

DCA, matching and PID parameterizations pairKK wagon

V. Riabov

PairKK wagon

- Location: mpdroot/physics/pairKK (last updated 06.04.2023)
- Uses centrality from evCentrality wagon, reaction plane from evPlane wagon
- Calculates reconstruction efficiency, mass resolution and invariant mass distributions (both foreground and mixed-event) for $\varphi(1020) \rightarrow KK$ decay
- Event selections:
 - ✓ events with reconstructed vertex by TPC, $|z\text{-vertex-TPC}| < 130$ cm
 - ✓ events with centrality 0-91% (events required to have centrality)
- General track selection:
 - ✓ charged tracks reconstructed in the TPC
 - ✓ tracks matched to PV by $DCA_{x,y,z} \rightarrow DCA$ parameterization for $n \cdot \sigma$ selections
 - ✓ tracks identified as kaons by $dE/dx \rightarrow dE/dx$ parameterization for $e/\pi/K/p$ for $n \cdot \sigma$ selections
 - ✓ tracks identified as kaons by TOF-beta if matched to the TOF \rightarrow TOF matching parameterization
 \rightarrow TOF-beta parameterization for $e/\pi/K/p$
for $n \cdot \sigma$ selections

DCA parameterization

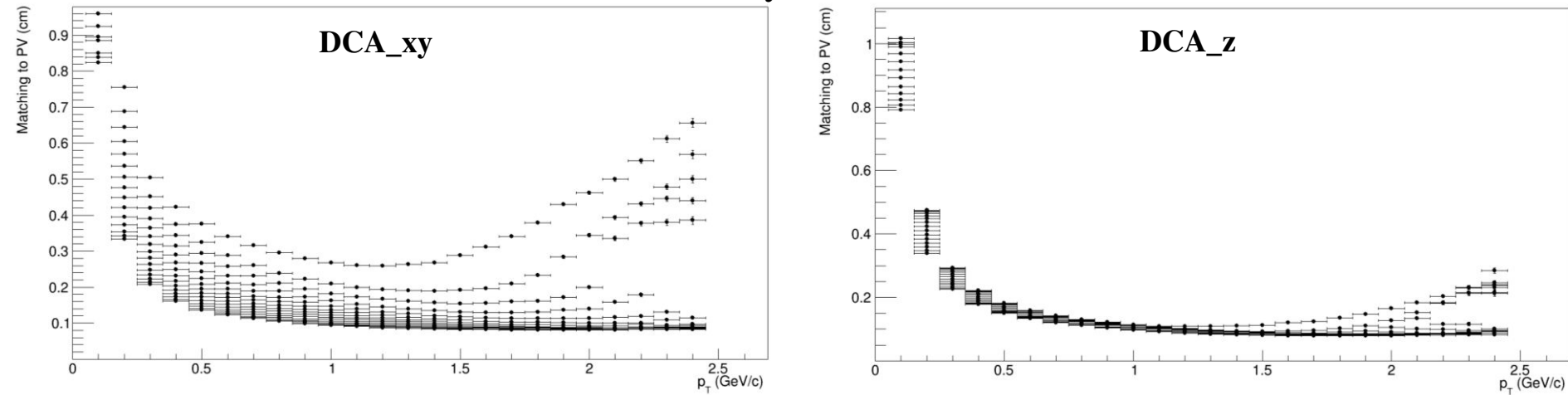
DCA selections

- DCA_{x,y,z} $n\cdot\sigma$ selections \rightarrow reject tracks not from the primary vertex (conversion, weak decays, secondary interactions etc.)
- DCA selections are p_T , rapidity and centrality dependent \rightarrow parameterization of the mean and width of DCA distributions (signalization) vs. p_T , rapidity and centrality \rightarrow apply $n\cdot\sigma$ cuts for selection of primary tracks
- Signalization of DCA is done using the inclusive sample of reconstructed charged particle tracks (mostly pions, although composition changes with momentum and centrality)
- Track selection cuts for parameterization:
 - ✓ number of TPC hits > 10
- DCA_{x,y} and DCA_z distributions are accumulated differentially:
 - ✓ 30 bins in η : $-1.5 < \eta < 1.5$
 - ✓ 10 centrality bins: 0 – 100%
 - ✓ 25 p_T bins: 0.05 – 2.55 GeV/c
- Number of bins and ranges are driven by available statistics
- Processed 5M events for the parameterization (Request 25 UrQMD production)

DCA fits

- DCA_{xy} and DCA_z distributions are fit to a single Gaussian in the range from [-2;2] cm to [-1,1] cm depending on p_T for each centrality- η - p_T bin, (10x30x25)
- Extracted mean values of DCA_{xy} and DCA_z distributions are found to be consistent with zero
- Extracted width values of DCA_{xy} and DCA_z distributions are plotted as a function of p_T for each centrality- η bin, (10x30)

DCA_{xy} (left) and DCA_z (right) widths as a function of particle p_T for 30 η -bins
Centrality bin: 0-10%



- p_T -dependence for each centrality- η bin is fit to a polynomial, (10x30)
- Fits are valid and used only in the fit ranges, $p_T \leq 2.5$ GeV/c
- All distributions and fits are stored in file: mpdroot/physics/pairKK/macros/DCA.root

Implementation of DCA selections

```
155 cout << "[MpdPairKK]: Reading DCA parameterizations ... " << endl;
156
157 // Read-out DCA parameterization
158 dcaFile = new TFile("DCAs.root", "READ");
159
160 for (Int_t etab = 0; etab < neta_bins; etab++) {
161     for (Int_t centb = 0; centb < ncent_bins; centb++) {
162         f_dca_xy[etab][centb] = (TF1 *)dcaFile->Get(Form("dcaxy_fitf_eta_%d_cent_%d", etab, centb));
163         f_dca_z[etab][centb] = (TF1 *)dcaFile->Get(Form("dcaz_fitf_eta_%d_cent_%d", etab, centb));
164     }
165 }

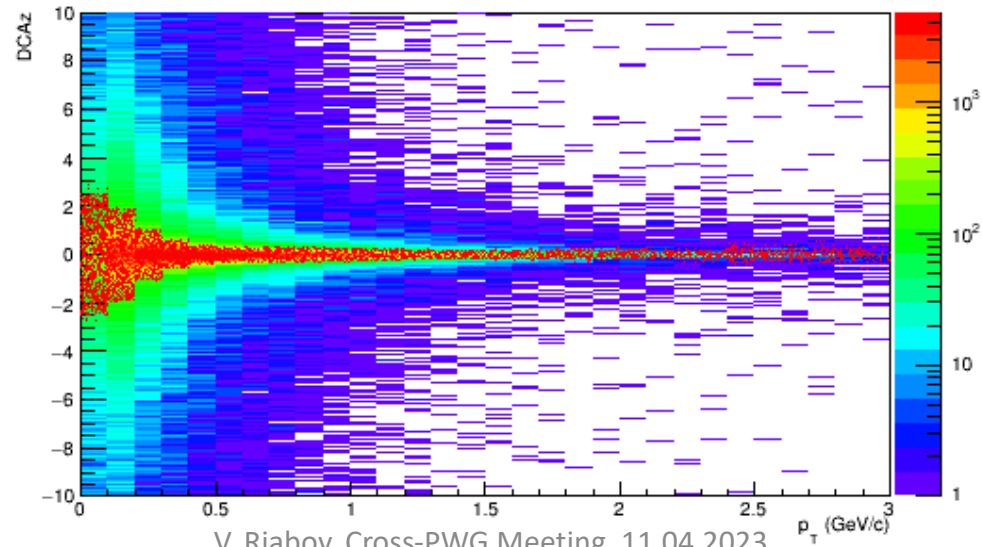
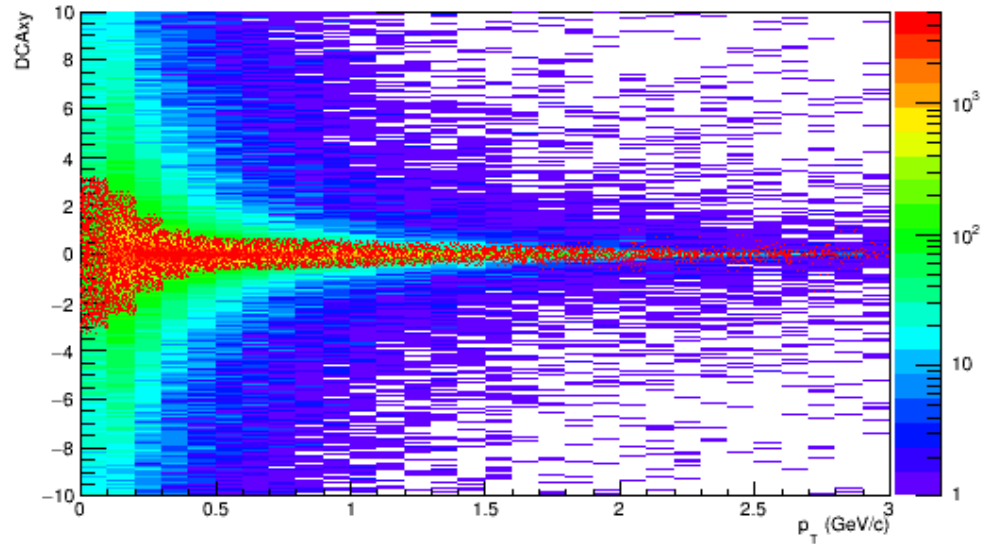
```

```
585 TF1 sigma_fit_XY = *f_dca_xy[eta_bin][cent_bin];
586 TF1 sigma_fit_Z = *f_dca_z[eta_bin][cent_bin];
587
588 // use lower/upper limits outside of parameterization ranges
589 float pt_dca = pt;
590 if (pt_dca < 0.05) pt_dca = 0.05;
591 double xfirst, xlast;
592 sigma_fit_Z.GetRange(xfirst, xlast);
593 if (pt_dca > xlast) pt_dca = xlast + 0.05;
594
595 float sigma_exp_xy = sigma_fit_XY(pt_dca);
596 float sigma_exp_z = sigma_fit_Z(pt_dca);
597
598 float dcax_sig, dcay_sig, dcaz_sig;
599
600 if (sigma_exp_xy != 0) {
601     dcax_sig = mpdtrack->GetDCAX() / sigma_exp_xy;
602     dcay_sig = mpdtrack->GetDCAY() / sigma_exp_xy;
603 } else {
604     dcax_sig = -999;
605     dcay_sig = -999;
606 }
607
608 if (sigma_exp_z != 0) {
609     dcaz_sig = mpdtrack->GetDCAZ() / sigma_exp_z;
610 } else {
611     dcaz_sig = -999;
612 }
613
614 if (fabs(dcax_sig) > mParams.mDCACut) return false; // |DCAX| < 2*sigmaxy(pT, eta, centrality).
615 if (fabs(dcay_sig) > mParams.mDCACut) return false; // |DCAY| < 2*sigmaxy(pT, eta, centrality).
616 if (fabs(dcaz_sig) > mParams.mDCACut) return false; // |DCAZ| < 2*sigmaz(pT, eta, centrality).

```

Results

- Total and 2σ selected DCA_{xy} and DCA_z distributions



dE/dx parameterization

Implementation

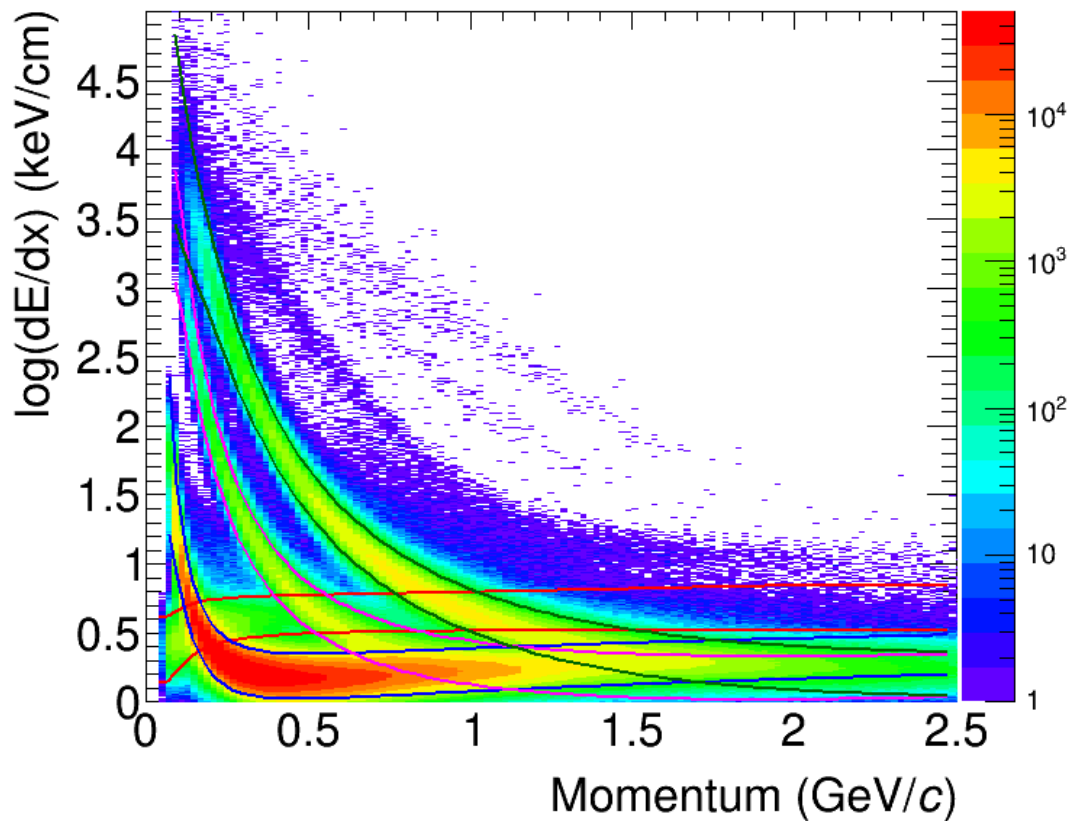
- Selected tracks:
 - ✓ hits > 20
 - ✓ $|\eta| < 1$
 - ✓ $|\text{DCA}_{x,y,z}| < 3 \sigma$

```
291 MpdTpcKalmanTrack *tr = (MpdTpcKalmanTrack *)mKalmanTracks->UncheckedAt(i);
331 if (fabs(dEdx_sigma_K(tr->GetDedx()), pmom) < mParams.mPIDsigTPC) isK_TPC = 1;
```

```
641 // dE/dx parameterizations for e/π/K/p
642 float MpdPairKK::dEdx_sigma_EL(float dEdx, float mom) const
643 {
644     if (mom < 0.04) return -999;
645     if (dEdx <= 0) return -999;
646
647     dEdx = log(dEdx);
648
649     if (mom < 0.06) mom = 0.06;
650     if (mom > 2.0) mom = 2.0;
651
652     float mean[7] = {6.869878e-003, -1.573146e-001, 1.847371e+000, -9.253678e-001,
653                    1.073907e+000, -3.394239e+000, 6.451762e-001};
654     float width[7] = {-4.359368e+006, -8.508504e-012, -3.958364e-009, 1.526816e-009,
655                      1.353776e-011, 3.426352e-009, 6.591542e-002};
656
657     float mean_exp, width_exp;
658
659     mean_exp =
660         mean[0] / mom / mom *
661         (mean[1] * log(mom * mom) - mean[2] * mom * mom - mean[3] * mom - mean[4] - mean[5] * mom * mom * mom) +
662         mean[6];
663     width_exp =
664         width[0] / mom / mom *
665         (width[1] * log(mom * mom) - width[2] * mom * mom - width[3] * mom - width[4] - width[5] * mom * mom * mom) +
666         width[6];
667
668     return (dEdx - mean_exp) / width_exp;
669 }
```

Results

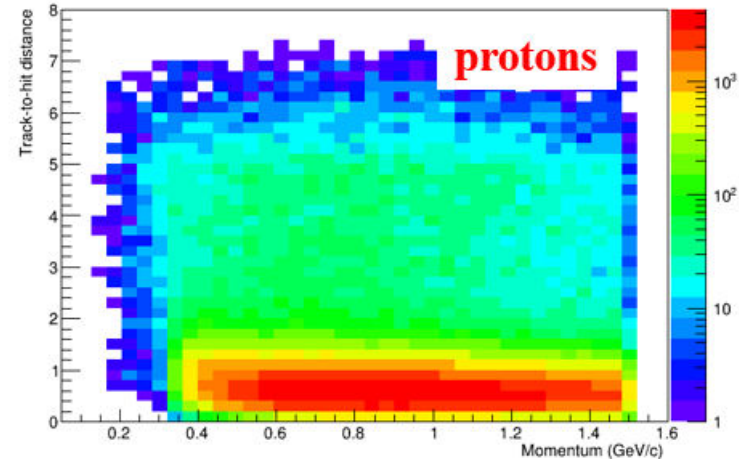
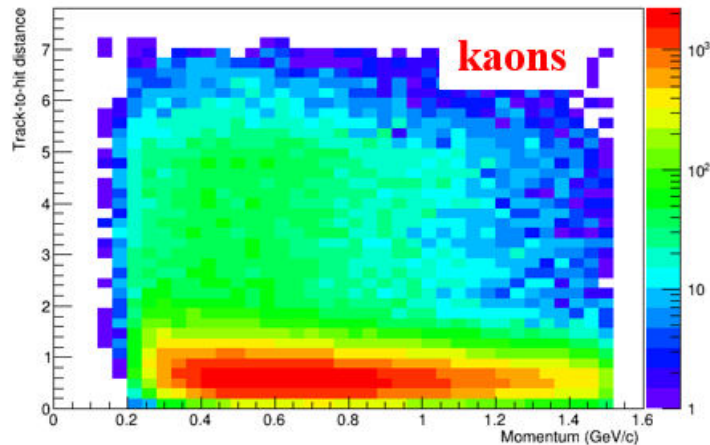
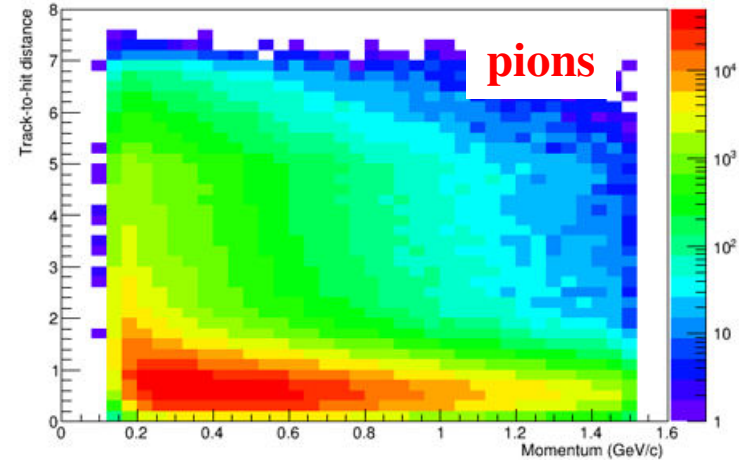
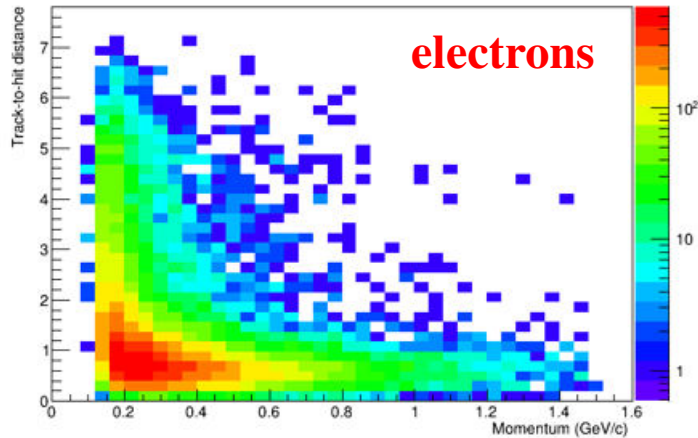
- Selected tracks:
 - ✓ hits > 20
 - ✓ $|\eta| < 1$
 - ✓ $|DCA_{x,y,z}| < 3 \sigma$
- Parameterized $\log(dE/dx)$ vs. momentum
- 2σ bands for **electrons**, **pions**, **kaons**, **protons**



Track-to-TOF matching

Default track-to-TOF matching

- Track is matched if $\text{mpdtrack} \rightarrow \text{GetTofFlag}() == 2 \parallel \text{mpdtrack} \rightarrow \text{GetTofFlag}() == 6$
- Track-to-hit distance in the TOF (or $1/\text{weight}$) vs. p_T , minbias BiBi@9.2



- Track is matched if distance is < 7 cm \rightarrow too wide ???

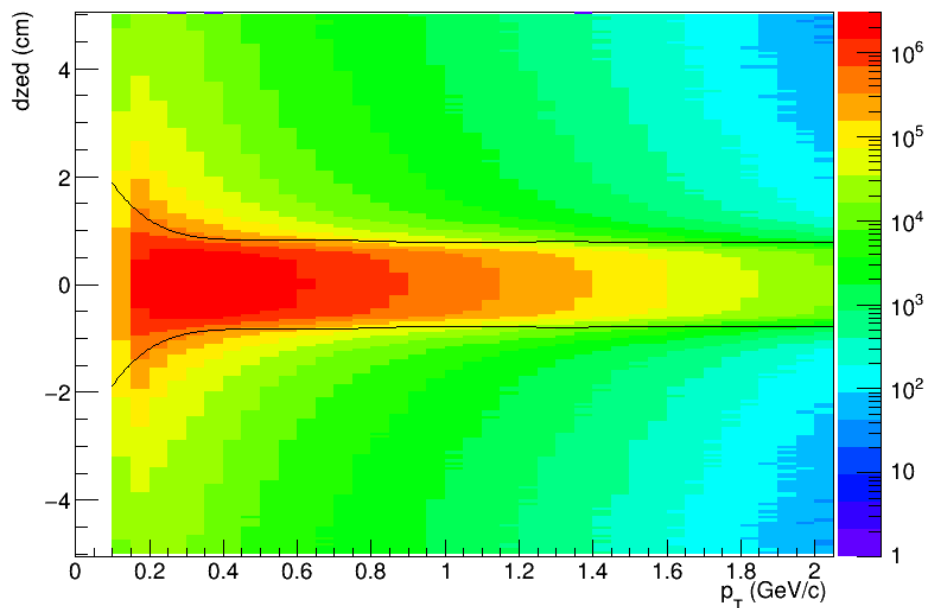
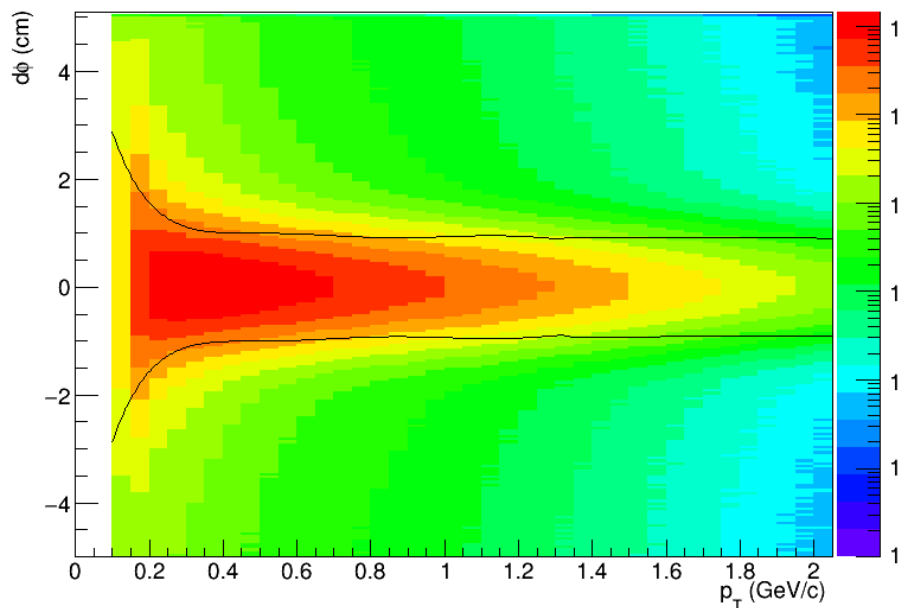
Implementation

```
325     if (matchingIndex > 0) {
326         MpdToFMatchingData *Matching = (MpdToFMatchingData *)mpdToFMatching->At(matchingIndex);
327         if (TestToFMatch(chARGE, fabs(mpdTrack->GetPt()), Matching->GetdPhi(), Matching->GetdZed()) == 1) isTOF = 1;
328     }
329 }
```

```
759 int MpdPairKK::TestToFMatch(int charge, float pt, float dphi, float dz) const
760 {
761     if (pt < 0.1) return 0;
762
763     float meanphiCoeff[7] = {2.745900e-001, -1.437619e+000, 2.824655e+000, -2.644983e+000,
764                             1.269623e+000, -3.012448e-001, 2.816063e-002};
765     float widthphiCoeff[7] = {3.211516e+000, -2.325625e+001, 8.024492e+001, -1.407389e+002,
766                               1.320427e+002, -6.304419e+001, 1.202106e+001};
767     float widthzCoeff[7] = {1.646438e+000, -9.744003e+000, 3.054315e+001, -4.799731e+001,
768                             3.994061e+001, -1.680153e+001, 2.810887e+000};
769
770     float meanphi, widthphi, meanz, widthz;
771
772     if (pt < 3.0) {
773         meanphi = meanphiCoeff[0] + meanphiCoeff[1] * pt + meanphiCoeff[2] * pt * pt + meanphiCoeff[3] * pt * pt * pt +
774                 meanphiCoeff[4] * pt * pt * pt * pt + meanphiCoeff[5] * pt * pt * pt * pt * pt +
775                 meanphiCoeff[6] * pt * pt * pt * pt * pt * pt;
776     } else {
777         meanphi = 0.135;
778     }
779
780     if (charge < 0) meanphi = -meanphi;
781
782     if (pt < 1.325) {
783         widthphi = widthphiCoeff[0] + widthphiCoeff[1] * pt + widthphiCoeff[2] * pt * pt +
784                 widthphiCoeff[3] * pt * pt * pt + widthphiCoeff[4] * pt * pt * pt * pt +
785                 widthphiCoeff[5] * pt * pt * pt * pt * pt + widthphiCoeff[6] * pt * pt * pt * pt * pt * pt;
786     } else {
787         widthphi = 0.454 + 0.00815163 * pt;
788     }
789
790     meanz = 0.0;
791
792     if (pt < 1.5) {
793         widthz = widthzCoeff[0] + widthzCoeff[1] * pt + widthzCoeff[2] * pt * pt + widthzCoeff[3] * pt * pt * pt +
794                 widthzCoeff[4] * pt * pt * pt * pt + widthzCoeff[5] * pt * pt * pt * pt * pt +
795                 widthzCoeff[6] * pt * pt * pt * pt * pt * pt;
796     } else {
797         widthz = 0.39844 - 0.00660409 * pt;
798     }
799
800     if (fabs(dphi - meanphi) / widthphi < 3.0 && fabs(dz - meanz) / widthz < 3.0) return 1;
801
802     return 0;
803 }
```

Results

- Selected tracks:
 - ✓ hits > 20
 - ✓ $|\eta| < 1$
 - ✓ $|DCA_{x,y,z}| < 3 \sigma$
- Split *distance* to dphi and dzed and then parameterized matching distributions for all charged tracks vs. p_T



- 2σ bands are shown with black lines
- Signalized TOF matching cuts are optional for each analysis

Beta-TOF parameterization

Implementation

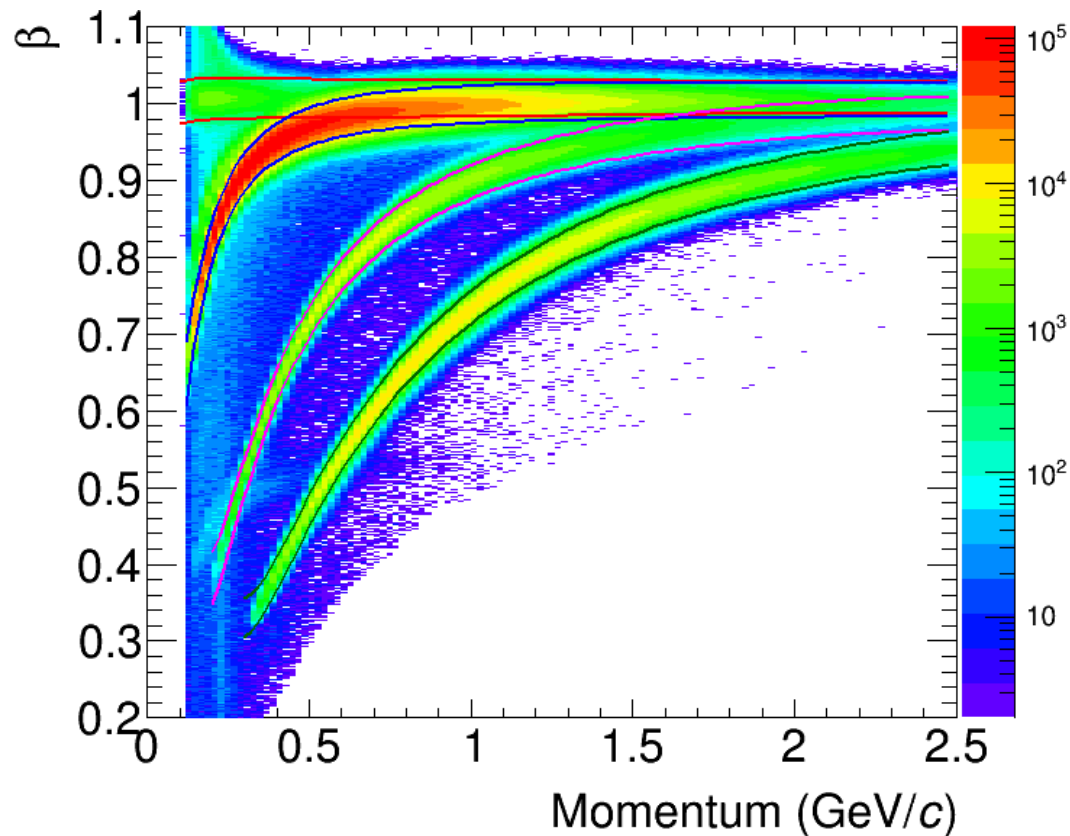
- Selected tracks:
 - ✓ hits > 20
 - ✓ $|\eta| < 1$
 - ✓ $|\text{DCA}_{x,y,z}| < 3 \sigma$

```
332     if (isTOF == 1 && fabs(Beta_sigma_K(mpdtrack->GetTofBeta(), pmom)) < mParams.mPIDsigTOF) isK_TOF = 1;
```

```
829 float MpdPairKK::Beta_sigma_Pi(float beta, float mom) const
830 {
831     if (mom < 0.12) return -999;
832     if (mom > 4.0) mom = 4.0;
833
834     float mean[7] = {3.150000e+003, -1.248103e-006, 1.108943e+000, -7.829288e-006,
835                    7.832307e-006, -4.780232e-007, 3.494167e+003};
836     float width[7] = {-3.587374e+007, 1.391129e-012, 3.658606e-011, -2.025494e-011,
837                     -7.243473e-013, -3.150429e-011, 1.273604e-002};
838
839     float mean_exp, width_exp;
840
841     mean_exp =
842         mean[0] / mom / mom *
843         (mean[1] * log(mom * mom) - mean[2] * mom * mom - mean[3] * mom - mean[4] - mean[5] * mom * mom * mom) +
844         mean[6];
845     width_exp =
846         width[0] / mom / mom *
847         (width[1] * log(mom * mom) - width[2] * mom * mom - width[3] * mom - width[4] - width[5] * mom * mom * mom) +
848         width[6];
849
850     return (beta - mean_exp) / width_exp;
851 }
```


Results

- Selected tracks:
 - ✓ hits > 20
 - ✓ $|\eta| < 1$
 - ✓ $|DCA_{x,y,z}| < 3 \sigma$
- Parameterized beta-TOF vs. momentum
- 2σ bands for **electrons**, **pions**, **kaons**, **protons**



Conclusions

- DCA, matching, dE/dx and beta-TOF parameterizations for $e/\pi/K/p$ are available in pairKK wagon
- Ideally we should use centralized n-sigma parameterized variables \rightarrow subject of further developments for the train
- Please provide other examples of the centralized parameterizations for the future developments