

Measure of CP violation in B meson

Adriano Del Vincio, (562946)

April 2, 2023

Fisica del processo

Studio dell'asimmetria CP nel decadimento in 3 kaoni dei Mesoni B^\pm

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 \bar{B}_0 . In questo progetto si è analizzato il decadimento dei mesoni carichi in $3K$, utilizzando dati collezionati a LHCb nel 2011.

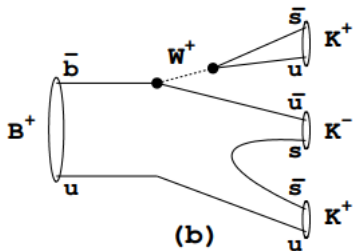


Figure: Decadimento del mesone B^+ in tre kaoni, l'asimmetria è dovuta al cambiamento di flavour del quark b

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, rigettando 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:

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

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:

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

```
//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.

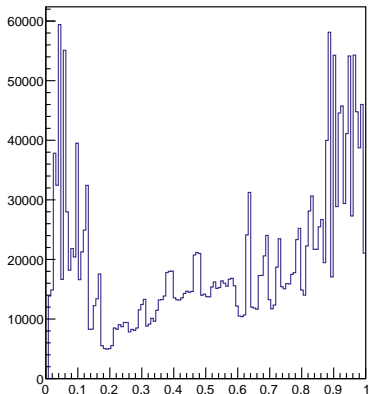
Sono definite anche altre quantità importanti, come energia, impulso del candidato B a partire dall'energia e impulso dei kaoni. Si filtrano quegli eventi in cui una o più delle particelle è identificata come un muone. Infine si calcola la carica del candidato B

```
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 && H3_E <= 1e6")
    .Filter("B_E <= 1e6");

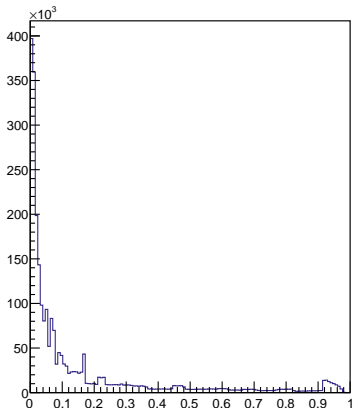
auto Bselection_up = rdf_up1.Define("MassB", invB, {"B_PX", "B_PY", "B_PZ", "B_E"});
auto Bselection_down = rdf_down1.Define("MassB", invB, {"B_PX", "B_PY", "B_PZ", "B_E"});
```

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.

Prob. is pion



Prob. is K

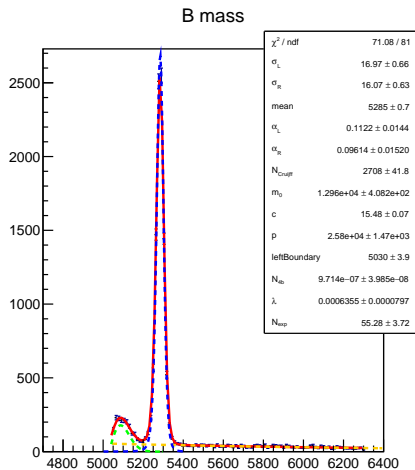
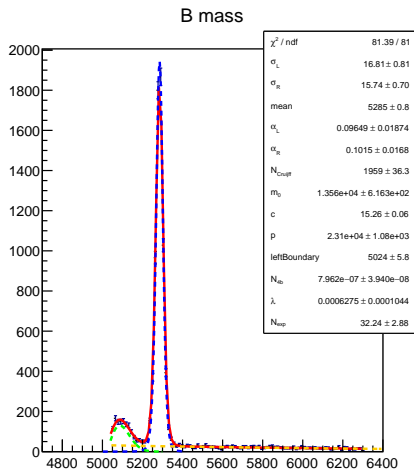


Nello stesso script `globalAsymmetry.cpp` si effettua il fit alla massa invariante dei candidati B . Per Modellizzare il segnale si utilizza la Cruiff function, una gaussiana asimmetrica nelle code, per tenere in conto dei fotoni emessi ISR e FSR. Il fondo è modellizzato dalla somma di un'esponenziale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

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

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

Fit mass B candidates



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 atteso 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.

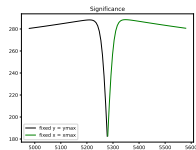


Figure: Frequency versus Voltage

Effettuate le selezioni, si costruisce il Dalitz Plot. I Dalitz plots sono creati tramite `dalitz.cpp`. Il codice calcola le masse invarianti di ogni coppia di kaoni. Le masse invarianti sono calcolate e tramite la funzione

```
//Invariant mass of two particles
auto mxy = [](double bx, double by, double bz, double bE, double kx, double ky, double kz, double kE){
    ROOT::Math::PxPyPzEVector k(kx, ky, kz, kE);
    ROOT::Math::PxPyPzEVector b(bx, by, bz, bE);
    auto c = b - k;
    return c.M2();};
```

A questo punto si plottano le masse invarianti:

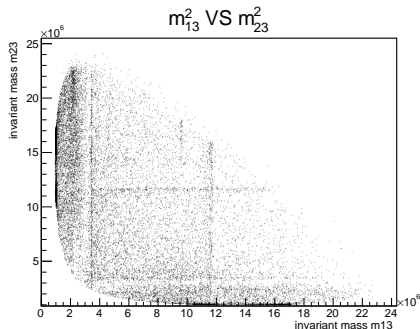
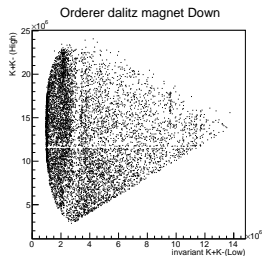
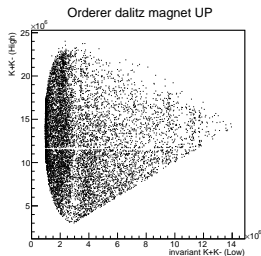
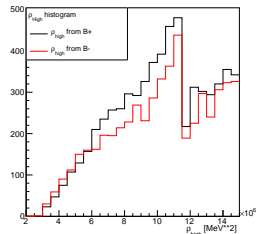
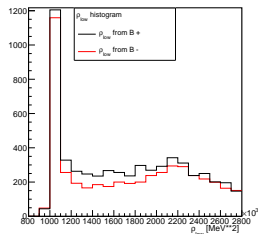


Figure: Masse invarianti dei Kaoni

Asimmetria locale materia/antimateria

A questo punto è utile ordinare le masse delle particelle in m_{low} ed m_{high} , qui si può confrontare le asimmetrie tra materia/antimateria.



L'asimmetria può essere studiata localmente, nel dalitz plot, eseguendo un binning e calcolando per ogni bin A_{CP} . Questo viene eseguito dallo script 2dPlot.py

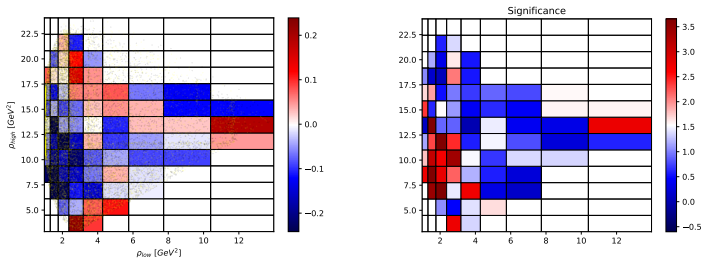


Figure: Local asymmetry on the left, significance (asymmetry divided by the error) on the right