Note on the 3 FD MWPC track finder and the track weight calculation in January 01 (a^0) beam-time.

S. Dymov, s.dymov@fz-juelich.de, dymov@nusun.jinr.ru

1 Track weight calculation

The track weight is calculated using the efficiency maps supplied. There are two factors defining the weight. First of them is that there are 3 X and 3 Y wire planes used in the search, but only any 2 X and any 2 Y planes are required to build a track. Let us denote the X1W, Y1W etc planes efficiency at the track crossing points as $\varepsilon_{x1w}, \varepsilon_{y1w}$ etc. Then, probability that at least 2 X planes are NOT fired is

$$\begin{aligned} P_x &= (1-\varepsilon_{x1w})\cdot(1-\varepsilon_{x2w})\cdot\varepsilon_{x3w} + (1-\varepsilon_{x1w})\cdot(1-\varepsilon_{x3w})\cdot\varepsilon_{x2w} + \\ &\quad (1-\varepsilon_{x2w})\cdot(1-\varepsilon_{x3w})\cdot\varepsilon_{x1w} + (1-\varepsilon_{x1w})\cdot(1-\varepsilon_{x2w})\cdot(1-\varepsilon_{x3w}).\end{aligned} \tag{1}$$

Similarly, for the Y planes

$$\begin{aligned} P_y &= (1-\varepsilon_{y1w})\cdot(1-\varepsilon_{y2w})\cdot\varepsilon_{y3w} + (1-\varepsilon_{y1w})\cdot(1-\varepsilon_{y3w})\cdot\varepsilon_{y2w} + \\ &\quad (1-\varepsilon_{y2w})\cdot(1-\varepsilon_{y3w})\cdot\varepsilon_{y1w} + (1-\varepsilon_{y1w})\cdot(1-\varepsilon_{y2w})\cdot(1-\varepsilon_{y3w}).\end{aligned} \tag{2}$$

Probability that at least one of the two projection is not reconstructible is

$$P_{xy} = P_x + P_y - P_x \cdot P_y, \tag{3}$$

and the pure combinatorial track weight is

$$W_{comb} = \frac{1}{1-P_{xy}}. \tag{4}$$

The second factor, described below, introduces a small correction to the W_{comb} value. This factor arises from a cut applied during the track search. The cut is applied on a value

$$In = \log(\prod_i (1-\varepsilon_i))/N_{sp}, \tag{5}$$

where N_{sp} is the total number of wire planes used in the track search, and the product is calculated over the wire and strip planes where no hit is found for this track. In case of equal and homogeneous efficiency this cut is equivalent to a cut on the number of not fired planes. One can see, that even if hits on 2 X and 2 Y wire planes are present, the track may be rejected by the In

criterion. In practice, the cut value $In > -1$ is very low and rejects $\approx 0.5\%$ of tracks (this has been checked using experimental efficiency maps), and the final track weight W_{final} is very close to W_{comb}. Technically, W_{final} is calculated by considering all combinations of the wire and strip planes being fired or not. The sum probability P_s of the combinations where less then 2 X (or 2 Y) wire planes are fired, or the In value falls below the cut, is calculated. Then, $W_{final} = \frac{1}{1-P_s}$.

The error of the track weight is calculated as the error of W_{comb} value according to (1–4) and the known statistical errors of the efficiency values.

2 The efficiency maps used

Two kinds of the maps are used to calculate In and the weight. The first one contains the unconditional probability of wire/strip planes to be fired (the *w_eff.map, *s_eff.map files). Let us denote the efficiency of a strip plane obtained on condition of its partner wire plane being fired, by ε^w, and the efficiency obtained with the opposite condition, by $\varepsilon^{\overline{w}}$. The second kind of maps contains values ε^w (*s_rel_eff.map files). Introduction of these maps is needed since the efficiencies of the partner wire and strip planes are strongly correlated. In result, the ε values used in (5) are the unconditional efficiencies for the wire planes, and either ε^w or $\varepsilon^{\overline{w}}$ for the strip planes, depending on the hit presence in the partner wire plane. $\varepsilon^{\overline{w}}$ value, for example, for X1S plane can be calculated as

$$\varepsilon^{\overline{w}}_{x1s} = \frac{\varepsilon_{x1s} - \varepsilon_{x1w} \cdot \varepsilon^{w}_{x1s}}{1 - \varepsilon_{x1w}}.$$

There is a limitation on the efficiency values used. Namely, if any of ε, ε^w or $\varepsilon^{\overline{w}}$ values is greater then 0.99, it is taken equal to 0.99. If any of these values is less then 0.01, it is taken equal to 0.01.

In case when no efficiency maps is supplied, a dummy value (of 0.8 in our case) is used for all planes.

3 Efficiency of the search algorithm

This efficiency has been tested on the experimental data and a special simulation. The experimental data test was done using $pd \to pd$ reaction with the deuteron detected in the spectator detector. The efficiency of single track search (that is, the efficiency of the algorithm, different from the MWPC registration inefficiency) was found equal to $97.5 - 99.5\%$. These figures do not take into account the effect of cut on the track fit confidence level of 0.5%. The resulting efficiency can be stated as $98 \pm 1\%$. **There is no correction for this efficiency introduced in the track weight, it has to**

be taken into account separately. There is also a cut on the In value applied, which rejects less than 1% of events, but this loss is taken into account during calculation of W_{final} (see section 1).

Since the MWPC efficiency is used to calculate In, which is used to select the best track candidate, the track search efficiency is sensitive to systematical errors of MWPC efficiency. This dependence has been studied and found to be rather weak. That is, varying the input MWPC efficiency in the range 0.9–0.6 caused $\approx 3\%$ change in the number of the tracks found. This variation included change of the input efficiency maps as well as using different dummy efficiency values. Using the unmodified experimental efficiency maps yielded the optimal (within $\approx 1\%$ uncertainty) results.

4 Correction of experimental data

The data discussed are found in
/data/fedorets/Run3290_SW_pion_FD_d_scal.event.gz to
/data/fedorets/Run3233_SW_pion_FD_d_scal.event.gz files on ikp488. The efficiency maps applied are stored in
/home/anke/fedorets/setup/january01/run_2.6/sp_run_3276_3303. There are 371588 non-saler events are found in these files. For each of them both 2 MWPC and 3 MPWC track searches were run.

Results of 2 MWPC search. There are 228188 (61.4 % of total) events with a single track, passing all background criteria. The sum of MWPC inefficiency caused weights for these events is 544784, and the average event weight is 2.39.

Results of 3 MWPC search. In 371002 event (99.8 % of all events) a single track is found. Obviously, these was a preselection done with a condition of track presence. The weight sum of these tracks is 485895, with average weight of 1.31.

Weight estimate by efficiency maps average. The average efficiencies of the wire planes were (found in the files mentioned): $\varepsilon_{X1W} = 95\%$, $\varepsilon_{X2W} = 78\%$, $\varepsilon_{X3W} = 68\%$, $\varepsilon_{Y1W} = 72\%$, $\varepsilon_{Y2W} = 97\%$, $\varepsilon_{Y3W} = 87\%$. For the 2 MWPC track search using MWPC1 and MWPC2 this leads to a weight of 1.93. For the 3 MWPC search, according to (1–4) $W_{comb} = 1.15$.

Comparison. One can see a contradiction of results of track search with 2 and 3 MWPCs, and their strong contradiction to a rough estimation by the maps average. The last case can be explaned by difference of the events distribution over the MWPC surface in the data mentioned and the data used to get the efficiency maps. In particular, in fig. 2 one can see that the original event distribution is peaking at the top and bottom of FD acceptance. Tipically these are the zones of a lower MWPC efficiency.

The track weight sums of 2 and 3 MWPCs search results after correction by the algorithm inefficiency (see section 3) differ by 10%. It seems likely to be a results of systematical error of the MWPC efficiency determination. In figs. 1, 2 one can see that the strongest correction in 2 MWPC case occurs at the ends of the distribution. In the central part $-2^0 < \theta_{yz} < 2^0$ the difference is already only 6%.

If one considers equal and homogeneous efficiencies of the MWPCs, then to compensate the 10% difference we need to change the efficiency value of each sensitive plane by 2% for the 2 MWPC search (what seems a rather large but not unreasonable error). In case of 3 MWPC search this has to be an 7% change.

Conclusion. 3 MWPC search results should be used as final ones since the correction factor is much smaller and less dependent on MWPC efficiency systematical error in this case. The difference of 10% may be considered as an upper estimate of systematical error.

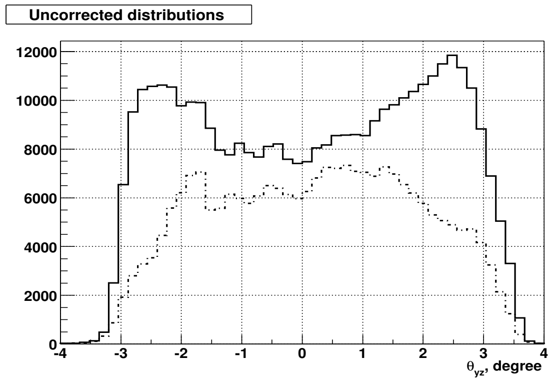

Figure 1: Uncorrected distributions of vertical angle. Full line is 3 MWPC, dashed line is 2 MWPC.

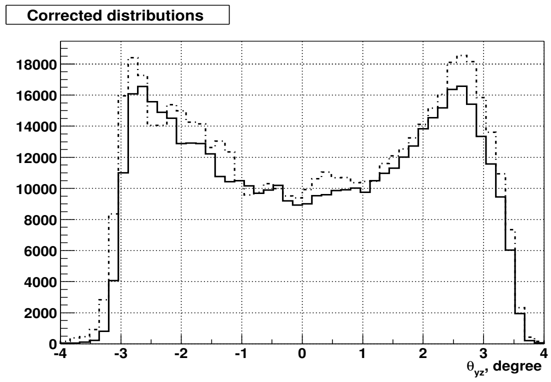

Figure 2: The same corrected distributions.

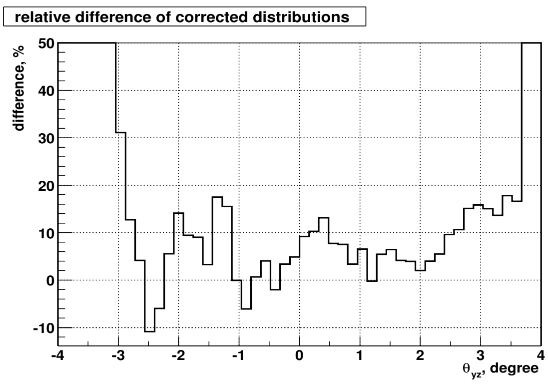

Figure 3: Ratio of corrected distributions.

5 Structure of the program

We will describe the program installed at ikp488: anke/fedorets/sorter3pc. A scheme of the part of the sorter dealing with the 3 MWPC track search is presented in fig. 4. The main code is located in the user_sergey directory.

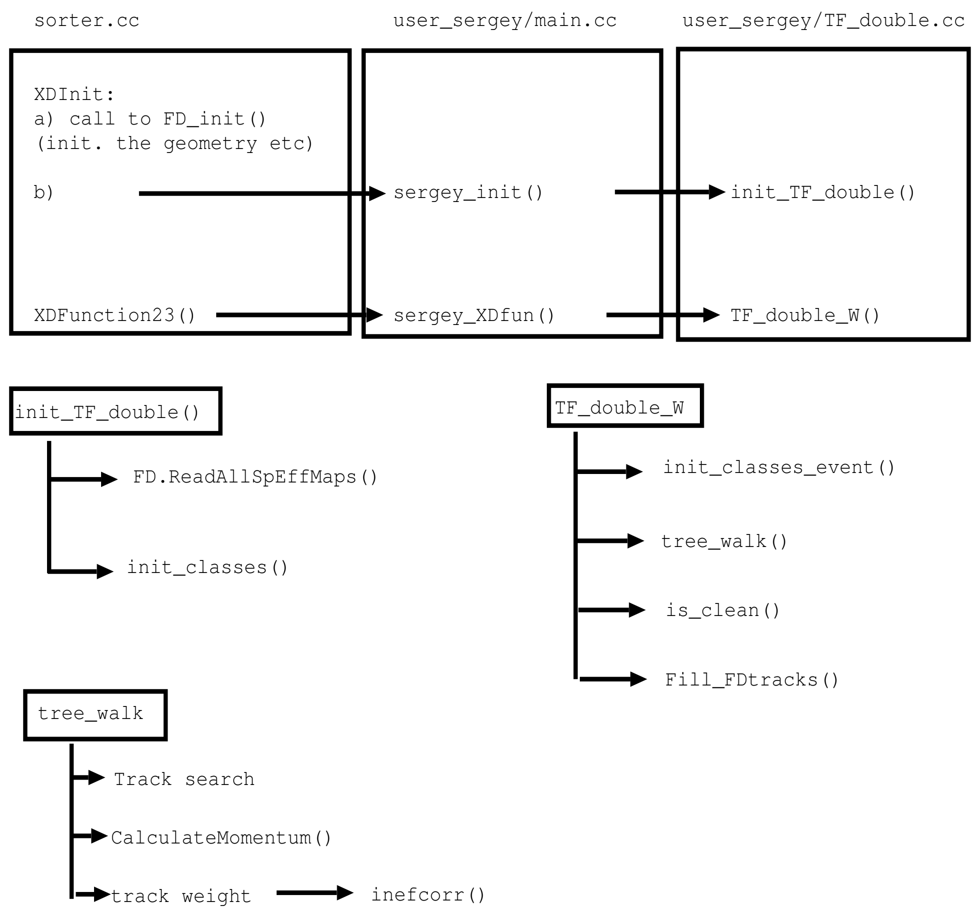

Figure 4: Scheme of the program

The main.cc file serves merely for interface with sorter.cc. Implementation of the search algorithm is kept in TF_double.cc, and the tracking class TWTrack is defined in classes.cc, classes.h files.

5.1 Efficiency maps initialization

The XDInit function contains a call to FD_init() function (forward/FD_internal.cc), which does the general initialization: reads the detector geometry, background cuts etc. In particular, this function initializes the efficiency maps. The geometry of each map is set (the Init functions) in the function FD-data_t::InitEffMaps() (forward/TF/TF_classes.cc). Then a call SetDefault-

Efficiency() is done for each map, and the default value defined in forward/sp_efficiency.h (0.8 in our case) is set. After that, ReadSpEffMaps() (forward/TF/TF_classes.cc) is called and reads the maps for the four wire planes used in the two MWPC search. The rest of the maps is read during execution of init_TF_double(), which calls the function ReadAllSpEffMaps() (forward/TF/TF_classes.cc) for that. In result, after call to init_TF_double() all the maps must be either read, or, if the reading fails, filled with the default values. In case one wants to overwrite the read/default values, this can be done in init_TF_double() after call to ReadAllSpEffMaps().

5.2 The track search routines

Actual search routines for each event are called from the TF_double_W() function. These routines include initialization for this event (init_classes_event()), call to the tree_walk() function that does the search, a loop where the background cuts are applied (by calling the is_clean function()), and filling an interfacing FWDtrack structure. The track search, momentum reconstruction and calculation of the track weight are done in the tree_walk(). It calls the inefcorr() function to calculate W_{comb}, W_{final} and the weight error as described in section 1.

The function Fill_FDtracks() serves to copy the data from TWTrack class to the FWDtrack structure, which is used also as an interface class for the two MWPC search algorithm. Since FWDtrack was originally designed for two MWPC search, the MWPC efficiency values are not filled (this structure contains fields for four wire plane only), and the track weight is copied from the TWTrack class.