Measure of CP violation in B+/- meson

Adriano Del Vincio, (562946)

April 2, 2023

Lo studio dell'asimmetria *CP* rappresenta una delle maggiori aree di indagine nella fisica delle alte energie. Diverse collaborazioni in passato (*BaBar* presso lo SLAC, *Belle* presso KEK) hanno misurato l'asimmetria nel comportamento tra materia/antimateria in differenti canali di decadimento, come i mesoni B_0 e $\overline{B_0}$. In questo progetto si è analizzato il decadimento dei mesoni carichi in $3K$, utilizzando dati collezionati a LHCb nel 2011.

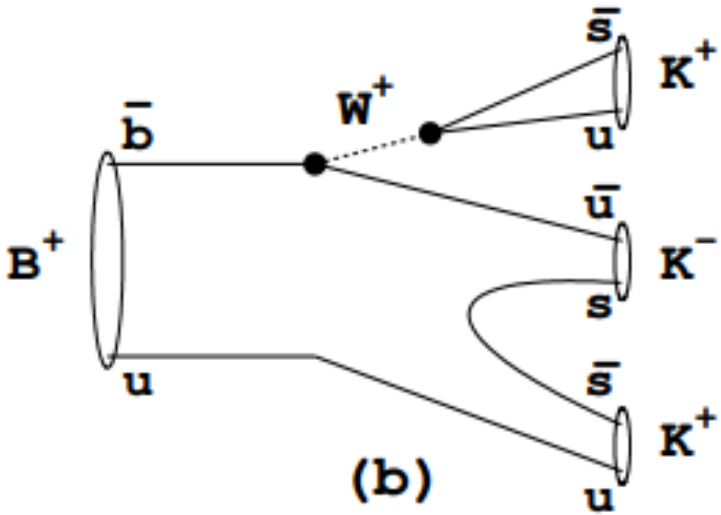


Figure: Decadimento del mesone $B+$ in tre kaoni, l'asimmetria è dovuta al cambiamento di flavour del quark b

Dataset

I dati a disposizione sono suddivisi in due TTree, che differiscono per l'orientazione del campo magnetico nell'esperimento, e sono analizzati separatamente. I due File contengono 25 variabili che descrivono la cinematica del processo. I dati sono analizzati principalmente con *RDataFrame*. L'analisi consiste nel selezionare gli eventi che provengono dal decadimento di interesse, rigettango eventi di fondo o le particelle che non possono essere identificate come Kaoni. Una volta selezionati gli eventi, si genera il Dalitz plot del decadimento e si rimuovono le risonanze che non sono di interesse. L'obiettivo dell'analisi è quello di ottenere una misura dell'asimmetria tra materia/antimateria nel decadimento, formalmente definita come:

$$A_{CP}(B^{\pm} \to f^{\pm}) = \frac{\Gamma(B^{+} \to f^{+}) - \Gamma(B^{-} \to f^{-})}{\Gamma(B^{+} \to f^{+}) + \Gamma(B^{-} \to f^{-})} \tag{1}$$

Compute the invariant mass

Per ricostruire il decadimento, è necessario calcolare la massa invariante dei mesoni B, a partire dalle impulso dei $3K$. La massa invariante è calcolata nello script `invmass.cpp`, utilizzando la funzione inline *invMass*. Nella funzione si è esplicitato il modulo quadro del quadrimpulso totale:

$$P^{\mu}_{tot} P_{tot,\mu} = (k_1 + k_2 + k_3)^2$$

```
//for the invariant mass
auto invMass = [] (double p1x, double p1y, double p1z, double p2x, double p2y, double p2z, double p3x, double p3y, double p3z){
double KaonMass = 493.677; // MeV/c**2
ROOT::Math::PxPyPzMVector k1(p1x,p1y,p1z, KaonMass);
ROOT::Math::PxPyPzMVector k2(p2x,p2y,p2z, KaonMass);
ROOT::Math::PxPyPzMVector k3(p3x,p3y,p3z, KaonMass);
double invariant = k1.M2() + k2.M2() + k3.M2() + 2*k1.Dot(k2) + 2*k1.Dot(k3) + 2*k2.Dot(k3);
return TMath::Sqrt(invariant);};
```

Per il calcolo, si è sfruttato la classe `ROOT::Math::LorentzVector`, che ha già implementato al suo interno i metodi per calcolare la massa invariante di un quadrivettore.

DataSelection

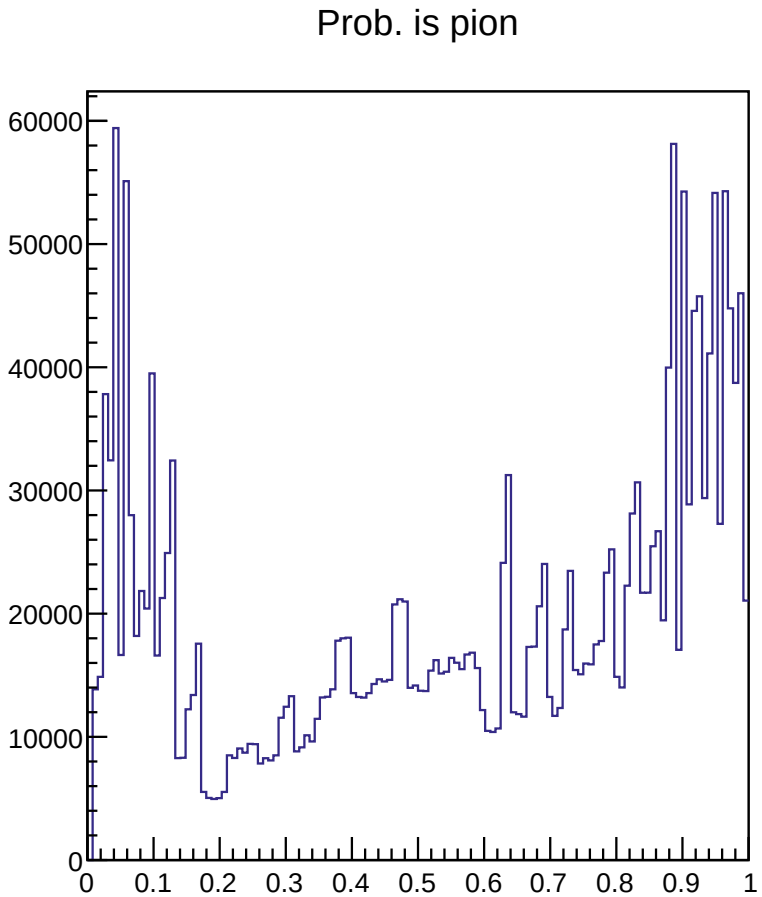
Sono definite anche altre quantità importanti, come energia, impulso del candidato B a partire dall'energia e impulso dei kaoni. Si filtrano quegli eventi in cui una o più delle particelle è identificata come un muone. Infine si calcola la carica del candidato B

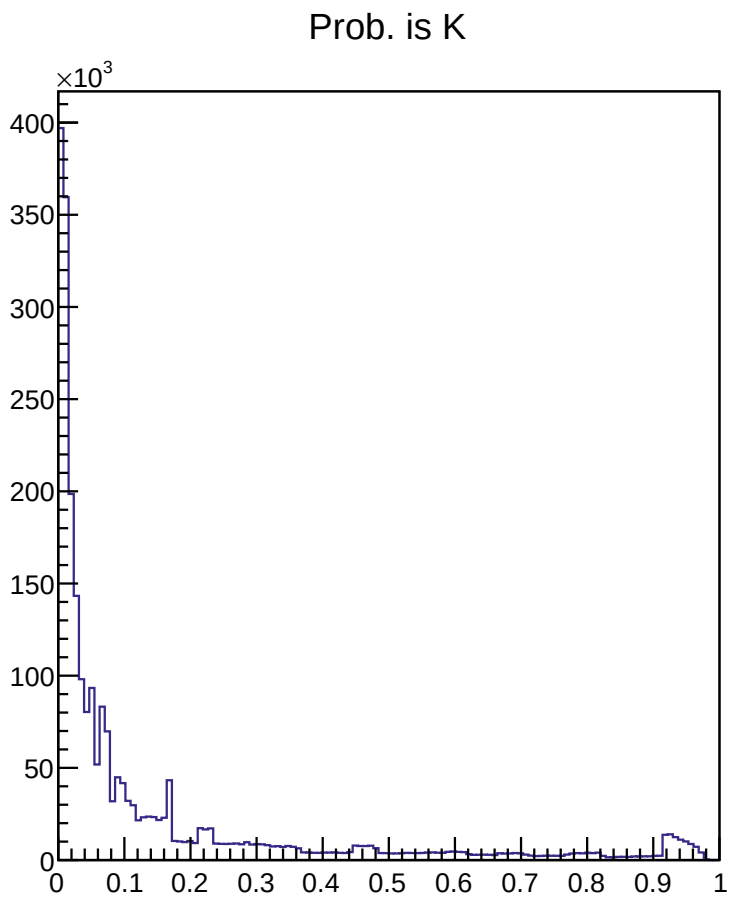
```
auto rdf_down1 = rdf_down.Define("invMass", invMass, {"H1_PX","H1_PY","H1_PZ","H2_PX","H2_PY","H2_PZ","H3_PX","H3_PY","H3_PZ"})
                 .Define("H1_E",energyKaon, {"H1_PX","H1_PY","H1_PZ"})
                 .Define("H2_E",energyKaon, {"H2_PX","H2_PY","H2_PZ"})
                 .Define("H3_E",energyKaon, {"H3_PX","H3_PY","H3_PZ"})
                 .Define("B_PX","H1_PX + H2_PX + H3_PX")
                 .Define("B_PY","H1_PY + H2_PY + H3_PY")
                 .Define("B_PZ","H1_PZ + H2_PZ + H3_PZ")
                 .Define("B_E","H1_E + H2_E + H3_E")
                 .Filter("!H1_isMuon && !H2_isMuon && !H3_isMuon")
                 .Filter("H1_E <= 1e6 && H2_E <= 1e6 && H2_E <= 1e6")
                 .Filter("B_E  <= 1e6");

auto Bselection_up = rdf_up1.Define("MassB",invB, {"B_PX","B_PY","B_PZ","B_E"});
auto Bselection_down = rdf_down1.Define("MassB",invB, {"B_PX","B_PY","B_PZ","B_E"});
```

DataSelection

Nello script `globalAsymmetry.cpp` si effettuano ulteriori tagli riguardanti l'identificazione delle particelle. Si escludono quelle particelle che hanno probabilità $> 50\%$ di essere Pioni (che sono interessanti per un altro canale di decadimento) e si selezionano solo le particelle che hanno probabilità $> 50\%$ di essere kaoni.



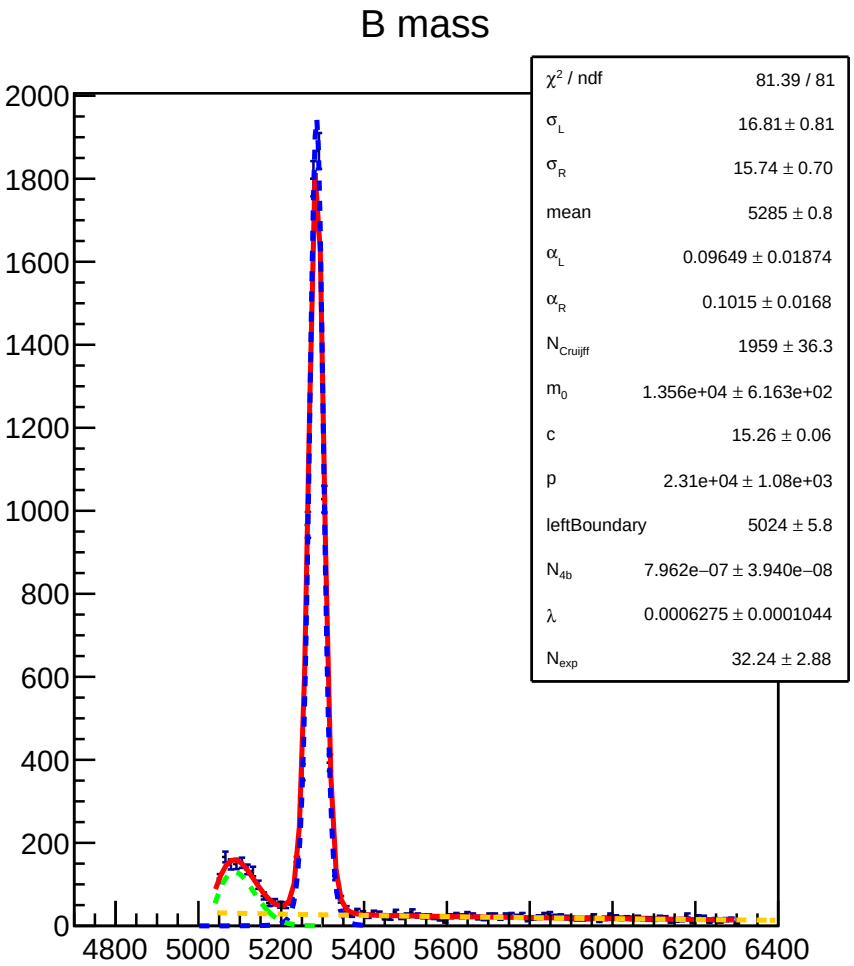


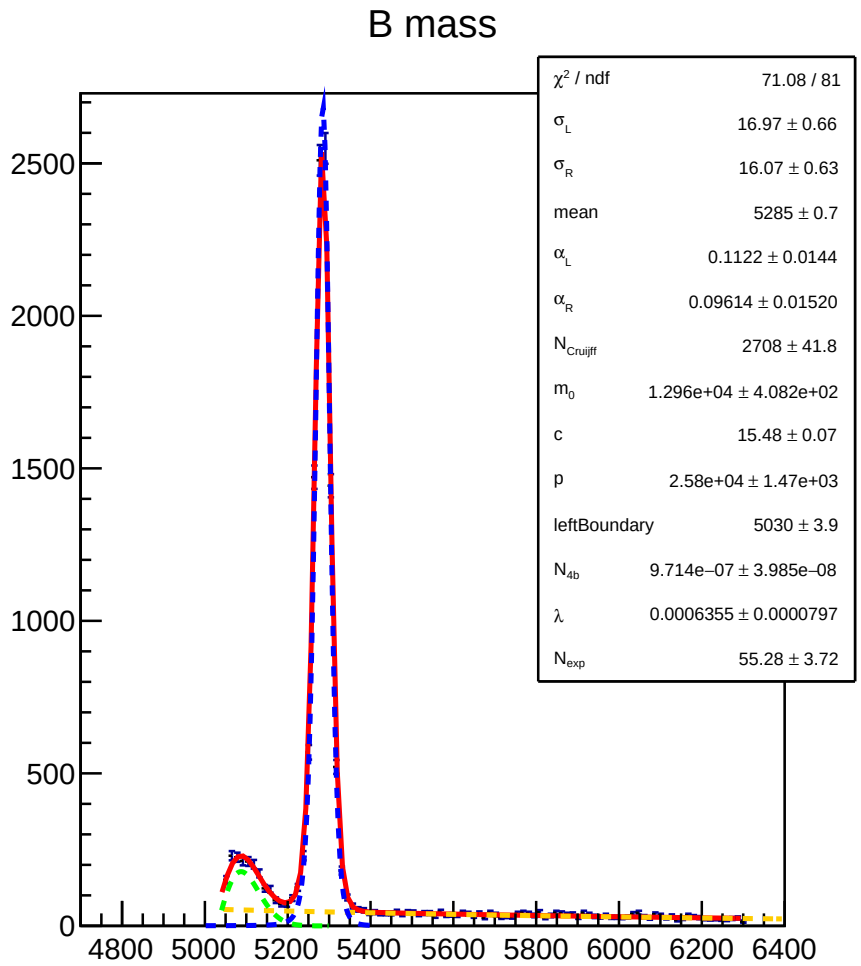
Fit mass B candidates

Nello stesso script `globalAsymmetry.cpp` si effettua il fit alla massa invariante dei candidati B. Per Modellizzare il segnale si utilizza si utilizza la Cruijff function, una gaussiana asimmetrica nelle code, per tenere in conto dei fotoni emessi ISR e FSR. Il fondo è modellizzato dalla somma di un'esponenziale e da un'ARGUS function, che descrive il caso del decadimento in 4 corpi, di cui solo 3 sono effettivamente "visti" dai detector.

$$Argus(m, m_0, c, p) = N \cdot m \cdot \left[1 - \left(\frac{m}{m_0}\right)^2\right]^p \cdot exp\left[c \cdot \left(1 - \left(\frac{m}{m_0}\right)^2\right]\right. \tag{2}$$

$$Signal(N, \mu, \sigma_{L,R}, \alpha_{R,L}) = N \cdot exp\left[\frac{-(x-\mu)^2}{2\sigma_{L,R}^2 + \alpha_{L,R}(x-\mu)^2}\right] \tag{3}$$


B mass
χ^2 / ndf 81.39 / 81
σ_L 16.81 ± 0.81
σ_R 15.74 ± 0.70
mean 5285 ± 0.8
α_L 0.09649 ± 0.01874
α_R 0.1015 ± 0.0168
$N_{Cruijff}$ 1959 ± 36.3
m_0 1.356e+04 ± 6.163e+02
c 15.26 ± 0.06
p 2.31e+04 ± 1.08e+03
leftBoundary 5024 ± 5.8
N_{4b} 7.962e−07 ± 3.940e−08
λ 0.0006275 ± 0.0001044
N_{exp} 32.24 ± 2.88
0 200 400 600 800 1000 1200 1400 1600 1800 2000
4800 5000 5200 5400 5600 5800 6000 6200 6400


B mass
χ^2 / ndf 71.08 / 81
σ_L 16.97 ± 0.66
σ_R 16.07 ± 0.63
mean 5285 ± 0.7
α_L 0.1122 ± 0.0144
α_R 0.09614 ± 0.01520
$N_{Cruijff}$ 2708 ± 41.8
m_0 1.296e+04 ± 4.082e+02
c 15.48 ± 0.07
p 2.58e+04 ± 1.47e+03
leftBoundary 5030 ± 3.9
N_{4b} 9.714e−07 ± 3.985e−08
λ 0.0006355 ± 0.0000797
N_{exp} 55.28 ± 3.72
0 500 1000 1500 2000 2500
4800 5000 5200 5400 5600 5800 6000 6200 6400

Dal fit si ricavano i parametri che descrivono il segnale ed il fondo. Per eliminare gli eventi di fondo si è quindi selezionato un intervallo intorno al picco della massa invariante. Gli estremi dell'intervallo sono stati calcolati in `optimalCut.py`. Massimizzando il rapporto tra segnale e rumore:

$$\frac{S(x,y)}{\sqrt{S(x,y)+B(x,y)}}$$

Il codice calcola il numero aspettato di eventi di segnale e di fondo per una data coppia di punti (x,y). Alla fine si selezionano i valori che massimizzano la funzione sopra.

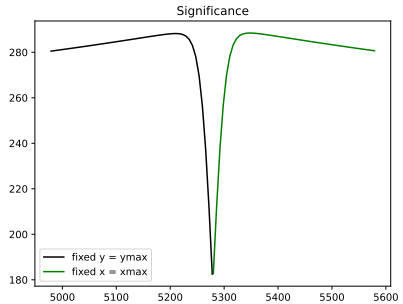


Figure: Frequency versus Voltage