Particle identification using dE/dx and its application for physics

January 14, 1999

1 Energy loss

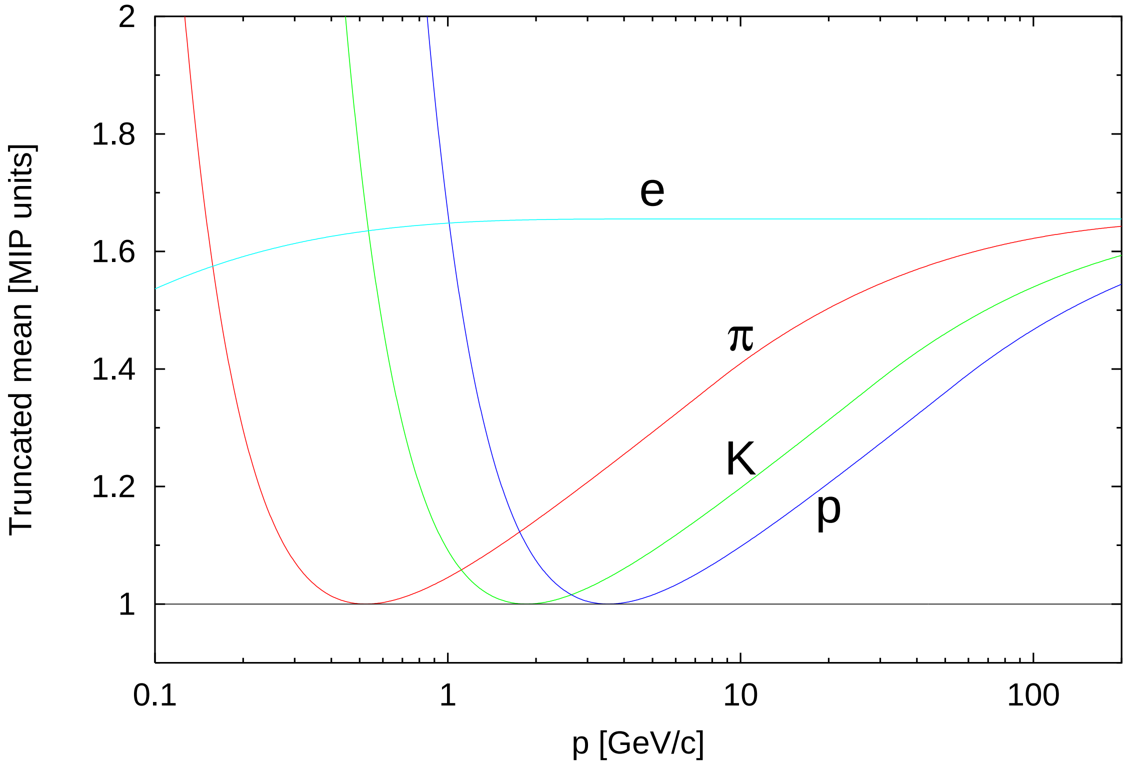


Figure 1: *Bethe-Bloch functions for different particles.*

1.1 Shape of the truncated mean distribution

The charged particle mostly loses its energy in many steps. That is why the energy loss closely follows the Landau-distribution. As we will see later, the determination of the particle yields builds strongly upon the precise knowledge of the shape of this distribution.

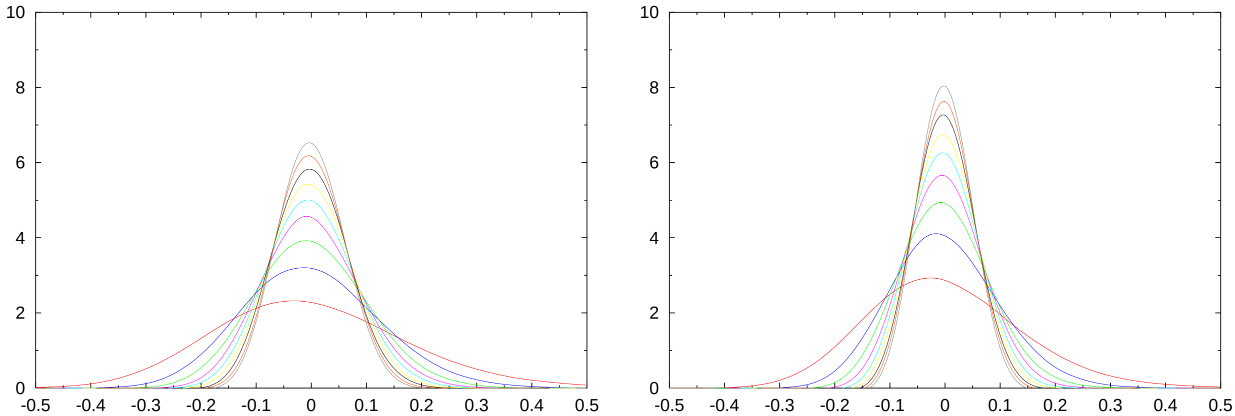


Figure 2: *Truncated mean distributions of Main and Vertex TPCs for tracks with different number of clusters (10,20, ... ,90).*

These normalized shapes for a (p, p_T) box were obtained in the following way.

1.2 Particle yields in a (p, p_T) box

In a given box we have the measured distribution of the truncated mean. By the procedure described above we also got the normalized distribution for all the four particle types, separately. The aim is to determine the particle yields. In the calibration phase the Bethe-Bloch curve was tuned to match all the particles for the largest possible (p, p_T) region. Nevertheless, there is still room for deviations when we go in a small box. Because the calibration is essentially a multiplicative method, we can leave a parameter, a β "Bethe-Bloch scaler", free:

$$dE/dx^{measured}_{\pi,K,p,e} = \beta \cdot dE/dx^{Bethe-Bloch}_{\pi,K,p,e} \tag{1}$$

So the task would be to fit five parameters, the N_π, N_K, N_p, N_e yields and the β scaler for each box. In order to speed up computation and to get rid of the instabilities during fit we propose a much faster and nicer method.

Let us grade the particles in ascending order using their β-scaled Bethe-Bloch values. For instance we have

$$dE/dx_p < dE/dx_K < dE/dx_\pi < dE/dx_e. \tag{2}$$

Using the averages of the ordered numbers, we divide the truncated mean distribution into four ranges. In these ranges we add up the h_i bins of the truncated mean distribution to get the M_π, M_K, M_p, M_e measured yields.

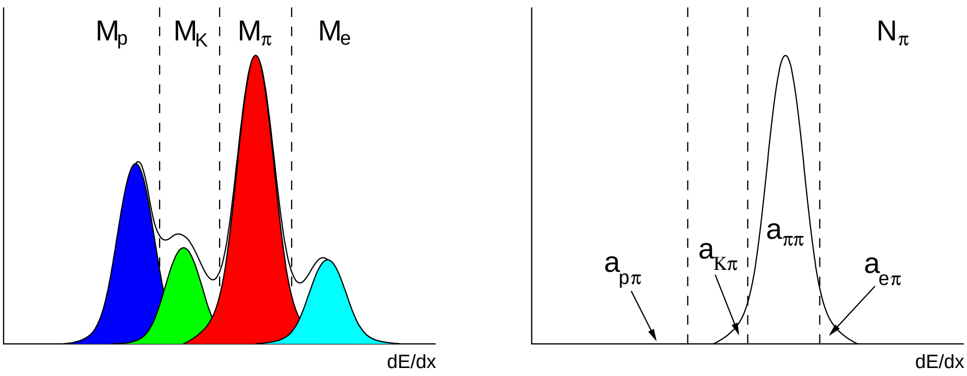


Figure 3: *Measured yields (left) and an example for the definition of the transformation matrix elements (right).*

$$M_p = \sum_{0}^{(dE/dx_p+dE/dx_K)/2} h_i \qquad M_K = \sum_{(dE/dx_p+dE/dx_K)/2}^{(dE/dx_K+dE/dx_\pi)/2} h_i \tag{3}$$

$$M_\pi = \sum_{(dE/dx_K+dE/dx_\pi)/2}^{(dE/dx_\pi+dE/dx_p)/2} h_i \qquad M_e = \sum_{(dE/dx_\pi+dE/dx_p)/2}^{\infty} h_i \tag{4}$$

$$\tag{5}$$

Each M has contribution from all particles. By integrating the normalized distributions, the "shapes", between the limits, we can determine these contributions, namely we can tell what portion of a particle will add to another particle. For example (see Fig. 3):

$$a_{K\pi} = \int_{(dE/dx_K+dE/dx_\pi)/2}^{(dE/dx_\pi+dE/dx_e)/2} f_\pi^{shape} \tag{6}$$

There is a quite simple connection between the measured M and the original clear N yields. The measured ones can be obtained from the original ones by multiplying with a matrix whose elements are the a contribution factors defined above:

$$\begin{pmatrix} M_\pi \\ M_K \\ M_p \\ M_e \end{pmatrix} = \begin{pmatrix} a_{\pi\pi} & a_{\pi K} & a_{\pi p} & a_{\pi e} \\ a_{K\pi} & a_{KK} & a_{Kp} & a_{Ke} \\ a_{p\pi} & a_{pK} & a_{pp} & a_{pe} \\ a_{e\pi} & a_{eK} & a_{ep} & a_{ee} \end{pmatrix} \begin{pmatrix} N_\pi \\ N_K \\ N_p \\ N_e \end{pmatrix} \tag{7}$$

thus $\mathbf{M} = A\mathbf{N}$. This equation can be solved by inverting the matrix.

$$\mathbf{N} = A^{-1}\mathbf{M}. \tag{8}$$

This way, instead of a multi-step fit we can get the wanted values by just inverting a 4×4 matrix. The β Bethe-Bloch scaler can be optimized by varying it and minimizing the χ^2/ndf value:

$$\chi^2(\beta) = \sum_i \frac{\left(h_i - \sum\limits_{\pi,K,p,e} N_x f_x(\beta)\right)^2}{h_i} \tag{9}$$

We choose those β which gives the smallest χ^2/ndf. In ideal case β should be close to 1. Especially kaons are sensitive to the right choice of this parameter.

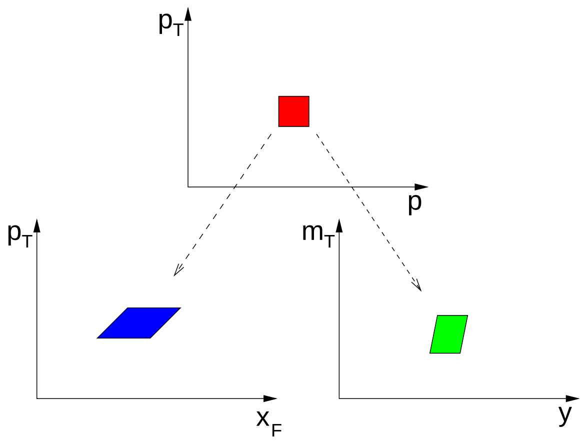
2 Physics results

The invariant cross-section (IXS) can be expressed in terms of several variables:

$$E\frac{d^3\sigma}{dp^3} = \frac{1}{m_T}\frac{d^3\sigma}{d\phi dy dm_T} = \frac{1}{p_T}\frac{d^3\sigma}{d\phi dy dp_T} = \frac{E_0}{p_T p_{projectile_0}}\frac{d^3\sigma}{d\phi dx_F dp_T} = \frac{E}{p_T}\frac{p_L}{p}\frac{d^3\sigma}{d\phi dp dp_T} \tag{10}$$

where the only problem arises when we use (x_F, p_T). There E_0 denotes the energy of the particle, $p_{projectile_0}$ is the momentum of the incoming proton, both in the center of mass system of the collision.

The natural variable for fitting energy loss is the total momentum p. Using the last expression we can have the IXS for a given (p, p_T, ϕ) place of phase space.



2.1 Rapidity

Inverse m_T slopes at midrapidity. After collecting IXS-points where $|y - y_{CM}| < 0.1$, we can plot them as a function of m_T and fit the points with the form:

$$E\frac{d^3\sigma}{dp^3}\bigg|_y = Ce^{-(m_T - m)/T} \tag{11}$$

where T is the inverse slope, C is the normalization constant.

***dN/dy* yields.** Because we do not have full coverage for all m_T – instead of summing up the contributions – we have to make extrapolation to the high m_T region. For a given y interval we select IXS-points and fit them as described above. (We assume that the exponential form of IXS stands also for non-midrapidity data.) With help of Eq. 10:

$$\frac{dN}{dy} = \int_0^{2\pi} d\phi \int_m^{\infty} dm_T m_T C e^{-(m_T - m)/T} = 2\pi CT(m+T) \tag{12}$$

2.2 Feynman-x

***dN/dx_F* yields.** Here again we do not have all p_T. For a given x_F range we collect rate-points and fit them. The function to fit is not easy to determine, but we choose it to have the form

$$\left.\frac{d^2N}{dx_F dp_T}\right|_{x_F} = C p_T^{\alpha} e^{-p_T/T} \tag{13}$$

A good choice turned out to be $\alpha = 2$. Having the result we get the yield:

$$\frac{dN}{dx_F} = \int_0^{\infty} dp_T C p_T^2 e^{-p_T/T} \tag{14}$$

Average p_T. Using the above result:

$$\langle p_T \rangle = \frac{\int_0^{\infty} dp_T C p_T^3 e^{-p_T/T}}{\int_0^{\infty} dp_T C p_T^2 e^{-p_T/T}} \tag{15}$$

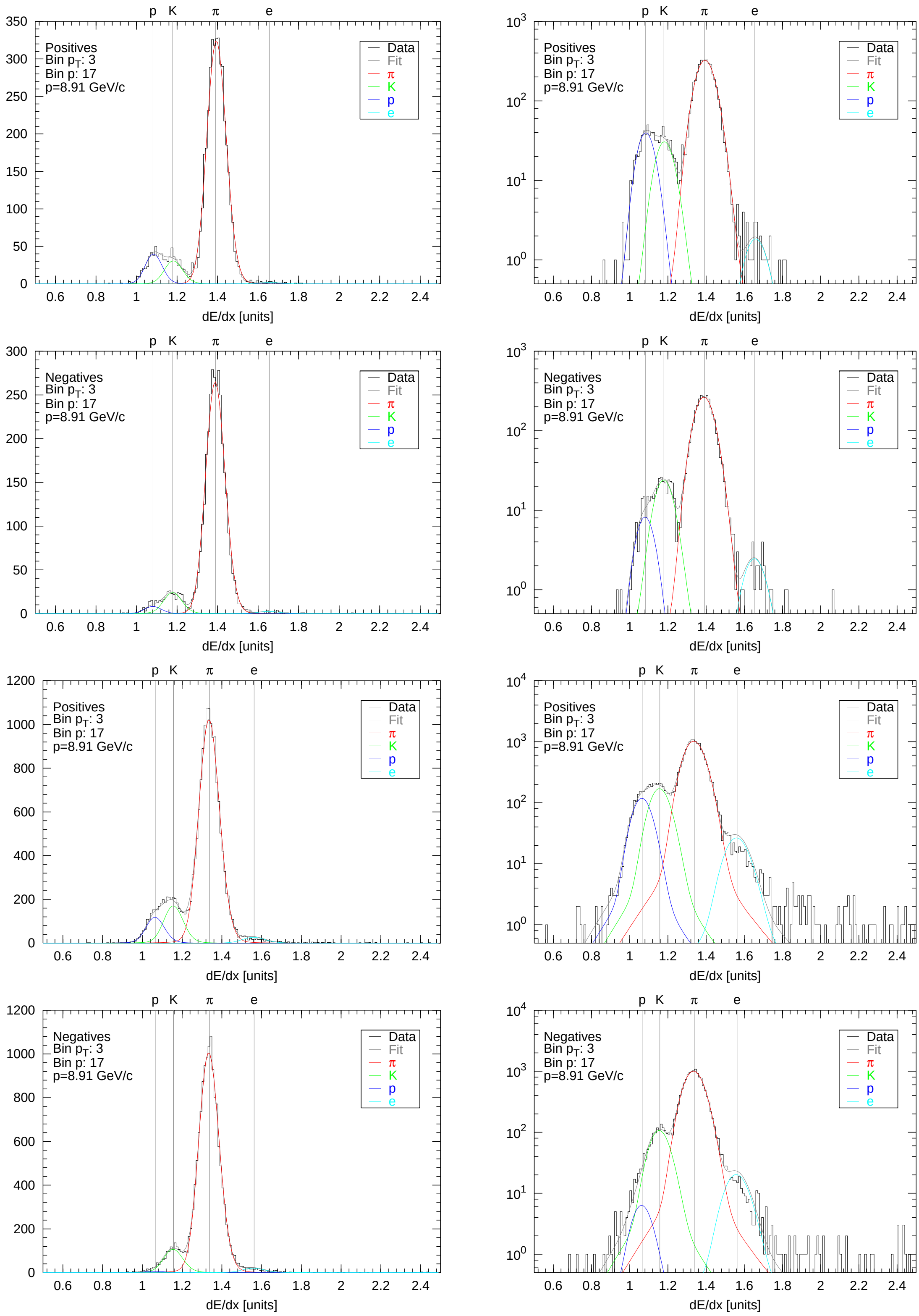


Figure 4: *Truncated mean distributions in a given* (p, p_T) *box.*