TRIUMF Experiment E614: TWIST

Efficiency Report

Marc Lalancette

August 30th 2002

Introduction

In order to get a precision measurement of the Michel spectrum, we must insure that all the detector components work adequately. Ideally, each cell of every drift chamber (DC) and proportional chamber (PC) would record a signal, or hit, when a particle goes through it. However, certain factors will cause hits to be missed, such as dead time after a hit (saturation caused by too much ionization) or instrumental problems. The efficiency code is used to find out the fraction of hits that are missed, allowing us to optimize detector settings, such as high voltage and post-amp threshold on the chambers, and to monitor every cell for potential problems. This knowledge also helps in preventing biases due to a lack of uniformity in efficiency across a chamber.

Algorithm

Plane efficiency

To calculate efficiency, we first find the (u, v) coordinates of the entry and exit points of the reconstructed track through each intersected plane. We then translate these coordinates in cell numbers and look if the expected hits were actually detected in these cells. We also look in adjacent cells because the resolution of the track is limited and there is always a chance that the particle went through the next cell when the track is close to the edge. Scattering will also cause the real trajectory of the particle to diverge from a straight line, but the effect is very small for high-energy pions (120 MeV/c) that are used to get the efficiency. A namelist variable allows us to modify how far away from the expected cell we want to look for the actual hit.

Two counters for each plane are then used to calculate the efficiency. One is incremented when a plane is intersected by the track and the other when at least one hit is found in the cells we identified before. The ratio of these two counters gives the final plane efficiency value. Note that when all the expected hits are near the edge of the plane, there are less adjacent cells, so we dismiss the expected plane hit to avoid bias.

Wire efficiency

Wire efficiency is calculated in the same way as plane efficiency. However, by considering adjacent cells when a hit is not found on an expected cell, we are assuming the fitted track was wrong[1] and the particle actually went through one or more adjacent cells instead of the expected cell. In such cases, we disregard the expected cell and don't increment any counters. We will discuss the effects such an assumption can have on efficiency in the systematic errors section.

Adjacent fractions

The efficiency code also uses counters to get the fraction of times an expected hit is not found in the expected cell but on an adjacent cell. Whether the expected hit is on a greater or lower cell number is also noted. The fraction of expected hits on a greater cell should be 0.5, and it is sensitive to tracking biases and alignment of the detector planes. This fraction was originally measured to verify that the cell numbers we get from (u, v) coordinates were correct. (An error in code at that time shifted everything by half a cell.)

Windowing

It is worth mentioning that the efficiency code can either look for the actual hits among all the recorded hits in an event or only among those within a given time interval (window) around the hits used to reconstruct the track. This option has been used in the past because of errors in the windowing code that caused PC hits to be missing from the track window, reducing the efficiency. These errors have been fixed. We usually want to restrict the hits in time so we don't find hits from other particles, which would wrongly increase the efficiency.

Mofia usage

Efficiency functions

To find plane and wire efficiency in Mofia, one must first initialize counters with the command func 14 (they are not initialized by default). After analyzing the desired events with correct namelist settings, one can print the results to files in the user directory and/or record them in histograms by issuing the command func 12.

Efficiency namelist variables

FindEff (F) To turn on efficiency analysis, set this variable to true.

The next two variables are used to control whether func 12 saves efficiency results to file and whether it fills histograms of efficiency per plane and wire and statistics on adjacent hit cells.
EffPrint (T), EffFillHist (T)

---

[1] Here of course, we mean that the finite resolution of the track caused it to be slightly off in this particular plane, not that the entire track should be rejected.

EffUseWindowing (T) To look for both DC and PC hits only in the track window, this variable must be true. Otherwise, all the hits in the event are considered and some found hits might be from a different particle.
EffDebug (F) When true, enables efficiency debugging: missing hits, track intersections and other hit information would be printed on screen during analysis.

The next two variables are the maximum number of adjacent *wires* to look on each side of the expected hit cell to try and find a hit.[2] Note that the code was adjusted to correct for ganged wires, so the distance from the expected cell is equivalent in all cases.
EffDCCellCut (2), EffPCCellCut (4)

The four next variables are cuts on the plane radius. If all hits are outside the radius on a plane, that plane is skipped for that event. The default values on these cuts are calculated so that all the expected hit cells considered will have enough adjacent cells on both side. That way, there is no bias from hits that are expected on the edge of a plane and that were actually just outside the plane. The formula is: (wire center of last cell) + (half cell) - CellCut * (cell width). Of course, while these cuts give us unbiased plane efficiency, they also prevent us to get meaningful wire efficiency for the edge wires. To get this
DCRadiusCutDense (8.8 cm) 9.4 + 0.2 - (2*0.4)
DCRadiusCutSparse (15.2 cm) 15.8 + 0.2 - (2*0.4)
PCRadiusCutDense (4.0 cm) 4.7 + 0.1 - (4*0.2)
PCRadiusCutSparse (15.2 cm) 15.9 + 0.1 - (4*0.2)

nHitPlanesCut (0) Not currently used in code, but would be a cut on the minimum number of hit planes for an event to be analyzed by the efficiency code. We can use dccuts/min_dc_planes and pccuts:min_pc_planes instead. Of course if these cuts were too strict, we would bias plane efficiency, but we currently require only 6 out of 44 DCs and 6 out of 12 PCs so the effect is too small to measure.

Sensitivity to other settings

Efficiency results are very sensitive to the accuracy of the fitted track. Therefore, to get the intrinsic efficiency of the chambers, it is best to use strict cuts on the track to only consider well reconstructed tracks. Without going into the details, cuts on χ^2 and adjusting the number of tracking iterations will improve the accepted tracks. Before computing efficiency, track quality tests should be performed to optimize parameters (with the latest code, efficiency changed by over 1% in such tests).

Another factor to consider is noise and crosstalk hits, which cause biases in wire efficiency (see below). To minimize these effects, we can remove crosstalk hits and limit the number of hits per plane with dccuts:dc_max_hits_in_plane and pccuts:pc_max_hits_in_plane, so noisy events are rejected.

---

[2] The default values are believed to have been too high, thus hiding some code or chamber problems in PCs. See table 1 and figures 3 and 4 in results section. The recommended values are now 1 for DCs and 2 for PCs.

Other settings that influence tracking or efficiency directly are time zero triggertime (which is important when using drift time in the fit), windowing winPCThreshold (which influences how the hits are separated in windows) and cuts on minimum TDC time (which impacts on whether or not window 2 is the trigger window).

Results

Efficiency per plane and wire

Figure 1 shows four plots of efficiency for run 5680. This is a run of over 1.5 million events of 120 MeV/c pions with a low focus beam to intersect more cells. High voltages are 1900 V for DCs, 2050 V for outer[3] PCs and 2000 V for target PCs, while thresholds are –125 mV for DCs, -150 mV for target PCs, -250 mV for central wires of outer PCs and –300 mV for ganged wires of outer PCs[4].

The plots on the left show the efficiency per plane for DCs (top) and PCs (bottom). All planes have a very high efficiency (most above 99.96%) and DCs don't show large variations except DCs 21 and 15 that are a bit lower. PCs on the other hand show variations between upstream, target and downstream. Higher efficiency for target PCs is due to a lower threshold. There could also be a contribution from hard scattering before the first DC or after the last DC. Note that if hard scattering occurs between the first and last DC, cuts on the χ^2 would reject the track, so we wouldn't expect it to affect DCs or target PCs.

The lower efficiency downstream is not fully understood, but has sometime been observed to disappear when tracking parameters were adjusted differently (timing on or off for example) so it is possible that it is not a real effect. The main difference between upstream and downstream PCs is that the beam is less focused when it reaches the end of the detector. Perhaps this drop is an indication of slightly lower efficiency near the end of the wires. In any case, these variations are very small and the efficiency remains more than adequate.

The plots on the right are efficiency per wire for DCs (top) and PCs (bottom). Figure 2 shows PC 4 in detail. The observed shape is consistent with the fact that outer wires have lower statistics since most particles go through the center of the detector. Wires that indicate 100% efficiency most probably have an efficiency between N/N+1 and 1, where N is the number of hits expected (and found) in these wires. Some outer wires show a lower efficiency because they missed one or more hit early on and there weren't sufficient expected hits afterwards to compensate. The observed value of the central wires (around 99.95%) indicate that we'd need at least a few thousand expected hits on every wire to get a good estimate of wire efficiency, compared to about 50 for the outer wires in figures 1 and 2. We will therefore need to get higher statistics for outer wires if we want to study possible efficiency variations across a plane that could be caused by the field shape close to the edge of the chambers for example.

---

[3] PCs are divided into three groups of four chambers. Upstream: 1-4, target: 5-8 and downstream: 9-12. Outer PCs refers to non-target PCs (upstream and downstream).

[4] These high voltage and threshold values are the latest recommendations from efficiency and crosstalk studies. They are conservative, aiming at higher efficiency first and then lowering crosstalk. Details are discussed in the crosstalk report.

It has been observed that outer PC efficiency drops when the PC cell cut is lowered (see table 1). The worst plane affected was PC 9. It dropped by 7.7% when changing the cut from 4 to 1. This indicated a mapping error that has now been fixed. The alternating up and down wire pattern in outer PCs (figures 3 and 4) would be consistent with wires switched in pairs combined with bad plane alignment, but tests indicate that cell mapping is not the cause. Although the alternating pattern is not understood, the efficiency drop is probably linked to tracking resolution and hard scattering. We must also remember that PC cells are 2 mm wide compared to 4 mm for DC cells.

	Efficiency		
PC	Cell cut = 4	Cell cut = 2	Cell cut = 1
1	0.99955	0.99812	0.96838
5	0.99967	0.99956	0.99776
9	0.99916	0.98973	0.92204
12	0.99886	0.99736	0.97681
DCs		0.99972	0.99960

Table 1: PC efficiency for various cell cuts.

Efficiency vs. high voltage

Efficiency of each DC and PC was measured as a function of high voltage. Figures 3-13 show the results for individual DCs from 1600 V to 2100 V. This data corresponds to runs 4680-4690 for half the planes (3, 4, 7, 8, ...) and runs 4732-4742 for the others (1, 2, 5, 6, ...), which were taken during June 2002. Note that DC 2 was not working during these runs, but subsequent runs indicated that it had a similar curve as DC 1. Later runs did not show significant differences from the data shown here.

Each plot shows the curve of an individual plane, on two different scales (right and left axes) to view the detail of the plateau region. Most planes reach over 99.9% efficiency at 1850 V (see plane 1 for example) but planes 11 and 12 only reach this mark at 1900 V and planes 3 to 7 and 40 to 42 reach it before, at 1800 V. Since we want the operating point to be safely above the edge of the plateau, it has been set to 1900 V.

Figures 14-19 show plots of efficiency vs. high voltage for each PC plane from runs 4692-4714 and runs 5265-5298. These runs were analyzed without window restrictions, as there were code problems at the time. By analyzing without windowing, the curves regained the expected shape with a plateau close to 100%.

PCs are divided in three groups of four: upstream (1-4), target (5-8) and downstream (9-12). In the first set of runs, with all thresholds at -125 mV, all planes reach the 99.9% mark at 1950 V except PC 7, which has a higher gain and reaches it at 1900 V. PC 8 also seems to have a good efficiency at 1900 V. For the more recent runs, with target PC thresholds at -125 mV and upstream and downstream PC thresholds at -175 mV, there are a few differences.

First, the plateau seems a bit lower throughout, more markedly downstream. This time, PCs 5, 7 and 8 reach the 99.9% mark at 1900 V, PC 6 not far behind. As for non-target PCs, the edge of the plateau seems to have shifted a bit towards higher voltage, but by less than 50 V.

It seems that 2000 V should be a high enough operating point, and we could probably manage well at 1950 V although it is on the edge for downstream PCs. Note that PC 10 seems to have the lowest gain and is the only plane below 99.8% at 1950 V in the recent runs.

Efficiency vs. gas

The same plots of efficiency vs. HV for PCs are found in figures 20-22. They are for runs 5327-5335, where PCs are filled with argon/isobutane gas instead of CF_4/isobutane gas. This modification was an attempt at gaining energy resolution in the target PC ADCs (see the report on energy loss analysis in the target PCs). These curves indicate that the chambers have a higher gain with this gas, as the efficiency plateau is reached at a lower high voltage. The operating point in this case would be 1950 V for the outer PCs and 1900 V for the target PCs.

Efficiency vs. threshold

Preliminary analysis of efficiency vs. threshold indicated that the latter could be raised somewhat without loosing much efficiency. This is beneficial to lowering noise levels and crosstalk, especially in ganged wires. As shown in figure 23, in this first analysis (runs 4715-4719) with PC high voltage at 2100 V, PC efficiency stayed above 99.9% up to -400 mV. Later runs (5623-5629), plotted per plane in figures 24-25, with PC HV at 2150 V, indicate that the efficiency stays above 99.9% up to -300 mV for upstream PCs and up to -250 mV for downstream PCs. Target PC thresholds were not modified in these runs.

Hits on adjacent cells

The fraction of hits that are not found on the expected cells but on adjacent cells is plotted for DCs on the left plot in figure 26. This pattern has also been seen in plane residuals while doing alignment analysis. It seems to indicate a bias in the straight track fitting code. This is currently being investigated. For PCs (right plot in figure 26), we notice that outer PCs have a larger fraction. This is expected since the track is not fitted in these regions. Also, PC 9 consistently displays a larger fraction than other PCs, sometimes by a factor of 2. This could be an indication of a mapping error in a few wires.

Figure 27 shows the fraction of adjacent hits that were expected on a greater cell number than where they were found. This fraction is significantly different from 0.5 and displays the same relative pattern from run to run. Statistic error bars are plotted, but too small to see. Such consistencies are difficult to explain if not a direct result of plane misalignments.

Statistical and systematic errors

Statistical errors

Statistically, the number of hits detected (n) out of N times a particle goes through a cell follows a binomial distribution with mean $\mu = Np$ and variance $\sigma^2 = Np(1-p)$, where p is the real efficiency (the probability of a particle going through being detected). We estimate p with $\hat{p} = n/N$. It is a good estimator, with p as its mean and $s^2 = p(1-p)/N$ as variance. A good estimator for s^2 is $\hat{S}^2 = \hat{p}(1-\hat{p})/(N-1)$, the square root of which gives us the statistical error on our measured efficiency $\hat{p}$. However, this error seems very small and according to *The Data Analysis BriefBook* (http://rkb.home.cern.ch/rkb/titleA.html) "the probability of obtaining n events out of N for a given p should not be estimated by comparing the difference of $\hat{p}$ and p against $\hat{S}^2$, but from a Poisson distribution with mean pN, particularly if $\hat{p}$ is close to 0 or 1." This certainly applies to our case, and this should probably be investigated further. In any case, there are other sources of error on these measurements apart from statistical and we currently use $\hat{p}(1-\hat{p})$ as the error on efficiency.

The error on the fraction of hits expected on a greater cell than the actual hit is more straightforward. It is also a binomial distribution, but we know the probability is 0.5. The statistical error is then $\frac{1}{2}\sqrt{N}$.

Systematic errors in wire efficiency

While we have not found any systematic error in the plane efficiency algorithm, wire efficiency is biased in more than one way by the finite track resolution and by how we treat cases where a hit is not found in the expected cell. All cases that introduce an error are listed below, however, all these errors can be minimized or prevented and are believed small enough to not affect our efficiency results. To be rigorous and increase confidence in our quoted efficiency, they should still eventually be quantified.

Note that other algorithm options have been considered, such as incrementing the found counter of the adjacent cell when it has a hit, but while they can reduce or remove some of these biases, they increase others or introduce new ones. We believe the current algorithm is the best tradeoff.

Case 1. If the track is right and there is more than one expected cell, but one hit is not found, we do not decrease the efficiency of the cell with the missing hit. As stated earlier, in such cases, we assume that the track was wrong. This will give a higher efficiency to that wire.[5]

---

[5] Although this has not yet been done, we can find the importance of this effect with a few additional counters (D and $\overline{\mathrm{D}}$) in the code and some statistical calculations. If we know the fraction of tracks that expect two cells hit but that are wrong (d), the count of two expected cells but only one hit present (D), the count of expected hits (e), and the calculated efficiency (W), a few iterations of the formula

E = (1 - 2(Dd/(e+Dd))E(1-E))W (starting with E=W)

Case 2. Same as case 1 except that an expected hit was present and removed by the crosstalk routine. We assume the track is wrong and do not increase (as we should) the efficiency. This bias will lower efficiency, but it can be minimized by optimizing crosstalk parameters (see report on crosstalk) and is absent when crosstalk hits are not removed, as is the case usually when computing efficiency.

Case 3a. If the track is right and there is no hit on the expected cell but there is a noise or crosstalk hit on an adjacent cell, we will not decrease the efficiency of that wire because we assume the track was wrong. This bias will increase the efficiency.

Case 3b. If the track is wrong and there is a noise or crosstalk hit on the expected cell, we will increase the efficiency of that wire while we shouldn't. However, noise levels are very low so the probability of having a false hit in the adjacent cell when a hit is missed is very small. As for crosstalk, it would require a close second wide hit in the same time window. This is also improbable these biases should be minimal.

Case 4a. When the track is wrong (but expects the right number of cells) and there is no hit on the expected cells or around them, we decrease the efficiency of the wrong cells. This case will "mix" the efficiencies of the expected and correct cells.

Case 4b. If the track expects more cells than really hit, we will globally decrease the efficiency. If it expects fewer cells than really hit, we'll increase efficiency. If both cases occur equally often, this will increase efficiency error, but will not introduce a bias.

To get more information on the importance of cases where more than one hit is expected, one could look at the difference in efficiency when only one hit is expected per plane. This could be done by using tracks with very small angles. The importance of cases where the track is wrong depends on resolution and scattering. Technical note 59 reported differences lower than 0.1% in plane efficiency for different tracking resolutions. The same information could be obtained for wire efficiency. Finally, note that while crosstalk removal influences wire efficiency, it does not influence plane efficiency since there must always be a wide hit present in the plane for the code to find crosstalk hits.

---

should give a good estimate of the real efficiency (E). The only parameter that is not readily available is d, but it can be estimated by the ratio of D over the count of two expected cells and two hits present ($\overline{D}$).

Conclusion

The efficiency of all chambers has been measured at different high voltage and threshold settings. After comparison with crosstalk, conservative operating value recommendations were given (see start of results section or crosstalk report) but some more data would allow us to optimize them further.

Most chambers have an efficiency over 99.9%. With 44 DCs used for tracking, the probability that enough hits would be missed to bias a track is incredibly low. Any effect on the final results would be much smaller than the accuracy we're aiming for.

Observations that are not yet understood include alternating wire efficiency in outer PCs when the cell cut is 1 and lower efficiency in downstream PCs.

Tracking biases are seen in the fraction of expected hits that are found in adjacent DC cells. The fraction of these on a lower cell indicate that plane alignment is not well calibrated.

Systematic errors were investigated. None were found for plane efficiency, and all errors in wire efficiency are negligible effects, although they have not yet been quantified precisely.

However, we don't have enough statistics to rule out biases that could arise from efficiency variation at the edge of the chambers. We'll need to get more data with a wider beam focus if possible and also modify the code to calculate efficiency separately according to which wires are hit (especially ganged and central wires) or where the hit is expected along the wire. As previously noted however, preliminary results indicate that any variation would be small and wouldn't affect our results.

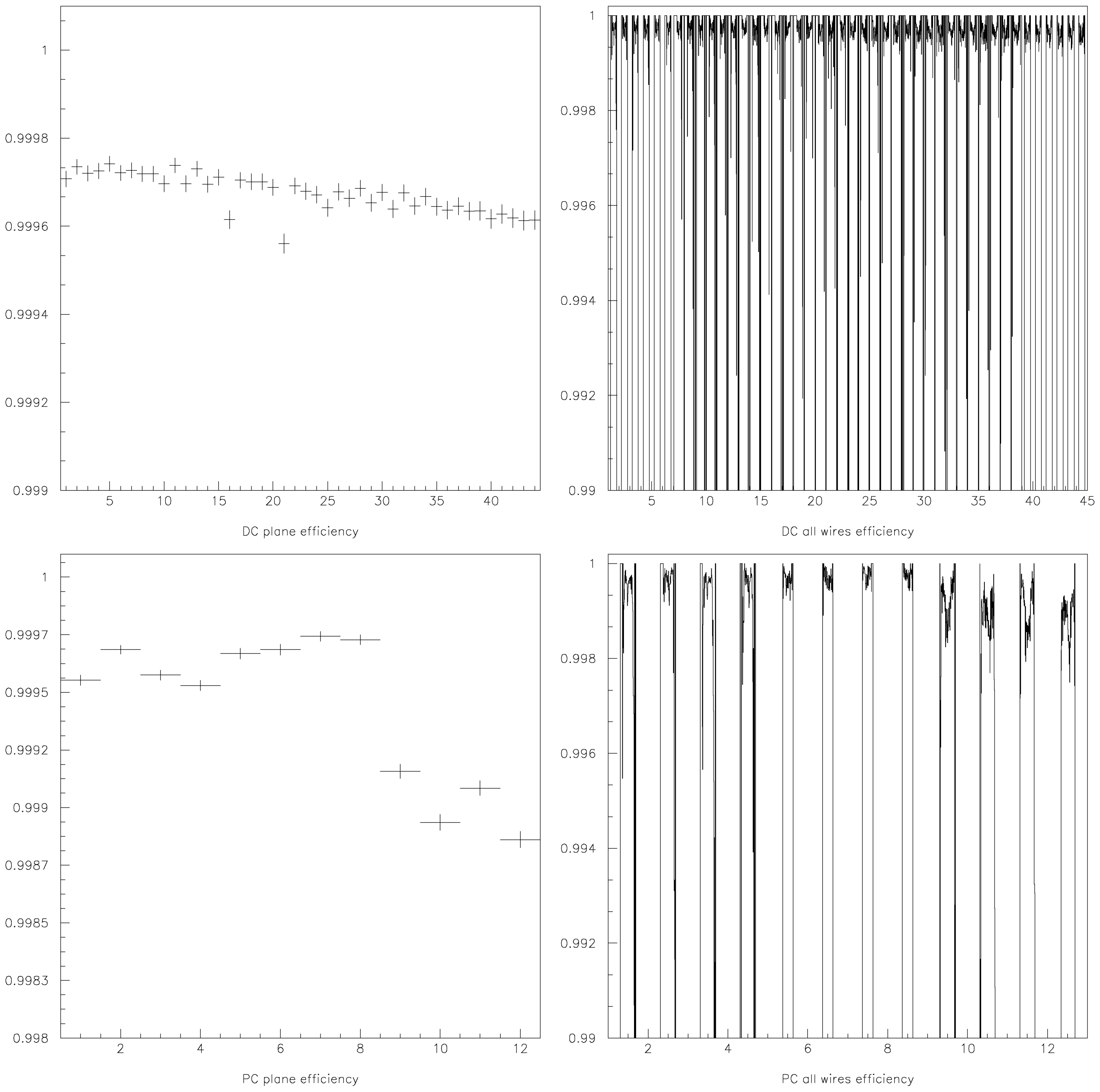

Figure 1: Efficiency per plane (left) and per wire (right) for DCs (top) and PCs (bottom). (HV and threshold settings in text.)

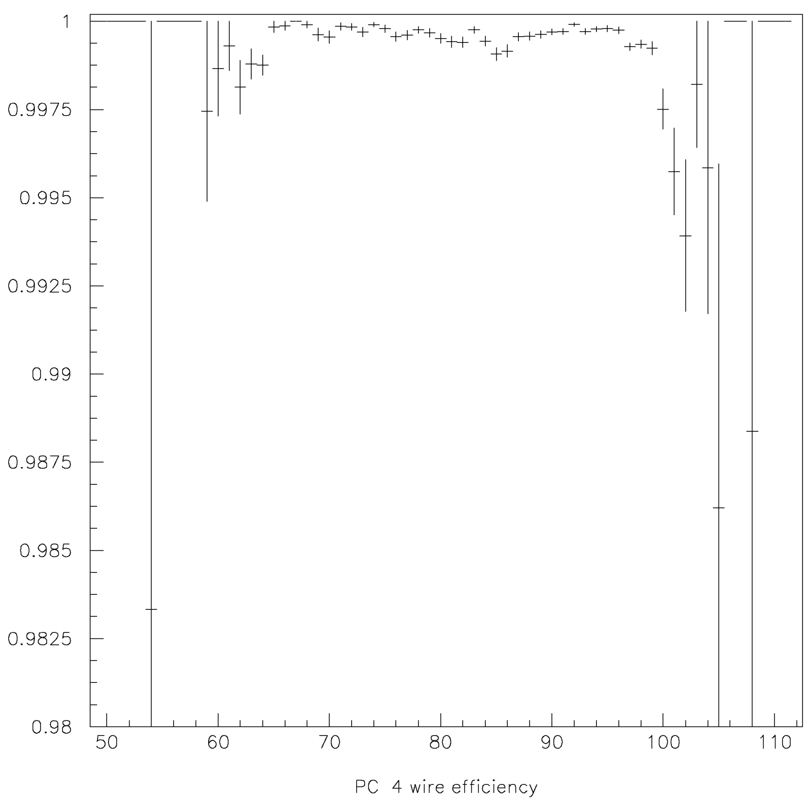

Figure 2: Efficiency per wire for PC 4, HV: 2050 V, threshold –300 mV.

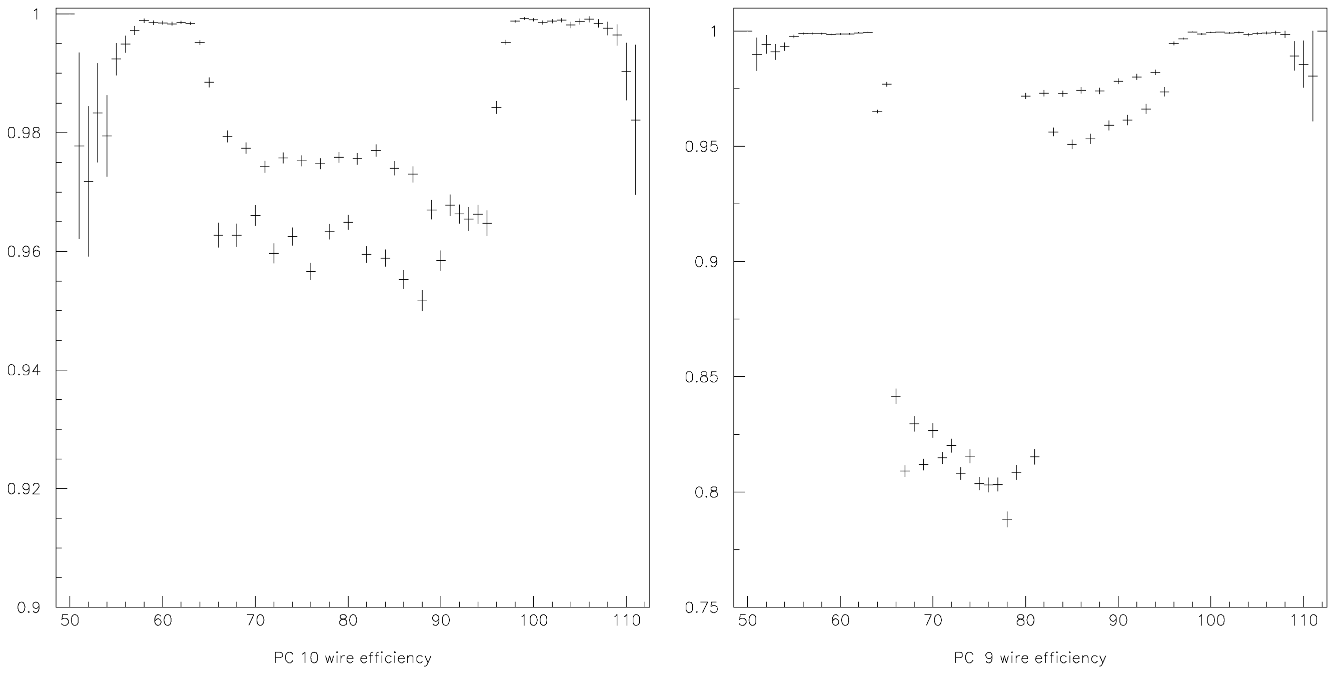

Figure 3: PC 10 (left) and PC 9 (right) wire efficiency when looking only one wire away when a hit is not found on the expected cell.

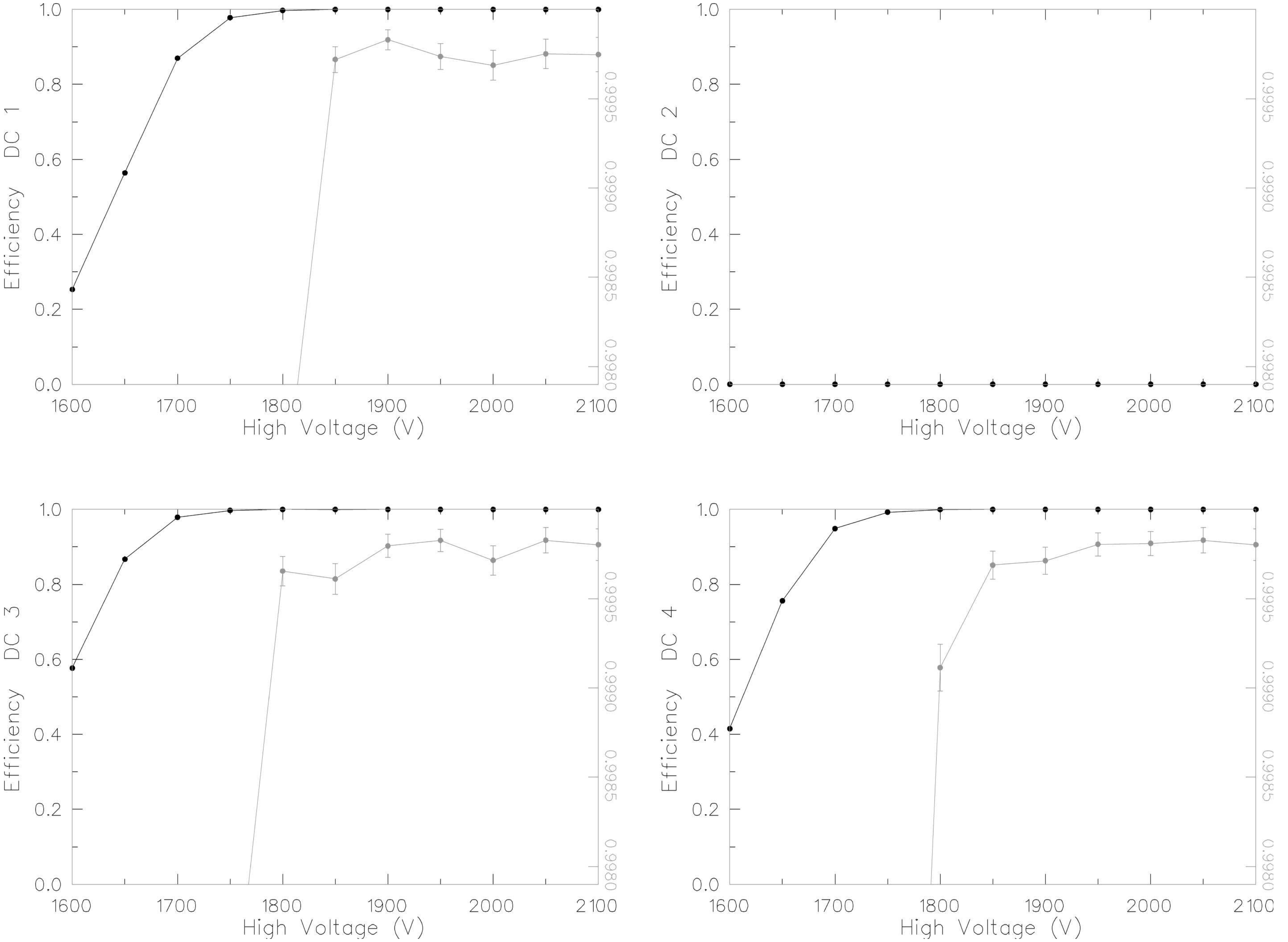

Figure 3: DC efficiency as a function of high voltage. Threshold: -125 mV

Efficiency DC 7

Efficiency DC 5

```
ERROR: ioerror
OFFENDING COMMAND: image

STACK:
```