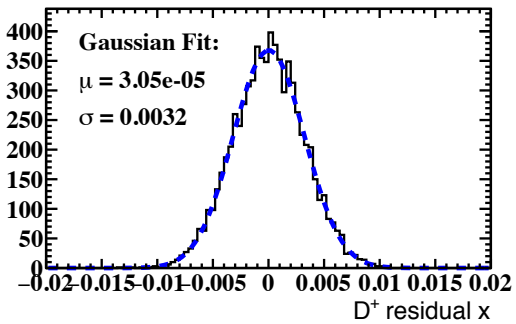


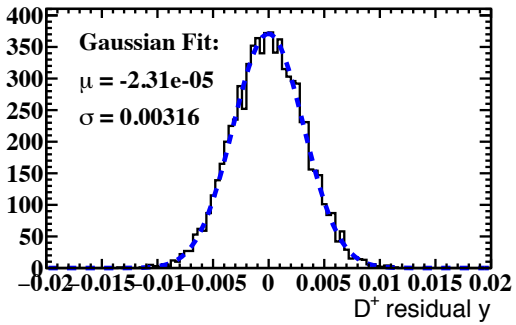


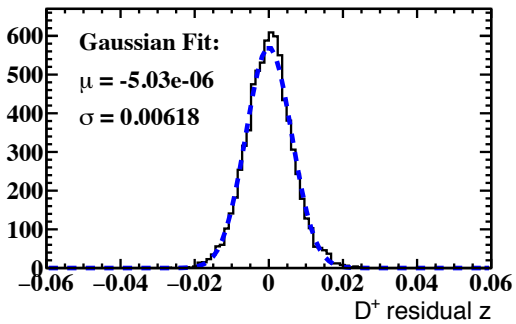


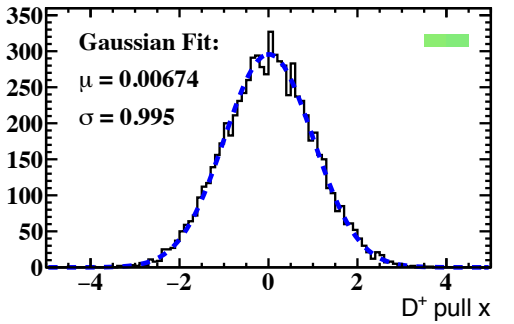
$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

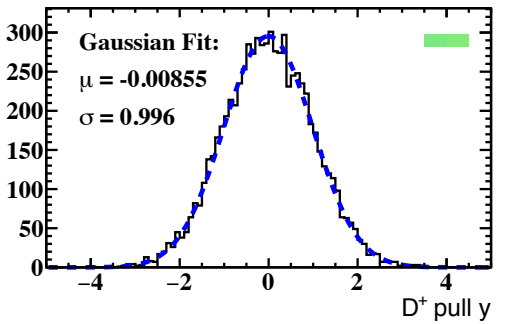
Leaf Fit D^+ Vertex

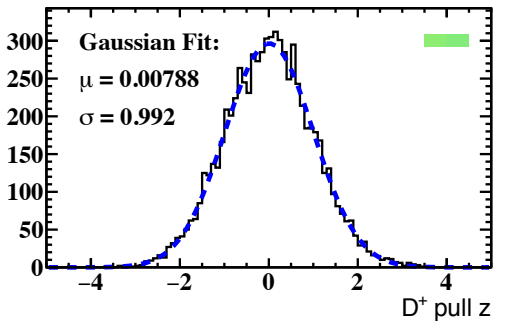






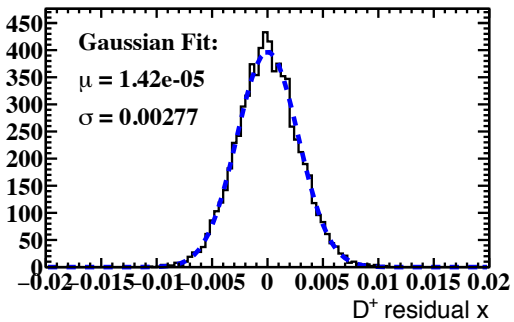


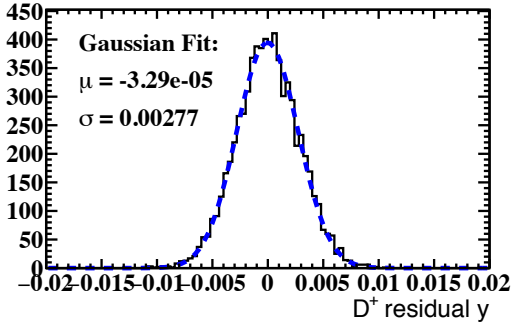


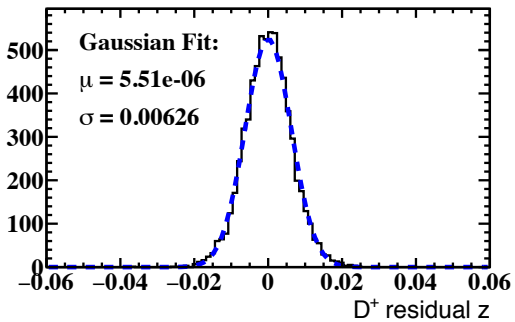


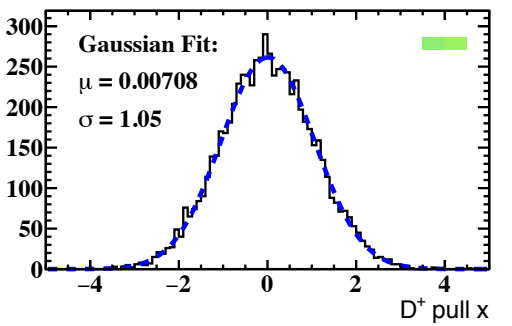
$\bar{p}p \to D^+D^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

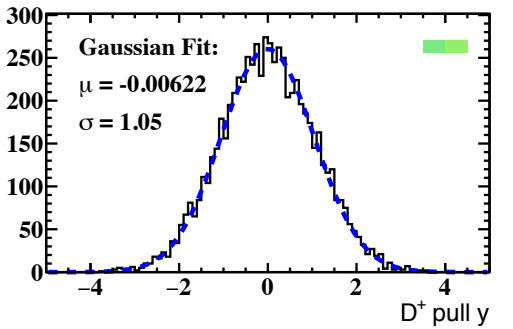
Treefit D^+ Vertex

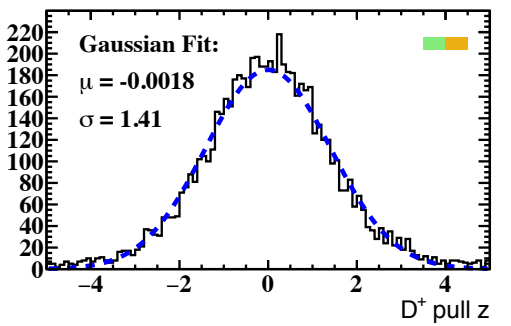






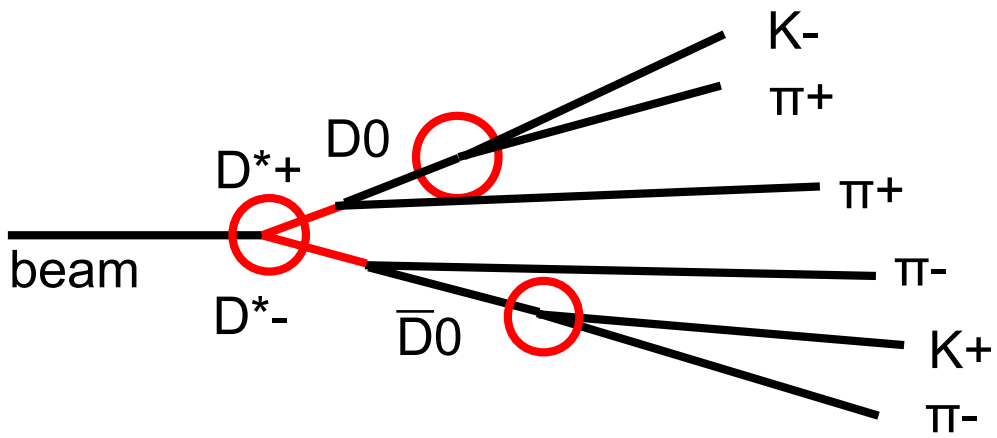






$$\bar{p}p \to D^{*+}D^{*-} \to D^0\pi^+\bar{D}^0\pi^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$$

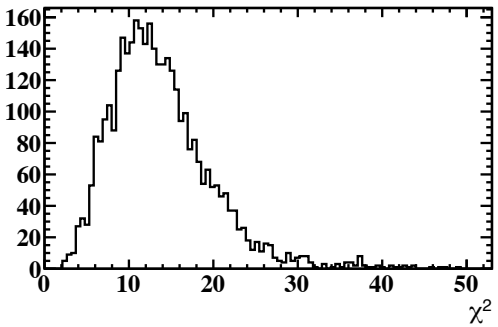
Channel

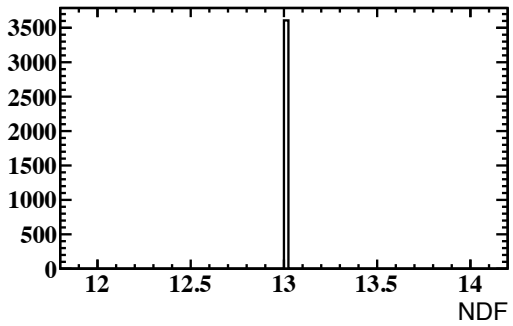


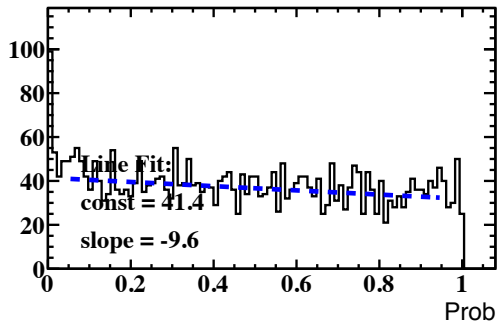
Mass constrain on D^0

$\bar{p}p \to D^{*+}D^{*-} \to D^0\pi^+\bar{D}^0\pi^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

Treefit QA

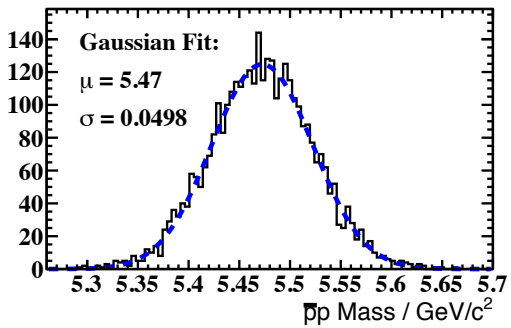


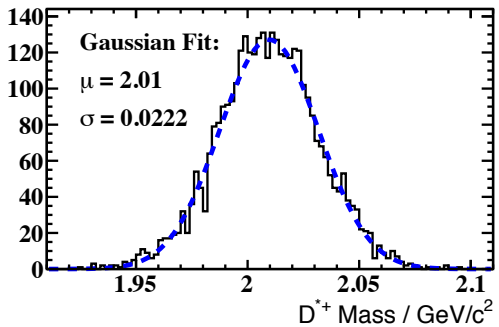


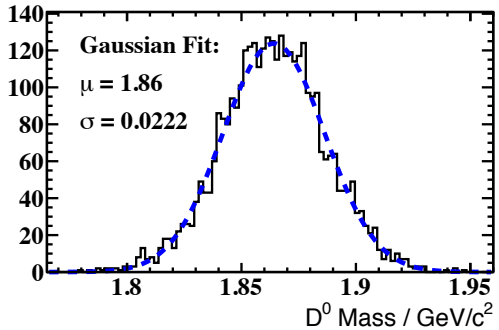


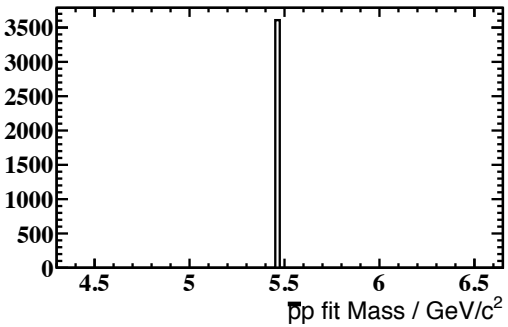
$$\bar{p}p \to D^{*+}D^{*-} \to D^0\pi^+\bar{D}^0\pi^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$$

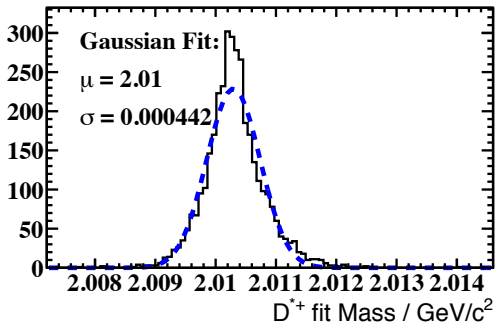
Treefit Masses

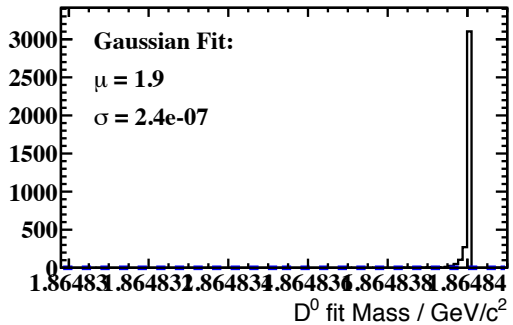






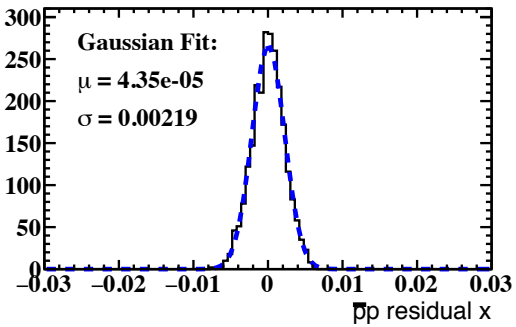


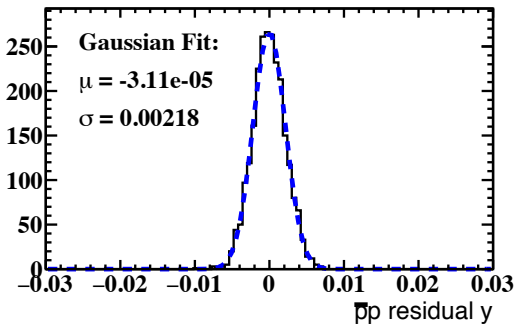


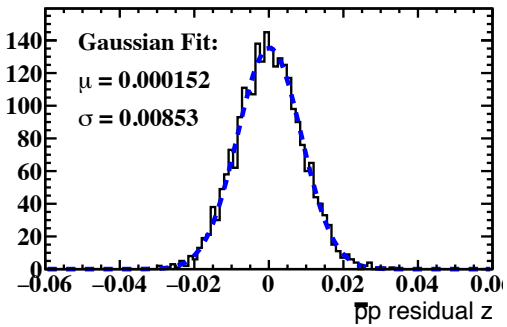


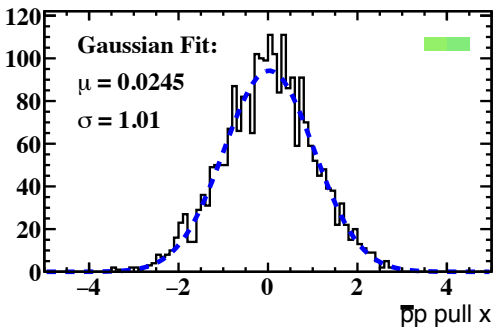
$\bar{p}p \to D^{*+}D^{*-} \to D^0\pi^+\bar{D}^0\pi^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$

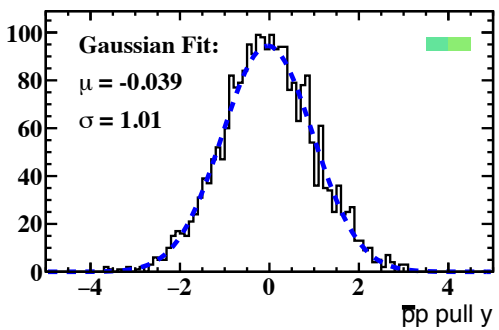
Treefit $\bar{p}p$ Vertex

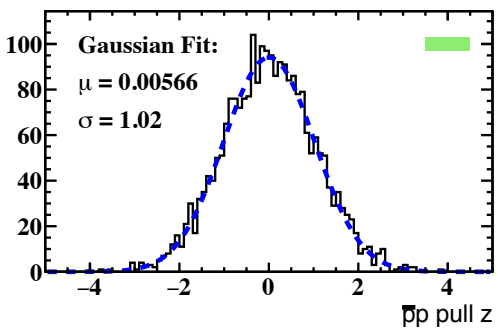






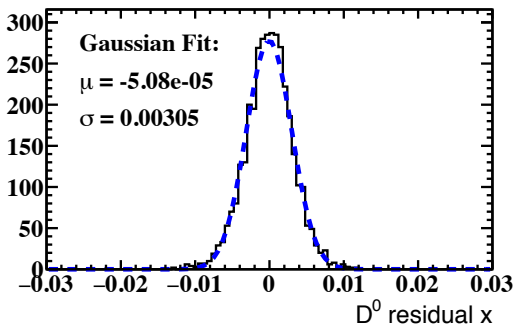


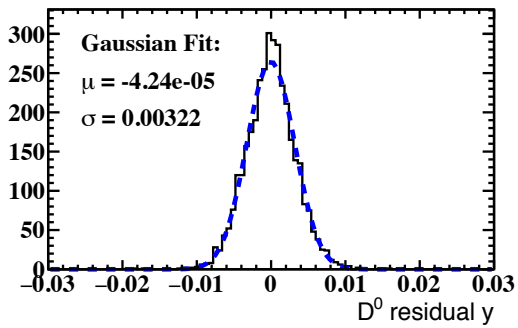


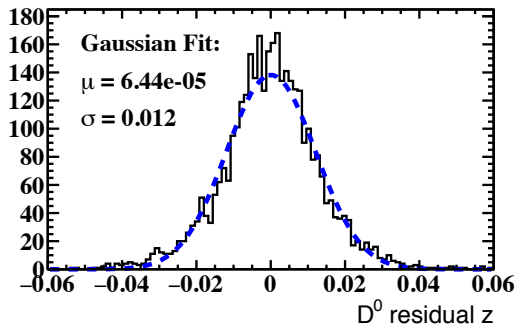


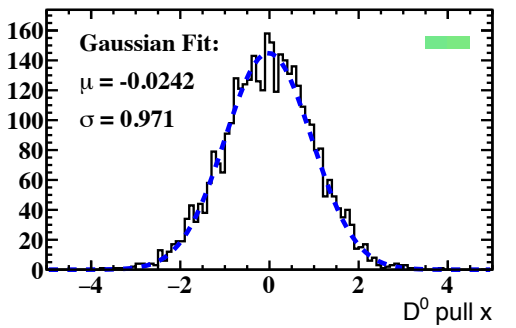
$$\bar{p}p \to D^{*+}D^{*-} \to D^0\pi^+\bar{D}^0\pi^- \to K^-\pi^+\pi^+K^+\pi^-\pi^-$$

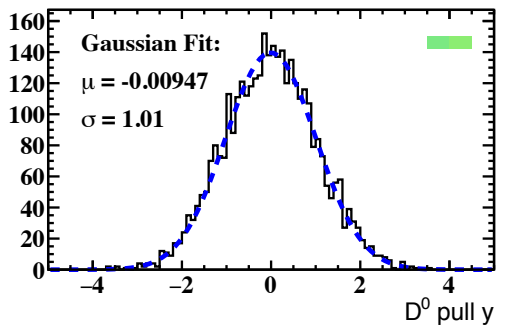
Treefit D^0 Vertex

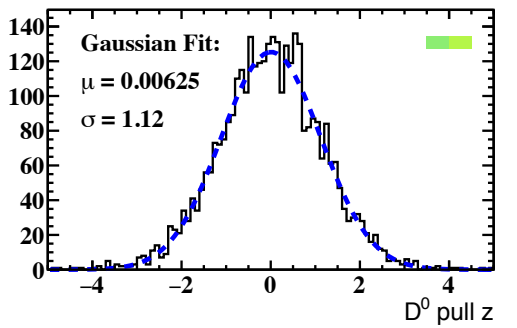






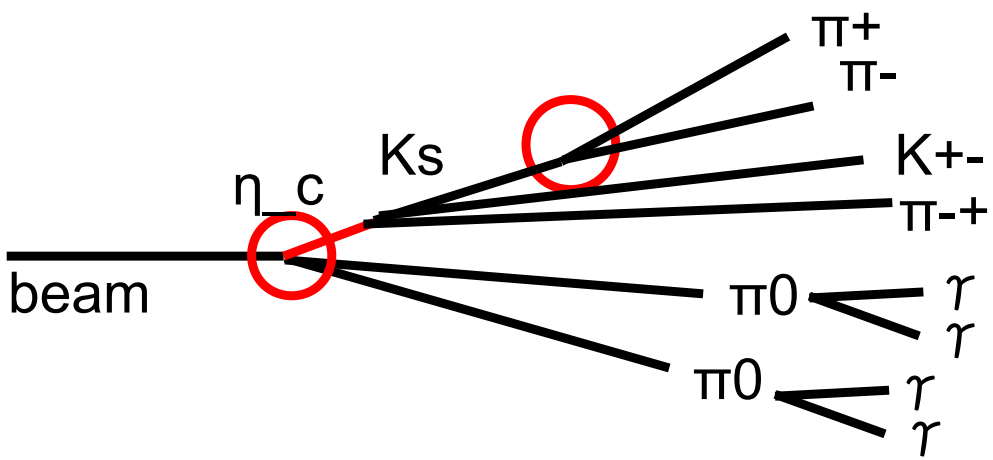






$\bar{p}p \to \eta_c \pi^0 \pi^0 \to K_s K^{\pm} \pi^{\mp} \pi^0 \pi^0$

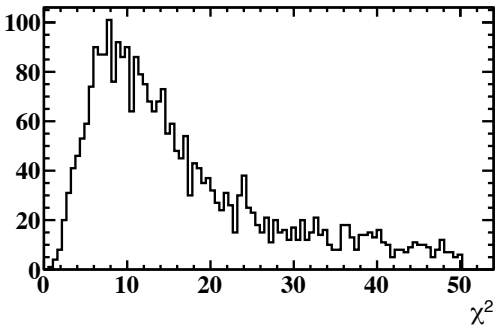
Channel

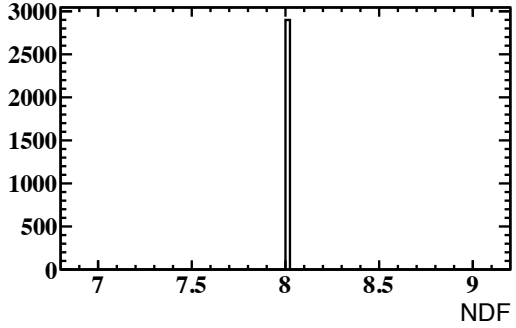


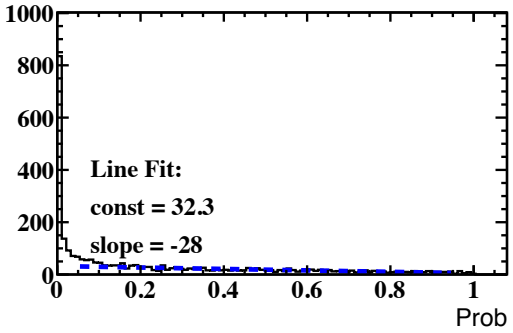
Mass constriain on π^0

$\bar{p}p \to \eta_c \pi^0 \pi^0 \to K_s K^{\pm} \pi^{\mp} \pi^0 \pi^0$

Treefit QA

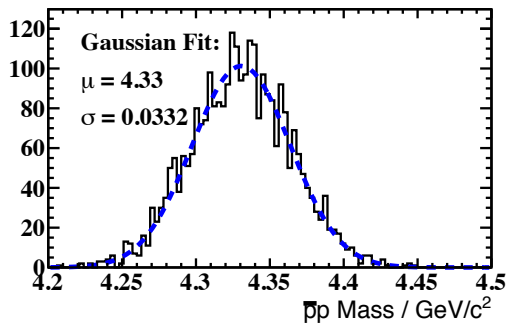


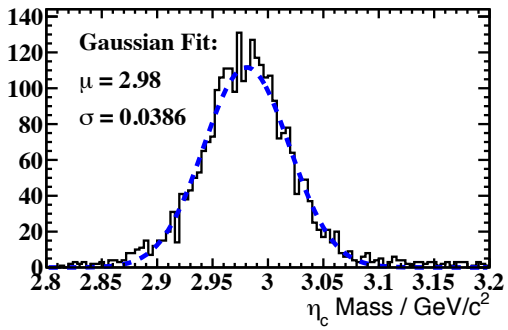


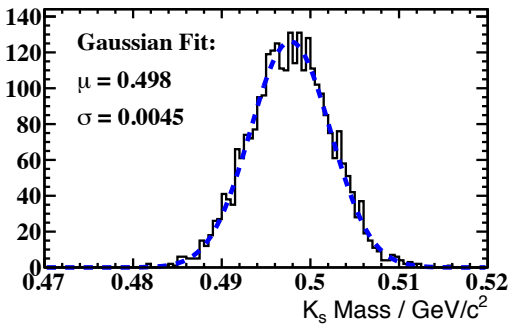


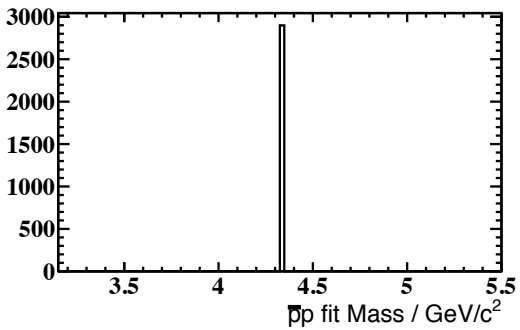
$\bar{p}p \to \eta_c \pi^0 \pi^0 \to K_s K^{\pm} \pi^{\mp} \pi^0 \pi^0$

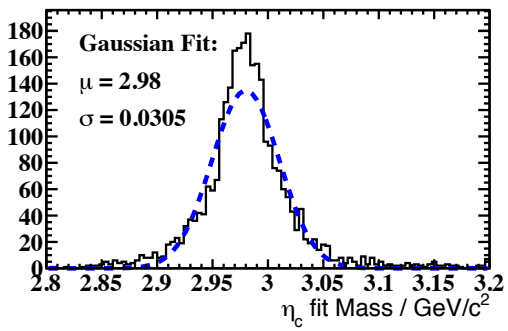
Treefit Masses

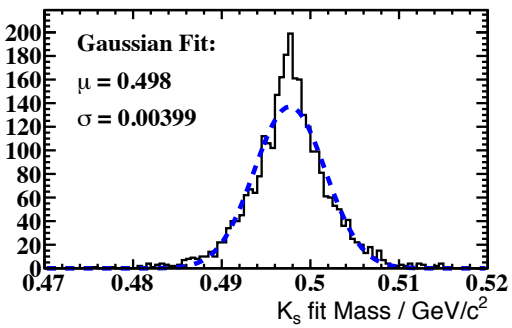






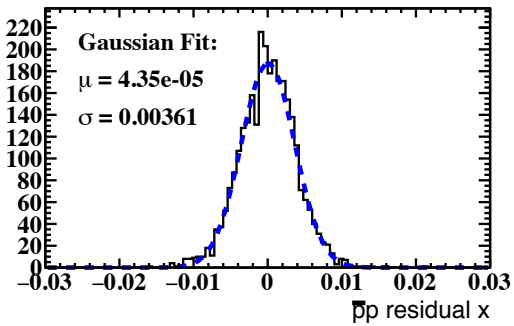


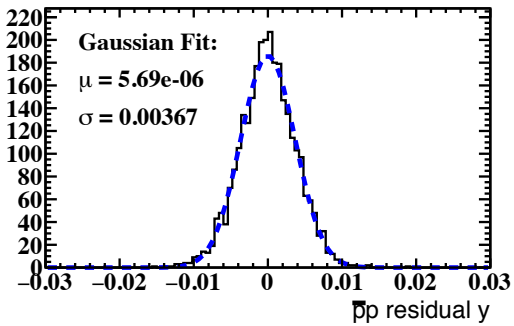


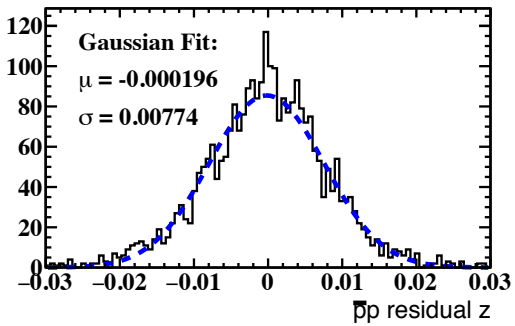


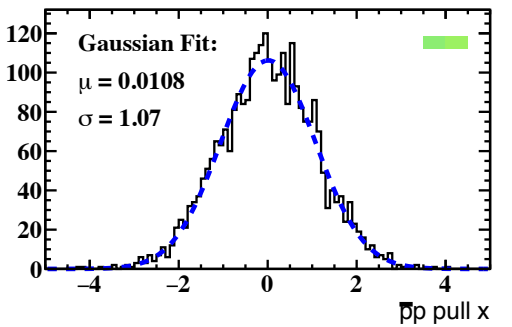
$\bar{p}p \to \eta_c \pi^0 \pi^0 \to K_s K^{\pm} \pi^{\mp} \pi^0 \pi^0$

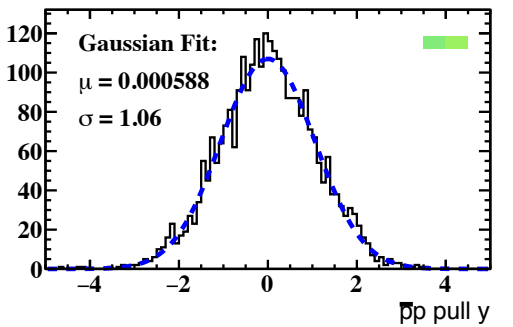
Treefit $\bar{p}p$ Vertex

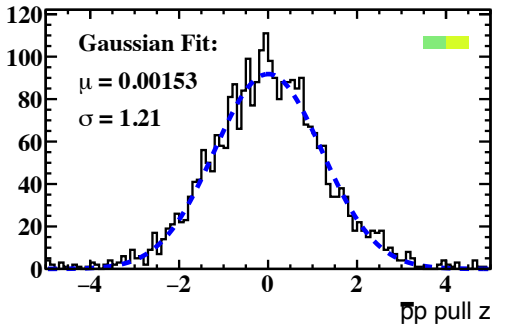






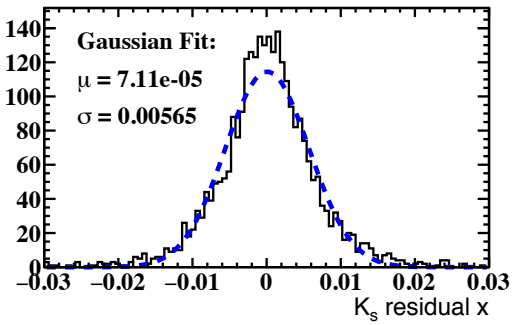


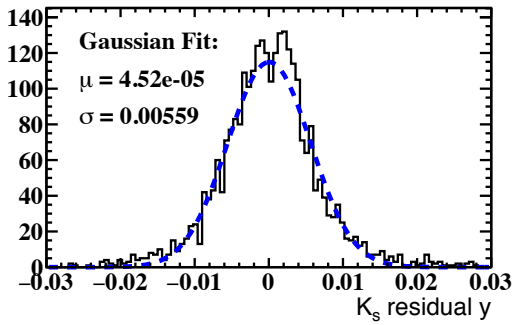


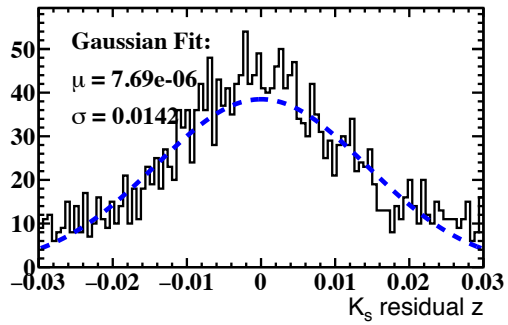


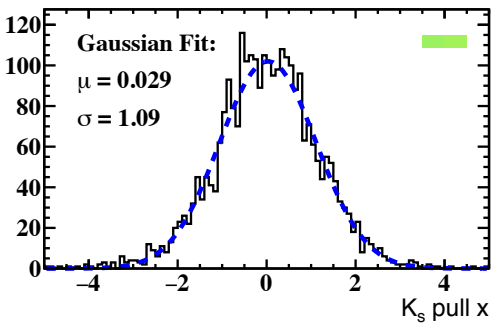
$\bar{p}p \to \eta_c \pi^0 \pi^0 \to K_s K^{\pm} \pi^{\mp} \pi^0 \pi^0$

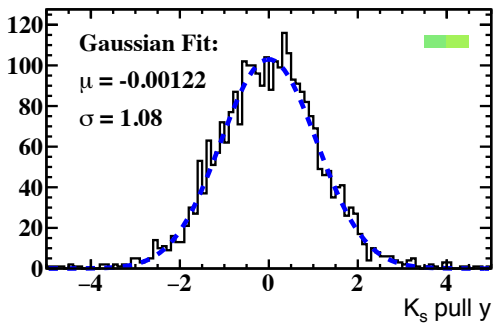
Treefit K_s Vertex

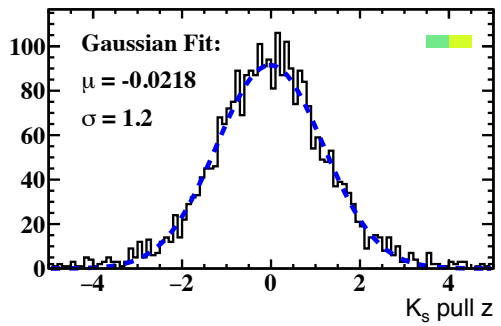












Summary

- ✓ Porting done
- ✓ Looks fine so far
- → Open for users (rev. $>$ 28854)

To Do:

- Remove decay length parameter & constraint
- Test more delayed Vertices (Lambdas)
- Test completely neutral channel
- Test complicated channel with all features
- Compare performance with available fitters
- Look at full simulations
- QA task & tutorial

A funny picture. Thanks for the attention.