KF Particle for the PANDA Experiment

Ivan Kisel[1,2,3] and Maksym Zyzak[1,2,3]

1 – Goethe-Universität Frankfurt, Frankfurt am Main, Germany
2 – Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
3 – GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

LIII PANDA Collaboration Meeting
Upsala, Sweden
09.06.2015

Concept of KF Particle

State vector

Position, momentum and energy

$$r = \{ x, y, z, p_x, p_y, p_z, E \}$$
$$C = \langle r r^T \rangle$$

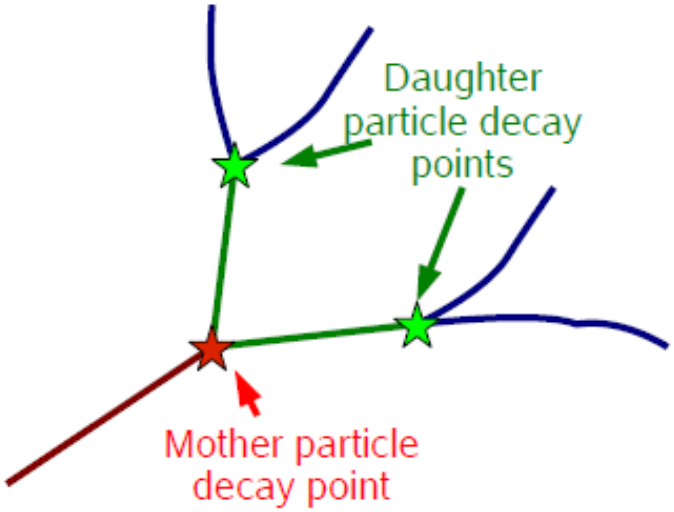
Covariance matrix

Concept:

- Mother and daughter particles have the same state vector and are treated in the same way
- Geometry independent
- Kalman filter based

Functionality of the package:

- Construction of the particles from tracks or another particles
- Decay chains reconstruction
- Transport of the particles
- Simple access to the particle parameters and their errors
- Calculation of the distance to point



Add, construct, propagate complete particles: parameters and their covariance matrices

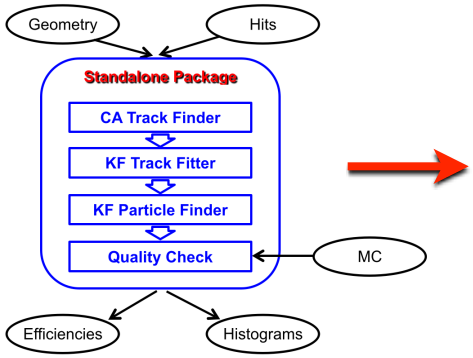
Functionality of KF Particle

Functions	CBM	PANDA	ALICE	STAR
Construction of mother particles	+	+	+	+
Addition and subtraction of the daughter particle to (from) the mother particle	+	+	+	+
+= and -= operators	+	+	+	+
Accessors to the physical parameters (mass, momentum, decay length, lifetime, rapidity, etc)	+	+	+	+
Transport: to an arbitrary point, to the decay and production points, to another particle, to a vertex, on the certain distance	+	+	+	+
Calculation of a distance: to a point, to a particle, to a vertex	+	+	+	+
Calculation of a deviation: from a point, from a particle, from a vertex	+	+	+	+
Calculation of the angle between particles	+	+	+	+
Constraints: on mass, on a production point, on a decay length	+	+	+	+
KF Particle Finder	+	+	+	+

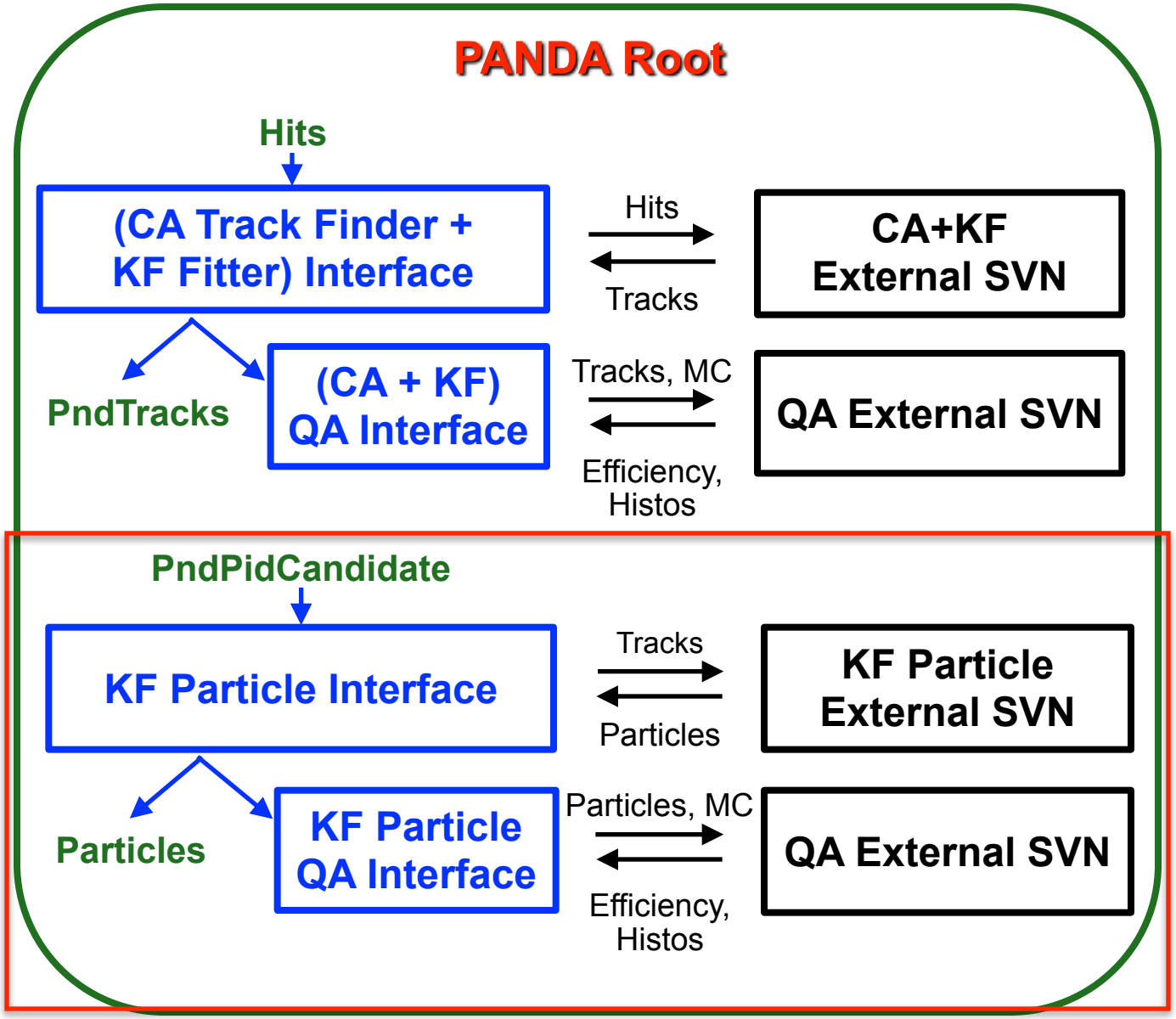
Exactly the same package in all four experiments: CBM, PANDA, ALICE and STAR

Proposed Structure within PANDA Root

We have started within a standalone package



In the PANDA repository



Structure of KF Particle in PANDA Root

Interfaces (folder PandaRoot/kfparticle)

- **PndKFParticleFinder** - runs reconstruction of PV and short-lived particles
- **PndKFParticleFinderPID** - determines the PID for tracks
- **PndKFParticleFinderQA** - collect histograms, calculates efficiency

PandaRoot/kfparticle/KFParticle

Input data:

- **KFPTrack** - track, input for **KFParticle**
- **KFPVertex** - vertex, input for **KFParticle**
- **KFPTrackVector** - array of tracks, input for **KFParticleSIMD**
- **KFPEmcCluster** - array of Emc clusters, input for **KFParticleSIMD**

Classes with mathematics and tasks for analysis:

- **KFParticle** - scalar version
- **KFVertex** - class for PV construction
- **KFParticleSIMD** - vectorised version
- **KFParticlePVReconstructor** - finds PVs
- **KFParticleFinder** - finds short-lived particles
- **KFParticleTopoReconstructor** - prepare tracks for further analysis, runs reconstruction of PV and short-lived particles

PandaRoot/kfparticle/ KFParticlePerformance

- **KFMCTrack** - stores parameters of MC tracks
- **KFMCVertex** - stores parameters of MC vertices
- **KFMCParticle** - stores dependencies between MC tracks KFPartMatch
- **KFPartMatch** - stores matching between reconstructed and MC particles
- **KFPartEfficiencies** - list of the decays to analyse
- **KFTopoPerformance** - calculates efficiencies and collects histograms for the particles listed in **KFPartEfficiencies**

Decay Chains Reconstruction with KF Particle

Reconstruction of a decay chain:

$\Xi^- \rightarrow \Lambda\ \pi^-$
$\quad\quad \hookrightarrow p\ \pi^-$

```
//Convert tracks into KF Particle objects
KFParticle pion1(kfptracks[0], -211); //pi-
KFParticle proton(kfptracks[1], 2212); //proton
KFParticle pion2(kfptracks[2], -211); //pi-

//Construct Lambda-candidate
KFParticle Lambda;
const KFParticle* LambdaDaughters[2] = { &proton, &pion1 };
Lambda.Construct(LambdaDaughters, 2);

//Set a mass constraint on Lambda
Lambda.SetNonlinearMassConstraint(1.115683);

//Reconstruct Xi-
KFParticle Xi;
const KFParticle* XiDaughters[2] = { &Lambda, &pion2 };
Xi.Construct(XiDaughters, 2);
```

Set of Test Macros with KF Particle Finder

- Pure signal (10000 particles) with a momentum of 4 GeV/c:

$D^{0*} \rightarrow D^0\ \pi^0$

$\pi^0 \rightarrow \gamma\gamma$

$D^0 \rightarrow K^0_s\ \pi^+\ \pi^-\ \pi^0$

$K^0_s \rightarrow \pi^+\pi^-$

$\pi^0 \rightarrow \gamma\gamma$

- The ideal track finder with Genfit from the PandaRoot is used to reconstruct tracks.
- MC Primary vertex is used.
- The decay channel has been added to the KF Particle Finder.

A set of macro was added to the trunk version:

PandaRoot/macro/kfparticle/D0Star:

run.sh	a bash script to run the full set
sim_complete.C	set of standard macros
digi_complete.C	
recoideal_complete.C	
pid_complete.C	
kfparticle.C	a macro to run KF Particle Finder

KF Particle Finder

Tracks: $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$, $K^{\pm}$, $p^{\pm}$
secondary and primary

$D^{0*} \rightarrow D^0\ \pi^0$
$\pi^0 \rightarrow \gamma\gamma$
$D^0 \rightarrow K^0_s\ \pi^+\ \pi^-\ \pi^0$
$K^0_s \rightarrow \pi^+\pi^-$, $\pi^0 \rightarrow \gamma\gamma$

Open-charm:
$D^0 \rightarrow K^-\ \pi^+$
$D^0 \rightarrow K^-\ \pi^+\ \pi^+\ \pi^-$
$D^0 \rightarrow K^0_s\ \pi^+\ \pi^-\ \pi^0$
$\bar{D}^0 \rightarrow K^+\ \pi^-$
$\bar{D}^0 \rightarrow K^+\ \pi^+\ \pi^-\ \pi^-$
$D^+ \rightarrow K^-\ \pi^+\ \pi^+$
$D^- \rightarrow K^+\ \pi^-\ \pi^-$
$D_s^+ \rightarrow K^+\ K^-\ \pi^+$
$D_s^- \rightarrow K^+\ K^-\ \pi^-$
$\Lambda_c^+ \rightarrow p\ K^-\ \pi^+$
$\bar{\Lambda}_c^- \rightarrow \bar{p}\ K^-\ \pi^+$

Strange particles:
$K^0_s \rightarrow \pi^+\ \pi^-$
$\Lambda \rightarrow p\ \pi^-$
$\bar{\Lambda} \rightarrow \bar{p}\ \pi^+$

$J/\Psi \rightarrow \mu^+\ \mu^-$

Light vector mesons:
$\rho \rightarrow e^+\ e^-$
$\rho \rightarrow \mu^+\ \mu^-$
$\omega \rightarrow e^+\ e^-$
$\omega \rightarrow \mu^+\ \mu^-$
$\varphi \rightarrow e^+\ e^-$
$\varphi \rightarrow \mu^+\ \mu^-$
$\varphi \rightarrow K^+\ K^-$

Strange resonances
$K^{*0} \rightarrow K^+\ \pi^-$
$\bar{K}^{*0} \rightarrow K^-\ \pi^+$
$\Lambda^* \rightarrow p\ K^-$
$\bar{\Lambda}^* \rightarrow \bar{p}\ K^+$

Gamma
$\gamma \rightarrow e^+\ e^-$

Multi-strange hyperons:
$\Xi^- \rightarrow \Lambda\ \pi^-$
$\bar{\Xi}^+ \rightarrow \bar{\Lambda}\ \pi^+$
$\Omega^- \rightarrow \Lambda\ K^-$
$\bar{\Omega}^+ \rightarrow \bar{\Lambda}\ K^+$

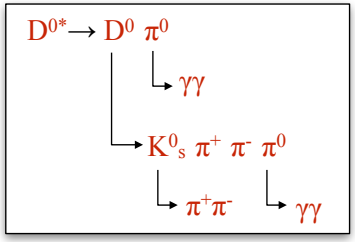
Strange and multi-strange resonances:
$\Sigma^{*+} \rightarrow \Lambda\ \pi^+$
$\bar{\Sigma}^{*+} \rightarrow \bar{\Lambda}\ \pi^-$
$\Sigma^{*-} \rightarrow \Lambda\ \pi^-$
$\bar{\Sigma}^{*-} \rightarrow \bar{\Lambda}\ \pi^+$
$K^{*-} \rightarrow K^0_s\ \pi^-$
$K^{*+} \rightarrow K^0_s\ \pi^+$
$\Xi^{*-} \rightarrow \Lambda\ K^-$
$\bar{\Xi}^{*+} \rightarrow \bar{\Lambda}\ K^+$

Open-charm resonances:
$D^{*0} \rightarrow D^0\ \pi^0$
$D^{*0} \rightarrow D^+\ \pi^-$
$\bar{D}^{*0} \rightarrow D^-\ \pi^+$
$D^{*+} \rightarrow D^0\ \pi^+$
$D^{*-} \rightarrow \bar{D}^0\ \pi^-$

Multi-strange resonances:
$\Xi^{*0} \rightarrow \Xi^-\ \pi^+$
$\bar{\Xi}^{*0} \rightarrow \bar{\Xi}^+\ \pi^-$
$\Omega^{*-} \rightarrow \Xi^-\ K^-\ \pi^+$
$\bar{\Omega}^{*+} \rightarrow \bar{\Xi}^+\ K^+\ \pi^-$

Gamma-decays
$\pi^0 \rightarrow \gamma\ \gamma\ /\ \gamma\ \gamma$
$\eta \rightarrow \gamma\ \gamma$
$\Sigma^0 \rightarrow \Lambda\ \gamma$
$\bar{\Sigma}^0 \rightarrow \bar{\Lambda}\ \gamma$

Structure of kfparticle.C for KF Particle Finder

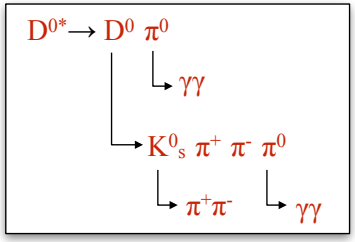



```
...
// ----- PID for KF Particle Finder -----------------------------------------
PndKFParticleFinderPID* kfParticleFinderPID = new PndKFParticleFinderPID();
kfParticleFinderPID->SetPIDMode(1); //MC PID
kfParticleFinderPID->SetMCTrackBranchName("MCTrack");
kfParticleFinderPID->SetChargedTrackBranchName("PidChargedCand");
kfParticleFinderPID->SetNeutralTrackBranchName("PidNeutralCand");
fRun->AddTask(kfParticleFinderPID);

// ----- KF Particle Finder -------------------------------------------
PndKFParticleFinder* kfParticleFinder = new PndKFParticleFinder();
kfParticleFinder->SetChargedTrackBranchName("PidChargedCand");
kfParticleFinder->SetNeutralTrackBranchName("PidNeutralCand");
kfParticleFinder->SetPIDInformation(kfParticleFinderPID);
fRun->AddTask(kfParticleFinder);

// ----- KF Particle Finder QA ----------------------------------------
PndKFParticleFinderQA* kfParticleFinderQA = new PndKFParticleFinderQA("PndKFParticleFinderQA", 0,
                                                kfParticleFinder->GetTopoReconstructor());
kfParticleFinderQA->SetMCTrackBranchName("MCTrack");
kfParticleFinderQA->SetChargedTrackBranchName("PidChargedCand");
kfParticleFinderQA->SetNeutralTrackBranchName("PidNeutralCand");
kfParticleFinderQA->SetPrintEffFrequency(100);
fRun->AddTask(kfParticleFinderQA);
...
```

Output with Efficiencies



After reconstruction the efficiencies are calculated and printed:

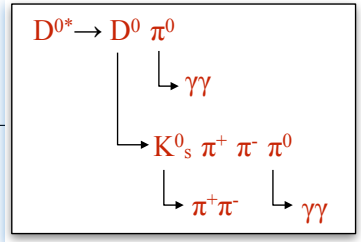
Efficiency in 4π

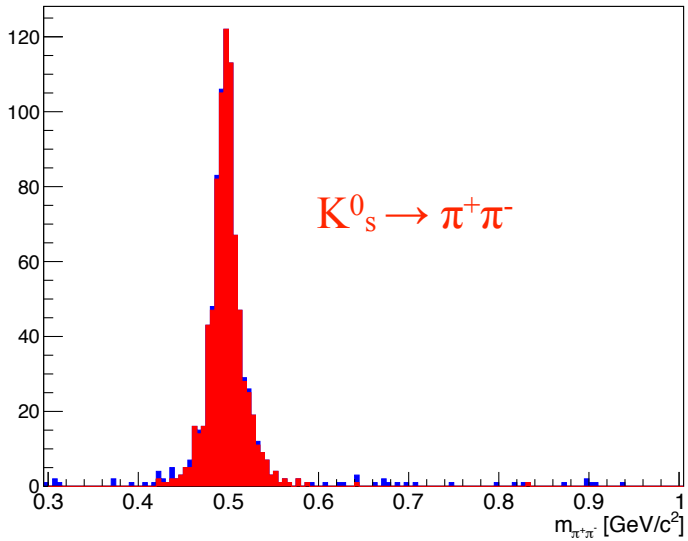
Efficiency of KF Particle

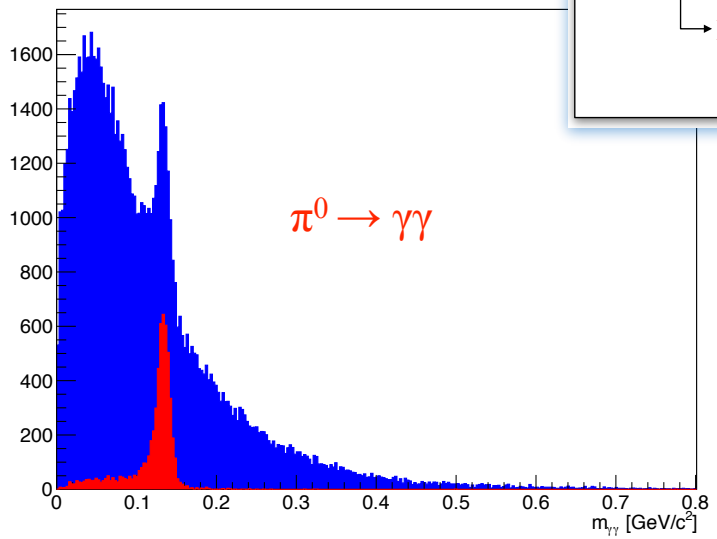
Particle		Eff1	Eff2	Eff3	Ghost	BackGr	N Ghost	N BackGr	N Reco	N Clone	N MC1	N MC2	N MC3
KShort		0.370	-1.000	0.728	0.037	0.020	31	17	786	22	2122	0	1079
KShort	Prim	-1.000	-1.000	-1.000	-1.000	-1.000	0	0	0	0	0	0	0
KShort	Sec	0.370	-1.000	0.728	0.000	0.000	0	0	786	22	2122	0	1079
...													
gamma		0.989	-1.000	1.169	0.307	0.000	6918	4	15628	760	15804	0	13368
gamma	Prim	-1.000	-1.000	-1.000	-1.000	-1.000	0	0	0	0	0	0	0
gamma	Sec	0.989	-1.000	1.169	0.000	0.000	0	0	15628	760	15804	0	13368
Pi0		0.699	-1.000	0.974	0.935	0.002	78126	156	5314	527	7598	0	5458
Pi0	Prim	-1.000	-1.000	-1.000	-1.000	-1.000	0	0	0	0	0	0	0
Pi0	Sec	0.699	-1.000	0.974	0.000	0.000	0	0	5314	527	7598	0	5458
...													
D0_#pi0		0.076	-1.000	0.279	0.839	0.000	803	0	154	11	2024	0	552
D0_#pi0	Prim	-1.000	-1.000	-1.000	-1.000	-1.000	0	0	0	0	0	0	0
D0_#pi0	Sec	0.076	-1.000	0.279	0.000	0.000	0	0	154	11	2024	0	552
...													
D0*_#pi0		0.037	-1.000	0.221	0.820	0.000	324	0	71	7	1917	0	321
D0*_#pi0	Prim	0.037	-1.000	0.221	0.000	0.000	0	0	71	7	1917	0	321
D0*_#pi0	Sec	-1.000	-1.000	-1.000	-1.000	-1.000	0	0	0	0	0	0	0
...													

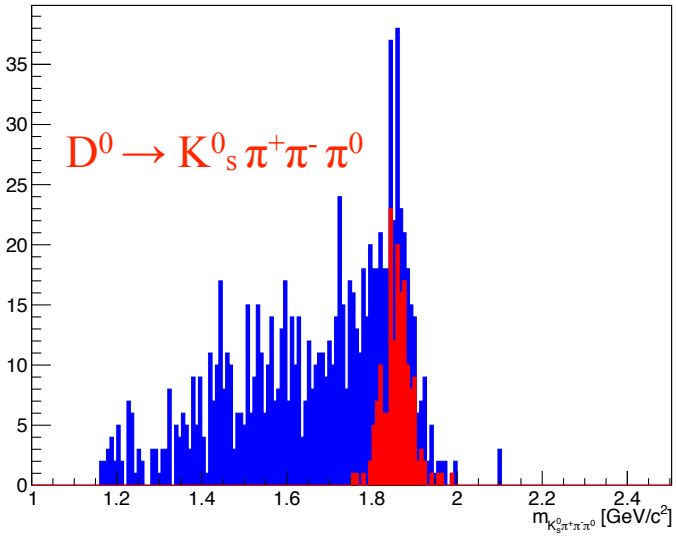
Output Histograms

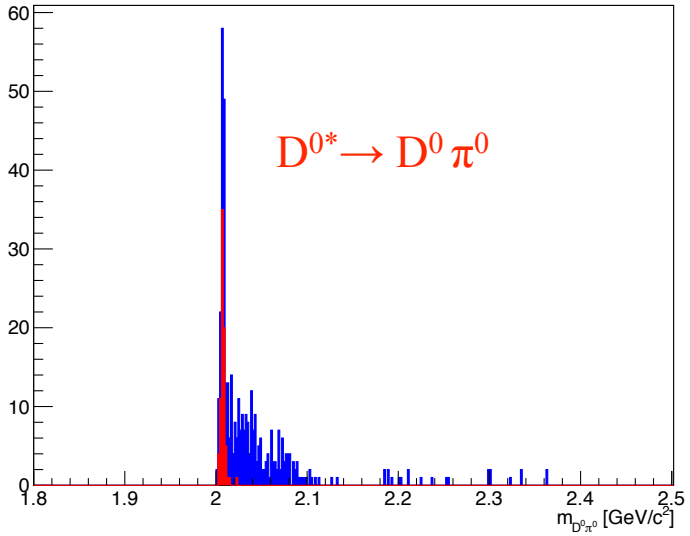
A set of histograms is stored in PndKFParticleFinderQA.root for each particle:











Summary

- ✓ KF Particle is integrated into PANDA Root (thanks to Stefano!)
- ✓ KF Particle has rich functionality to investigate complex decay channels, like shown Ξ^- and D^{0*}.
- ✓ The package together with its interfaces is located in PandaRoot/kfparticle
- ✓ Macros in PandaRoot/macro/kfparticle/D0Star demonstrate how to run KF Particle Finder with more than 50 decays pre-implemented.