STAR Forward Time Projection Chamber (FTPC)

Software and Calibration Overview

March 15, 2010

Version 2.2

Terence J. Tarnowsky

Brijesh K. Srivastava

for the

STAR FTPC Group

Brookhaven National Laboratory

Max-Planck-Insitut für Physik (Werner-Heisenberg-Insitut)

Purdue University

TABLE OF CONTENTS

LIST OF FIGURES

Version Changes

Changes from Version 2.1:

- In the **Gain Tables** section

  Added reference to the FTPC NoiseFinder web documentation

- In the **Laser Calibration** section

  Removed the paragraph explaining why at least one of the two quantities Δt_0 or Δg should be non-zero.The laser analysis macros have been corrected so that this is no longer necessary.

- In the **Analyzing Laser Runs** section

  Change "dt.000001" to "dt0" in output filenames since it is no longer a requirement that least one of the two quantities Δt_0 or Δg must be non-zero.

Janet Seyboth, March 15, 2010

Changes from Version 2.0:

- Added Appendices

Janet Seyboth, March 15, 2010

Changes from Version 1.1:

- Added new sections:

  Database Initialization before a New RHIC Run Period

  Database Updates Once Run Starts

  FTPC Voltage Status

- In the **Gain Scan** section

  The special set of Ftpc point histograms used to evaluate Ftpc gain scan runs are created and filled when bfc [1] option 'fgain' is included in the reconstruction chain

  The anode volage settings for 2006 have been added to the list of voltage settings.

- Changes and additions in the **Slow Control** section:

  The slow control archives can only be accessed behind the starp firewall

- Changes to the **Laser Calibration** section:

  Laser runs are reconstructed with a 'perfect' gain table and with adjustAverageWest = adjustAverageEast = 0.0

  The inner cathode correction is applied

  Changed **lasertest_single.C** and **lasertest.C** macro arguments

- In the **Gain Tables** section:

  With the advent of DAQ1000, the **NoiseFinder** programs can be run on the rcas machines.

- Removed all information which was marked as obsolete in v1.1

- Text in red or marked with a red ?? is still being worked on

Janet Seyboth, 2009-08-07

Changes from Version 1.0:

- Added Table of Contents and List of Figures.

- Updated “Laser Calibration” to reflect implementation of StFtpcCalibMaker. Information about Lase rMaker now in “Obsolete”.

- Added some additional information about macros used for “Rotation Calibration”.

1 FTPC Contacts

FTPC Hardware/Ops: Alexei Lebedev, BNL.

Contact information: (631)-344-3101 (Office)

FTPC Homepage at BNL: `http://drupal.star.bnl.gov/STAR/subsys/ftpc`

Access to FTPC Computers at BNL:

Janet Seyboth, MPI

Terry Tarnowsky,Michigan State University

Ask about username/password for STAR protected area or offline database browser.

2 Before Run/Start of Run

2.1 Database Initialization before a New RHIC Run Period

The STAR offline database is a MySQL database [2]. It contains all the time-dependent parameters needed for reconstruction. The top-level structure is the **domain**. The two revelent domains for the FTPC are Calibrations_ftpc and Geometry_ftpc. Each domain contains timestamped[3] **table** entries. The MySQL timestamps are in GMT. Each table has a header structure and a data structure which contains data fields. The data structure is defined by an **idl** file. The idl files are in the STAR program library branch **StDb/idl**.

Before a new run period starts, new ftpcAmpSlope, ftpcCoordTrans and ftpcGas tables must be entered into the Calibrations_ftpc database with the "initial" timestamp for the upcoming run year which Jerome announces before a new run year starts.

The ftpcAmpSlope table must contain the 'perfect' gain table. In 'perfect' gain table all entries = 1.0.

In the ftpcCoordTrans table the observedVertexOffsetX and observedVertexOffsetY data fields must be set to 0.

In the ftpcGas table the following data fields set:

temperatureDifference = adjustAverageWest = adjustAverageEast = 0.0

defaultTemperatureWest = defaultTemperatureEast = 25.0

percentAr = percentCO2 = 50.0

The instructions for initializing these database tables are in Appendix A.

2.2 Gain Scan

The purpose of a gain scan is to verify the best operating voltage for the FTPC anodes. This should be done at the start of yearly data taking. The FTPC QA histograms for cluster charge, max ADC, track residuals, and multiplicity are the primary sources of information.

In the charge and max ADC plots, if the gain voltage is too high these distributions will show deviations from their Landau-like behavior.

The reconstructed multiplicity versus gain voltage should demonstrate a rapid rise followed by a plateau above some saturation voltage. This plateau has NOT been seen, so the primary factor in deciding the gain voltage is one at which the electronic integrity of the FTPCs are not compromised. In other words, use a safe operating voltage. A gain scan from 1750-1850 V should cover an appropriate range.

Reconstruct ∼500 events for each gain scan voltage setting using the standard reconstruction chain [4] plus the "fgain" option. When the "fgain" option is used, the special set of Ftpc hit and track histograms used to evaluate the Ftpc gain scan runs are created and filled. These histograms will be included in the *.hist.root output file and can be converted to a *.ps file with the 'bfcread_hist_to_ps.C' [7] macro.

Modification to the anode voltages may be necessitated by changes in RHIC energy or luminosity.

Anode voltages for previous runs:

- 2001 Au+Au, 200 GeV: 1825 V (after a grounding problem was fixed, the voltage was lowered in subsequent years due to improved gain at a lower voltage.)
- 2003

  d+Au, 200 GeV:

  Days 1-50, 1800 V.

  Days 51-87, 1760 W, 1750 E.

  pp, 200 GeV:

  Day 88-End, 1800 V.

- 2004

  Au+Au, 200 GeV:

  1775 V $\rightarrow$ 1800 V.??

  Starting 2/20/04, 1760 W, 1750 E.

  Au+Au, 62.4 GeV:

  Day 70-92, 1760 W, 1750 E.

  Day 92-End, 1800 V.

  pp, 200 GeV:

  1800 V.

- 2005

  Cu+Cu, 200 GeV:

    Starting 1/12/05, 1800 V → 1775 V on 1/27/05.

  Cu+Cu, 62.4 GeV:

    1775 V.

  Cu+Cu, 22 GeV:

    1775 V.

  pp, 200 & 400 GeV:

    1775 V.

- 2006

  pp, 200 GeV:

    Starting 1/03/06, 1775 W, 1775 E

  pp, 62 GeV:

    Starting 6/06/06, 1775 W, 1775 E

- 2007 ??

- 2008 ??

- 2009 ??

2.3 Database Updates Once Run Starts

For physics runs, the 'uSecondsPerTimebin' value (used in StFtpcClusterMaker for finding clusters) is normally calculated from the RHIC clock. If no RHIC clock value is available for a run, the default value is taken from the Calibrations_ftpc/ftpcElectronics database table.

Whenever the RHIC clock changes for physics running (this can happen when either the beam energy or the beam species changes), recalculate the 'uSecondsPer-Timebin' value and enter the new value into Calibrations_ftpc/ftpcElectronics [6].

3 During/After Run

3.1 Slow Control Archive

The Slow Control Archives are behind the starp firewall. They can ONLY be accessed from a computer in the starp domain.

FTPC operational information is stored in the slow control archive. This is accessible from `http://www.star.bnl.gov` → Experiment → Subsystems → FTPC → Slow Control → FTPC Slow Control Monitor . This information can be viewed graphically or as a spreadsheet. Text information is also stored in the offline database, accessible from `http://www.star.bnl.gov` → Computing → DB Browsing Interface → STAR DB Browser → Calibrations → Calibrations_ftpc.

Especially important quantities are FTPC extra temperatures and the anode or cathode voltages. Various FTPC problems can often be traced to malfunctions that manifest as discontinuities in these quantities. The best way to check this is to examine them graphically from the Slow Control archive.

At the conclusion of physics running (but before data reconstruction begins), the entire run should be examined for discrepancies that may have been missed. FTPC temperatures are a priority because incorrect temperature readings cause clusters to be reconstructed in the wrong positions, making the data useless. Temperature problems in the offline QA manifest as a shift in the radial step and/or an irregularity in the chargestep.

Starting in Y2006, the FTPC average gas temperatures are calculated from the extra temperatures only; the body temperatures are no longer used. There are 7 usable readings for FTPC East and 6 for FTPC West.

Other values to check include anode and cathode voltages for the entire run.

The information for the gas system is stored in the offline database in **Calibrations_ftpc/ftpcGas**. Whenever the value for any of the ftpcGas table data fields changes, a new time-stamped table entry must be added to the database [6].

3.2 Laser Calibration

The FTPC laser system is integral for checking the gas composition and $\vec{E}\mathbf{x}\vec{B}$ corrections. A change in laser t_0 may also be detected (laser and data t_0's are different). The gas composition for the FTPC is kept at a 50%-50% mixture of Ar and CO_2. There are a maximum of 15 laser tracks per FTPC, 5 each in three laser sectors. The laser sectors correspond to FTPC hardware sectors 2, 4, and 6. In each sector are 3 straight tracks that run parallel to the beam pipe, at a specific radius, and 2 inclined (diagonal) tracks. The measured radial positions of the 3 straight tracks are 11.91

cm, 19.55 cm, and 28.56 cm from the beam line. Related diagrams can be found at `http://www.star.bnl.gov` → Experiment → Subsystems → FTPC → Calibrations → Laser System.

If the gas composition has changed dramatically and permanently, see the section regarding the "DriftMapMaker" for further information.

Laser runs are taken every few days during the run, or as needed. Once a run is taken, several hundred events (∼200 minimum) from the *.daq file should be processed through bfc.C using the chain options in [8]. A file called **debug.ini** [9] must be present for laser analysis. A sample **debug.ini** file is located in $STAR/StRoot/StFtpcCalibMaker/examples. Additionally, other parameters than those necessary for data files are used. For example, a 'perfect' gain table is used, no gas temperature adjustments are applied and LaserTracking is used. These settings are automatically used when the **flaser** option is included in the bfc [1].

This will produce several root files, including a "*laser_test.root" file, unless the output filename is changed in "debug.ini". The "*laser_test.root" file is used in the next step as the input for StFtpcCalibMaker.

The macros in **StFtpcCalibMaker** analyze the laser file produced by the chain used in the bfc.C macro with the laser ('flaser') and debug ('fdbg') options enabled. These macros can be run by adding **$(STAR)/StRoot/StFtpcCalibMaker/macros** to the **Root.MacroPath:** in your **.rootrc** file.

StFtpcCalibMaker plots the results from the 'bfc.C' chain, but includes corrections for temperature, pressure, clock frequency, and magnetic field.

The two macros 'lasertest_single.C' and 'lasertest.C' analyze the "*laser_test.root" file with the appropriate corrections and output several other files. The command line parameters for these macros are:

- FTPC (1, FTPC W; 2, FTPC E),
- Laser sector (1, 2, or 3),
- Tracks, (0 - incline 1, 1 - straight, 2 - incline 2, 3 - all),
- Gaussian fit (0 or 1),
- Δt_0 ($\pm$ 0.1, 0.2, μs)
- Δgas (or Δg) (0.1, 0.2, % gas, ex. 0.1 = 50.1% Ar, 49.9% CO_2)
- $\Delta Temperature$ (or ΔT) (default = 0)
- B-Field (-1, 0, +1)

The only difference between the two macros is that 'lasertest_single.C' only runs over one user defined selection of Δt_0 and Δg, whereas 'lasertest.C' iterates through many values and outputs results for each selection (this range can be user defined in the code).

The output from the "lasertest*.C" macros includes an additional root file, a log file with residuals and track positions, and a postscript (ps) file containing several histograms plotted from the root file. These files are labeled with the particular Δt_0

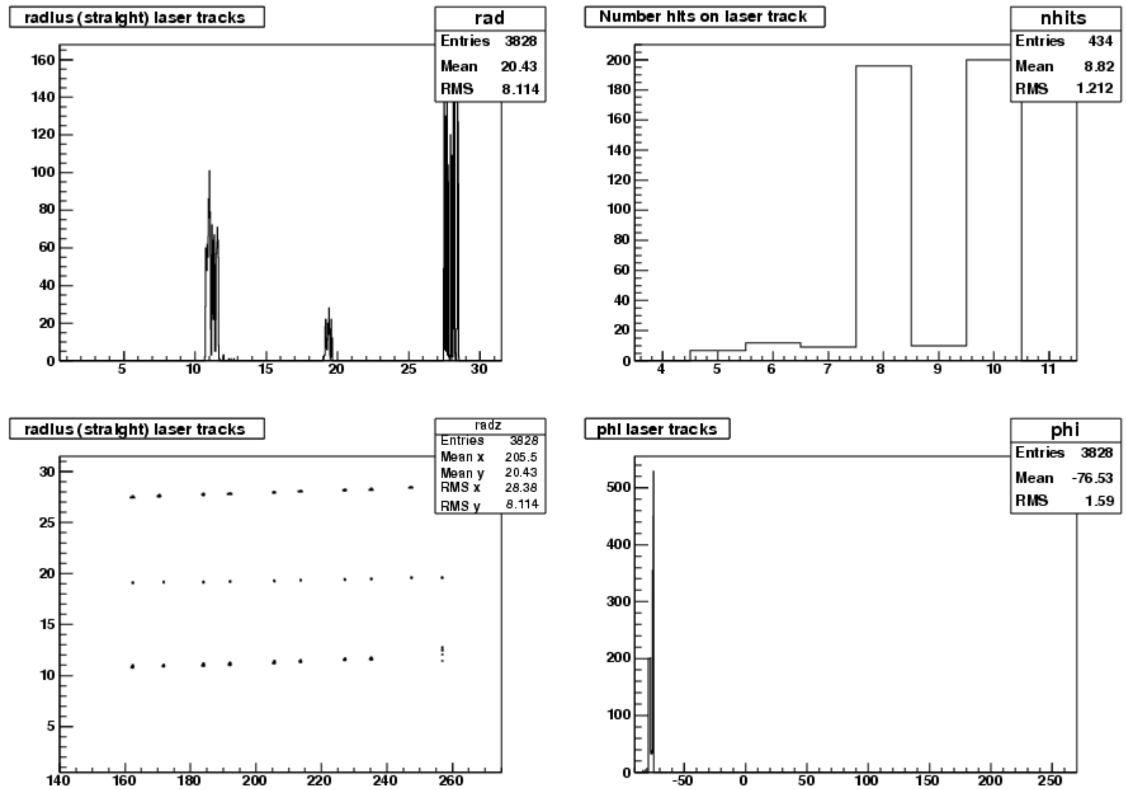

Figure 1: Page 1, clockwise from upper left: Laser ADC vs. radial position in the FTPC; # hits on track; Phi position; Laser radial position vs. z-position.

and Δg values used. The most important plots in the ps file are on the first, second, and third pages. They include:

Figures 1, 2, and 3 are from 2004 AuAu, run 5006004 with no change in Δt_0 and Δg (in reality, a change of 0.00001 in Δg).

The straight beams are used to check the gas composition, temperature corrections and the laser t_0. Since the inner beam has the longest drift distance, it will be more affected by changes to the gas composition and temperature corrections. Conversely,

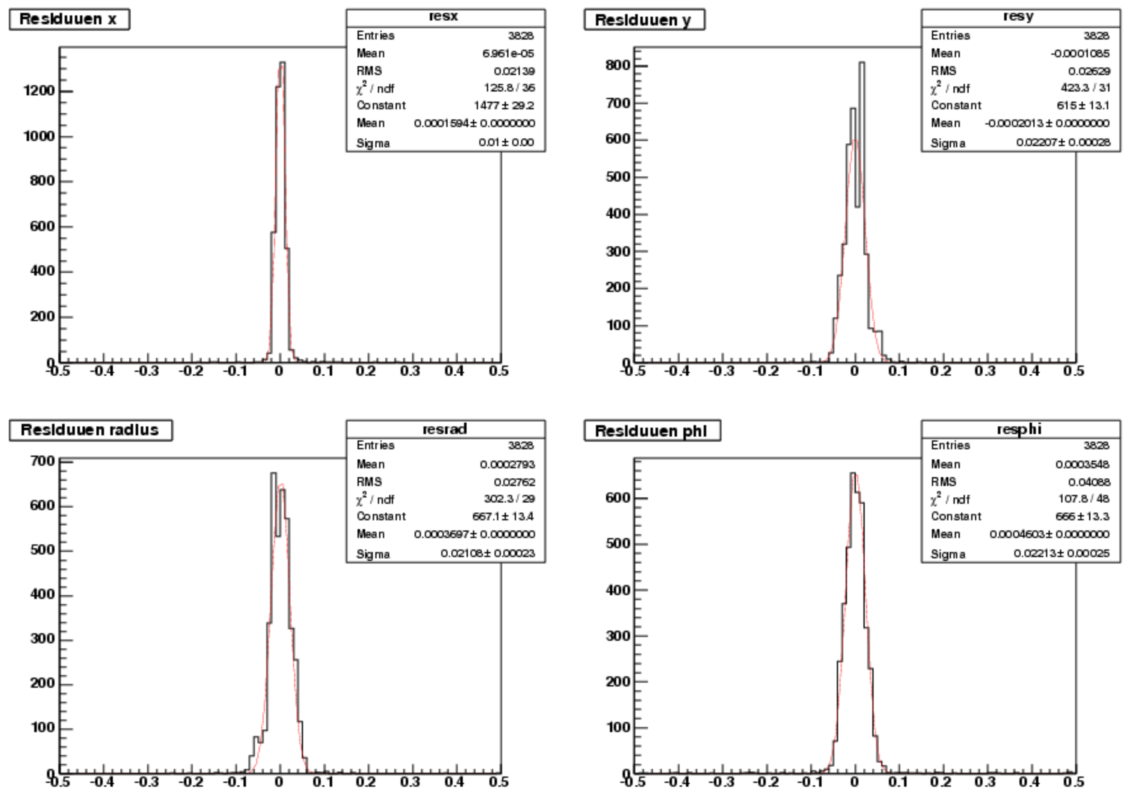

Figure 2: Page 2, clockwise from upper left: x, y, phi, and r laser track residuals

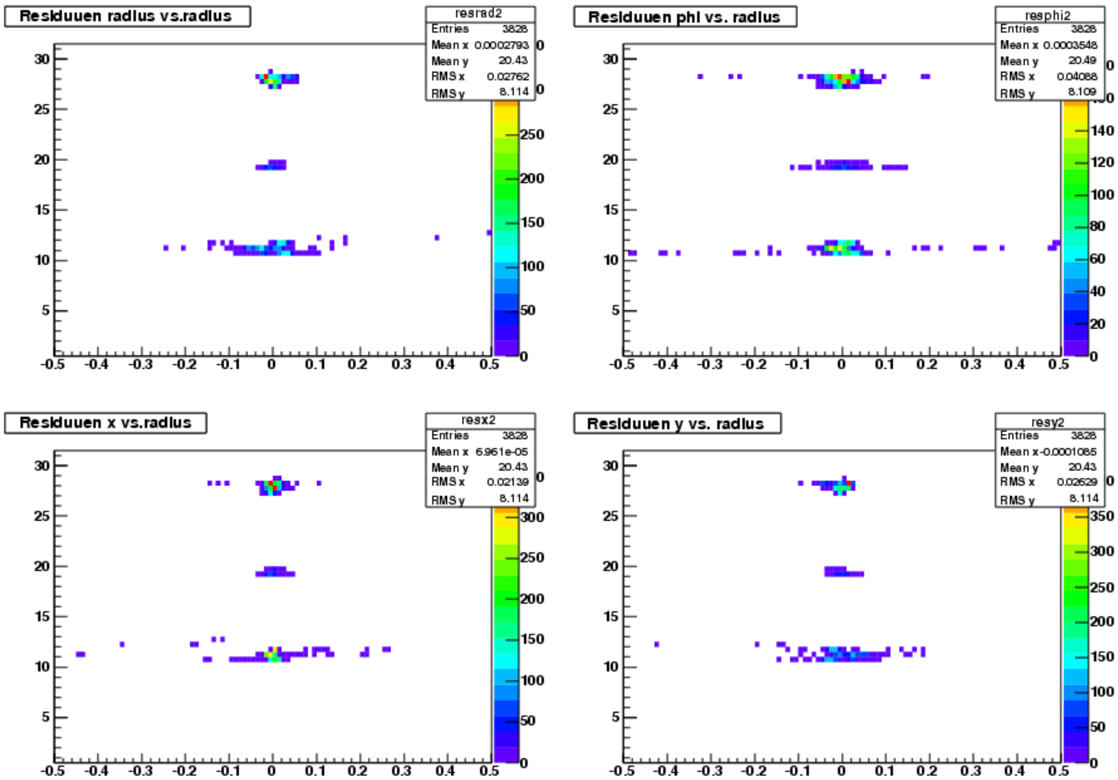

Figure 3: Page 3, clockwise from upper left: r, phi, y, and x laser track residuals vs. radius

the outer beam will not be as affected by the gas composition, but will be influenced more by changes to the laser t_0 [12]. Ideally, the positions of the straight tracks will agree (or be close to) the theoretical positions with no change in either Δt_0 or Δg. If that is not the case, iterating through various values of Δt_0 and Δg is required. If the straight beams are found to be in the proper location after any changes greater than $\pm$ 0.5, this is probably in error, or a serious problem exists with the gas regulation or electronics timing.

A macro called 'laser_2d_pos.C' uses the log files containing residuals and track positions from 'lasertest.C'. All the log files should be combined into one file ('cat' command), and used with this macro. The macro will produce a 2-D histogram of Δt_0 vs. Δg for all three straight tracks [options ("d","colz")], with colors representing the difference between the theoretical positions and the measured positions (Figure 4). Ideally, the smallest difference lies at, or close to, (0,0), or less than $\pm$ 0.1 away from that point.

Additional documentation about these and other macros from "StFtpcCalibMaker" can be found at `http://www.star.bnl.gov/public/ftpc/Software/Calibration/StFtpcCalibMaker.html`

The residuals of the inclined tracks are used to check the $\overrightarrow{E}\mathbf{x}\overrightarrow{B}$ corrections. If the $\overrightarrow{E}\mathbf{x}\overrightarrow{B}$ corrections are accurate, the residuals vs. radius (Figure 3) for the inclined tracks should be relatively straight, and centered around zero. Drawing a vertical line from zero should intersect the residual range for all hit points. If the applied Δt_0 and Δg values are correct, the residuals should be at a minimum value when compared to other settings.

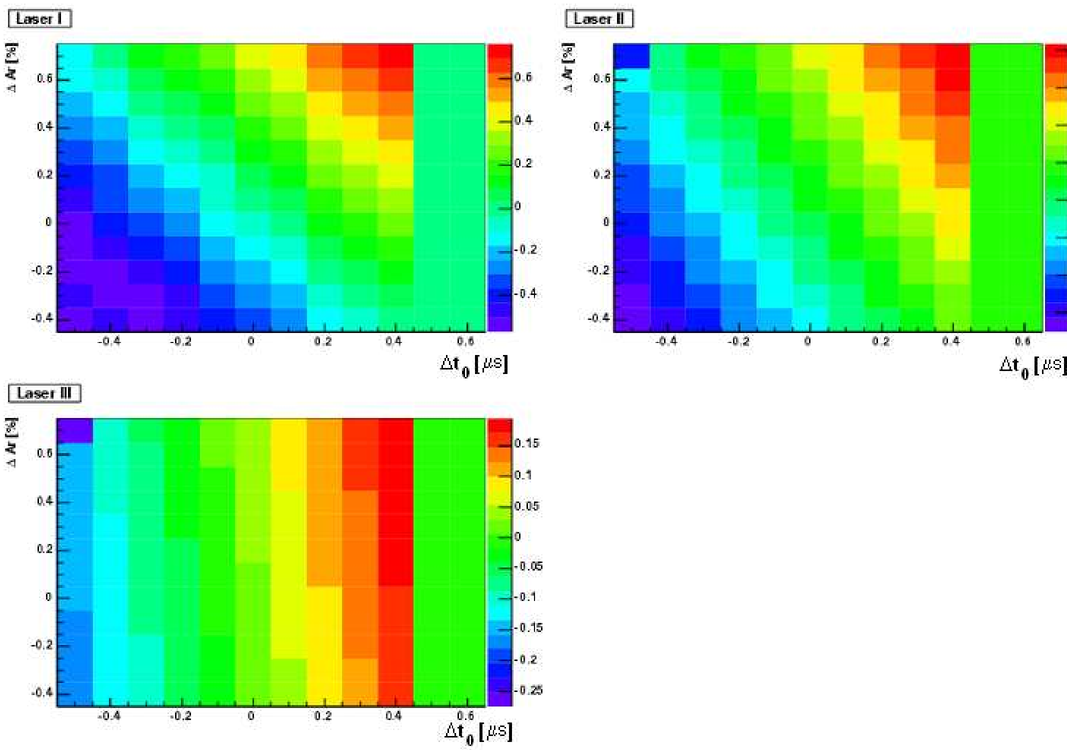

Figure 4: Difference between theoretical and reconstructed radial positions for the 3 straight laser tracks as a function of Δt_0 and Δg

From year-to-year the laser intensity has been declining. At lower intensities, fluctuations will increase and the residuals become worse. This can be seen in the plots of 2003 d-Au, 2004 Au-Au, and 2005 Cu-Cu (`http://www.star.bnl.gov/protected/ftpc/tjt/Laser_Runs_Over_the_Years/`) and 2006 pp, 2007 Au-Au, 2008 d-Au (`http://www.star.bnl.gov/protected/ftpc/jcs/Laser_Runs_Over_the_Years/`).

Remember, data and laser t_0's are not the same. One cannot use laser data to determine if the data t_0 has changed.

3.3 DAQ Mask

The DAQ expert (Tonko) may ask for a masking file to disable noisy electronics. This is produced in a similar fashion to the gain tables, using the "GetGain" program (see "Gain Tables"), except the masking file is created from the macro 'WriteDaqFlagg.C'. The concern with utilizing a DAQ mask is that noisy electronics have been known to fix themselves. It may also be possible to find useable clusters in these areas, regardless of the noise. This should be studied in the future. When these pads are masked out by DAQ, they appear in the FTPC QA plots as dead regions, so there is no way to ensure that they are still "bad" without removing the mask.

A representation of the FTPC DAQ mapping can be seen in Figure 5. A similar graphic (Figure 6) shows the FTPC RDO (readout board) numbering.

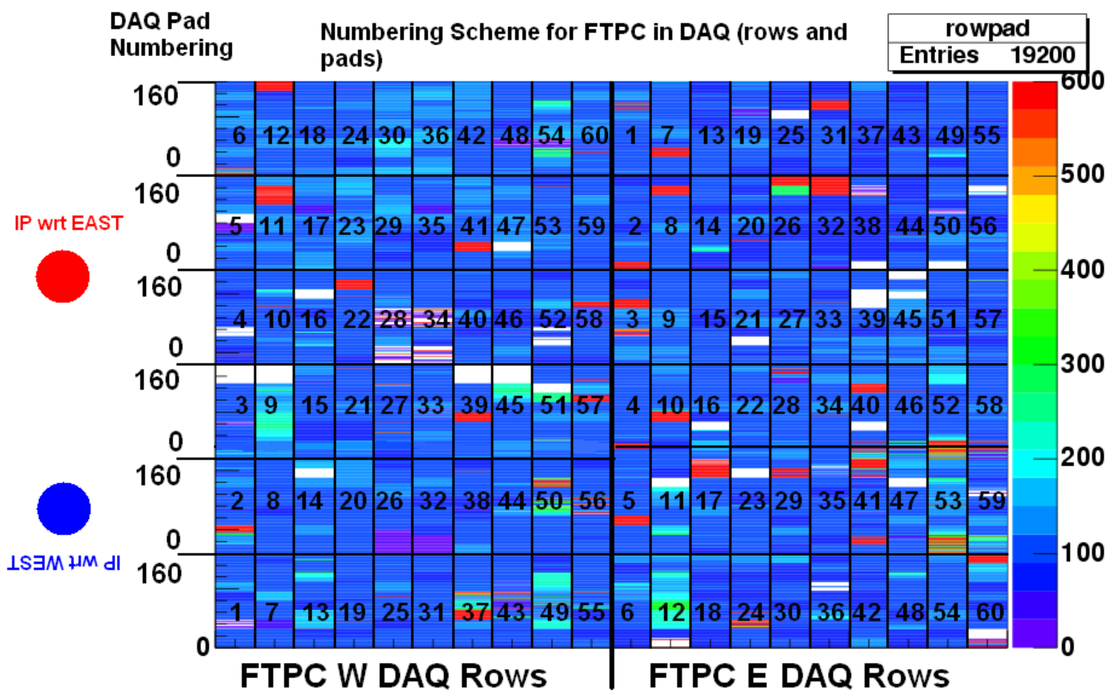

Figure 5: FTPC DAQ Pad Numbering Scheme

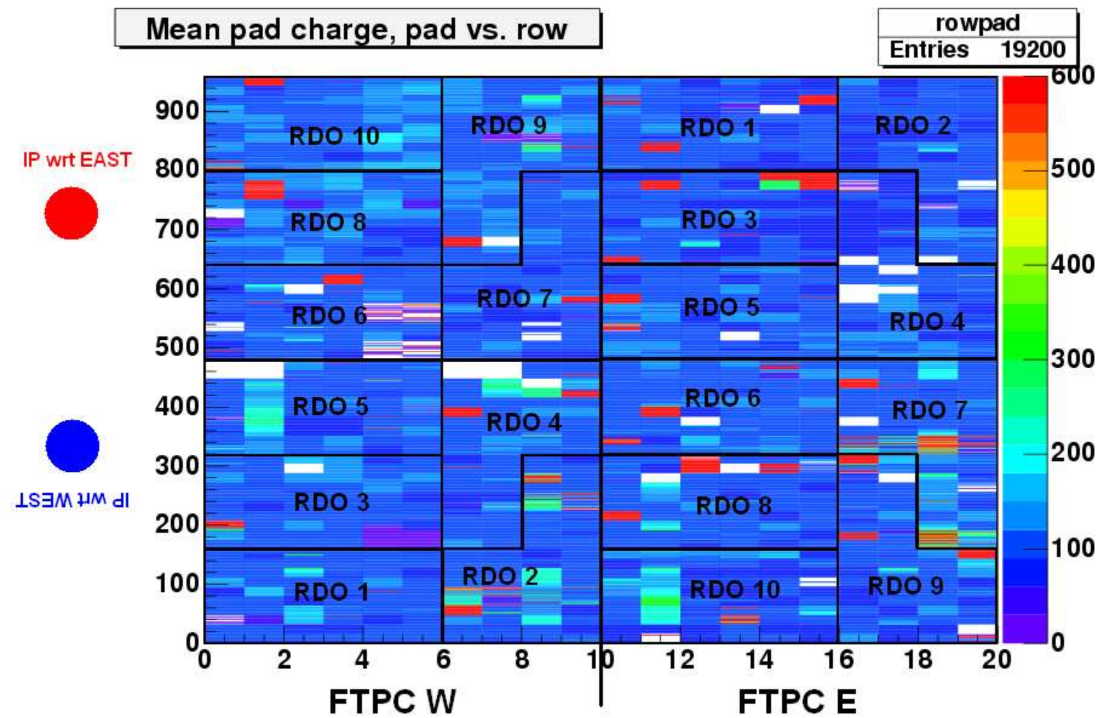

Figure 6: FTPC RDO Numbering Scheme

3.4 Gain Tables

Dead and noisy electronics reduce FTPC tracking efficiency. Dead areas can be corrected with efficiency results from embedding, but noisy electronics can cause problems when reconstructing data. The solution is to mask out these noisy electronics after data taking, but before data reconstruction. This is accomplished using a gain table. The gain table multiplies each pad by a calculated gain factor. If a pad exceeds a user defined noise cut, the gain factor is set to 0 and the noisy pad is now a "dead" region.

Because the percentage of bad FTPC electronics changes over the course of a run, it is probable that multiple gain tables will be required. Changes in either FTPC that would necessitate a new gain table include, but are not limited to:

- Loss of an RDO board.
- Many new and persistent dead/noisy regions.

Creation of a gain table requires three steps. The requisite programs [5] are:

- GetGain (input is a pulser *.daq file)
- FindNoise (input is a data *.daq file)
- WriteAmpSlope.C

All programs are part of the FTPC online software library, which is located on virgo-run09.starp. When you login as 'ftpccrew' the FTPC online library enviroment will automatically be set up for you.

With the advent of DAQ1000, the RTS (Real Time System) library has been moved to the STAR program library in **StRoot/RTS**. Since then all FTPC online library programs can be run on the rcas machines. It is no longer necessary to copy *.daq files from the rcas disks to the virgo-run09 disk.

“GetGain” reads the pulser daq file and creates “Gain.root”, and “Gain.ps” in your working directory, “GainTable.dat” in the “gain_table” subdirectory. “FindNoise” uses the “GainTable.dat” file and implements a noise cut to mask the bad electronics. It creates the files “NoiseCorrected_Gaintable.dat” in the “gain_table” directory, and “NoiseFinder.ps” in the working directory. The first time running FindNoise should be done without a noise cut. The noise cut is decided by looking through the first 20 pages of the NoiseFinder.ps file and choosing a cut that is a several counts above the signal, but that effectively eliminates most noisy channels. Once the choice of noise cuts is made, one can rerun FindNoise with the cut implemented (-Exxx -Wyyy; xxx, yyy (ex. xxx = 200) can be same or different cuts, but for symmetric collisions systems and similar FTPC gain voltage, they will be the same.) The signal will change depending on collision system or energy, so the noise cut should be determined again if either changes. As mentioned previously, it should be studied if noisy areas yield good clusters. If there is no negative affect, or even a net positive affect from not removing noisy electronics, production **using gain tables without a noise cut** may be an option.

WriteAmpSlope.C reads the **NoiseFinder_GainTable.dat** as input and writes out **ftpcAmpSlope.C**. This is the file that is uploaded to the database. It must be named **ftpcAmpSlope.C** [6].

3.5 Rotation Calibration

The 2 FTPCs use tracks to independently reconstruct the primary vertex. These FTPC vertices will not be identical to the TPC vertex due to several factors. Factors include changes in t_0 (affects z-vertex component), the long lever arm of FTPC tracks to the primary vertex, and a slight physical shift (or rotation) about the FTPC mounting points. This slight shift is exacerbated by the long lever arm when projecting tracks to the primary vertex, resulting in an offset of several millimeters in the transverse (x, y) plane. This offset needs to be corrected to ensure that the FTPC and TPC vertices match as closely as possible.

To ascertain the offset values, several thousand events (∼10,000) from test production data, should be examined. This test production can be obtained by asking Jerome Lauret and Lidia Didenko. Once the files are in place, the QA histogram (found in the *.hist.root files, EventQA) **STEMBQaVtxFTPCE[or W]TpcXY** should be drawn. The *.hist.root files can be added together with the macro **bfcread_hist_files_add.C** [7]. The input for this macro is a text file that lists all the *.hist.root files to be added. Redirecting the output of an "ls" of all the *.hist.root files to a text file will create the input file. Other command line parameters can be found in the header of the macro itself [7].

There are two vertex histograms, one for each FTPC. Taking the x and y projection of each of these yields a 1-D histogram with a Gaussian profile that shows the x or y vertex hit distribution. The offset is the mean of a Gaussian fit to these histograms.

Due to some confusion with the x, y coordinate definitions, the negative of both x values obtained from the Gaussian mean should be used in the database (eg. x $\rightarrow$ -x).

Once these values are in the database, the same events should be reproduced with the newly implemented corrections. The Gaussian mean should be closer to 0, and under 1 mm.

In 2004 it was discovered that the reconstructed FTPC vertices were different for various STAR magnetic field settings. There appeared to be an almost independent, orthogonal shift in the vertex position for each FTPC (one moved mostly in the x-direction, the other in the y-direction). When the magnetic field polarity was returned to its initial setting, the reconstructed FTPC vertices almost, but not quite, returned to their initial values. This could possibly be due to a small movement of the entire STAR detector and magnet coils due to a change in the force vector after a polarity flip. The TPC has reported a minor unexplained effect that may be related to this issue.

The solution was to perform this rotation calibration after every change in the magnetic field configuration. Instead of one overall data sample, several were used, one from each magnetic field setting. These data samples are all produced with the rotation correction off (0, 0).

This procedure must be done every time the FTPCs are removed and replaced from the detector, and the result should be checked for the case where the FTPCs are not

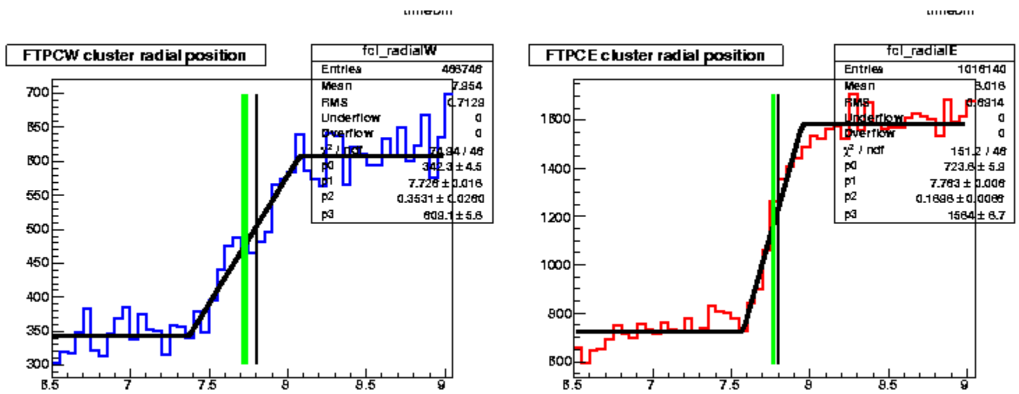

Figure 7: The FTPC cluster radial position (radial step) at ≈ 7.8 cm, the start of the inner volume of the FTPC.

removed.

3.6 FTPC Cluster Radial Step

The useable, inner volume of the FTPC begins at approximately 7.80 cm from the beam line. Therefore, good, reconstructed clusters should not be found at distances smaller than 7.80 cm. There are a small amount of bad clusters, either from electronics noise, beam background, or other sources, that are reconstructed below 7.80 cm. At ∼7.80 cm, there is a rapid rise in the cluster count, seen in Figure 7.

Figure 7 is the FTPC radial step (also referred to as cluster radial position). The half height of the radial step should be at 7.80 cm and both FTPCs should have the radial step in the same position. Some leeway exists, the radial step may be at 7.79 or 7.81 cm (both FTPCs should track together), but larger deviations indicate a problem. If the radial step is not at ∼7.80 cm, the most likely culprits are bad temperature measurements, or a change in t_0. A problem with the cathode or anode voltages can also cause a change in the cluster radial position. (The data in this figure is from Run 8345052. 730 events were reconstructed using $t_0 = 2.73$)

Because it is an important indicator of overall FTPC health, the radial step plots are included in the QA histograms. When checking the position of the radial step, one should use a sample of several thousand events.

If there is a deviation in the position of the radial step that both FTPCs share, the likely culprit is a change in t_0. A t_0 change will manifest in any reproduced data as a shift in the radial step position proportional to the change in t_0. Test data should be examined for this possibility before real data production begins.

If the radial step has shifted in a dissimilar fashion for one or both FTPCs, it is likely related to temperature readout or voltage problems. If there is a problem with the voltages, it will be apparent from the QA files or by checking the SC database. This type of problem is corrected during the run.

If there is a temperature problem that does not affect a large data sample, these runs should be marked as unusable. If it affects a large portion of the data, it is possible to enable a temperature offset that "fudges" the position of the radial step. This is an option of last resort.

3.7 t_0

The FTPC t_0 is the time between the event trigger and the readout of the pads, initiated by discharge of capacitors in the switched capacitor array, part of the FEE cards.

Since this occurs when a signal reaches the readout electronics, t_0 is sensitive to the total drift time of charge in the detector. There are two useful methods to check the values of t_0, both of which are also related to the drift characteristics of the FTPC gas.

The position of the FTPC radial step can be used to determine if there has been a change in t_0. If the FTPC temperatures are known to be operational and other problems can be ruled out, the deviation of the radial step from ∼7.80 cm could be due to a change of t_0, if the radial step is shifted in the same fashion for both FTPCs. Sample data can be used to check the current t_0, and be used to test changes to t_0. This can be done with a local copy of 'StarDb/ftpc/ftpcElectronics.C' and modifying 'StFtpcClusterMaker.cxx' to use this local copy. Increasing (decreasing) t_0 will decrease (increase) the position of the radial step. To check this change carefully, fit the radial step with a Gaussian and use the fit results to determine the shift.

A second method to check the accuracy of the current t_0 is the z position of the FTPC reconstructed vertices wrt the TPC z-vertex. The difference between the FTPC vertices and the TPC vertex in the z-direction should be 1 millimeter or less. This does not hold for data taken at zero field due to the lack of track curvature, leading to a decrease in reconstructed resolution.?? If both W and E FTPC z-vertices are displaced from the TPC z-vertex, a change in t_0 is a likely explanation. Increasing (decreasing) t_0 will decrease (increase) the longitudinal position of the z-vertex from each FTPC.??

The same t_0 value should produce radial step and z-vertex positions within their expected ranges. One should also consult with the TPC calibration experts to ascertain if they have noticed a change in their t_0. Note that the TPC t_0 is not the same as that for the FTPC.

Finally, a new ftpcElectronics table with the new t_0 value must be added to the Calibrations_ftpc offline database with the correct timestamp.

3.8 FTPC Voltage Status

When an FTPC trips the anode voltage shuts off. Since the voltage doesn't go to zero immediately data may be taken with a bad voltage. This data should not be reconstructed.

Jamie Dunlop wrote a perl script (StRoot/StFtpcCalibMaker/macros/GetHV.pl) which gets the FTPC voltage status information from the Slow Control Archive and writes it out in a file named hvtransitions.txt. Dmitry Arkhipkin uses this file to fill the Calibrations_ftpc/ftpcVoltageStatus table for us.

The ftpcVoltageStatus table has a time-stamped entry for statusEast and statusWest for each time the voltage status for FTPC East or West changes. If the status = 1, the voltage is good for data taking; if the status = 0, the voltage is not sufficient for data taking. This table is checked for each event in **StFtpcClusterMaker** to determine if the data for this event and FTPC West and/or FTPC East should be reconstructed.

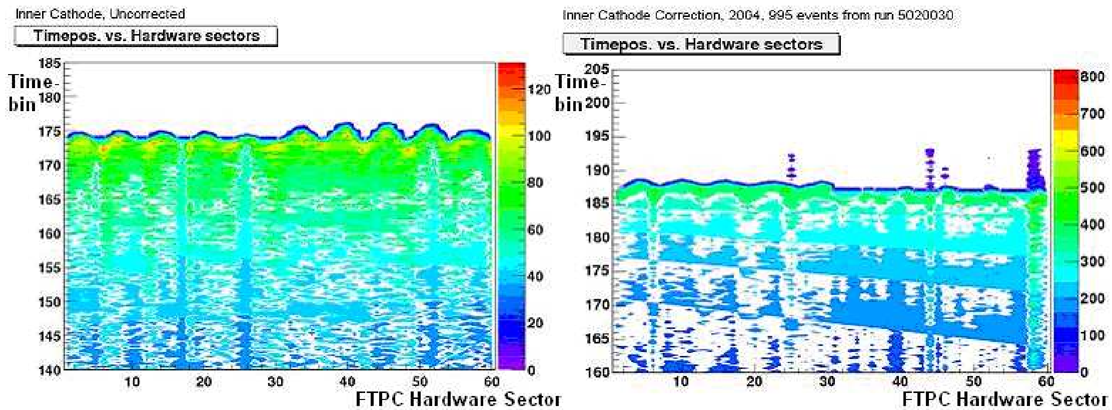

Figure 8: The time position of clusters at the outer radius of the FTPC as a function of hardware sector. The affect of the inner cathode offset is clearly seen as an oscillatory structure in both FTPCs (0-30, FTPC W, 30-60, FTPC E) (Left panel). After corrections are applied, the magnitude of the oscillations is reduced (Right panel).

3.9 Inner Cathode Correction

The FTPC utilizes a cylindrical cathode located at the inner radius. This would ideally be located in the direct center of the FTPC and provide perfect cylindrical symmetry for the produced electric field. However, due to slight machining errors, the cathode is not perfectly centered in the FTPC. This shift is on the order of 0.25 mm, but corresponds to a factor of 10 increase at the outer radius, or 2 mm. Additionally, the detector is sensitive enough that the effect of gravity on the inner cathode, which causes a slight warping in the vertical direction, is also noticeable. The inner cathode offset has two, sympathetic effects. It increases (decreases) the electric field in a particular hemisphere, while simultaneously decreasing (increasing) the drift distance, thereby affecting the drift time. This effect was seen as an oscillatory structure in the time position of the chargestep (Figure 8).

The inner cathode correction was implemented for 2003 data, and was reexamined in 2004 to attempt an improvement. There was no compelling result that showed an improvement over the correction already in place. Though this is an overall small affect on the p_T resolution ($<\sim$5%), it is possible that an improved correction can be found. The requisite programs are located in the STAR CVS repository in (still to be determined).

3.10 Online/Offline QA

Quality Assurance (QA) plots are the first step in verifying proper FTPC operation or troubleshooting problems that may arise.
For each new run year, a reference set of FTPC QA histograms should be created for the FastQA crew.
For more information see the link to ‘Data Quality Assurance’ on the FTPC homepage.

3.11 Cluster Finder Tuning

There are several cluster finding parameters that can be changed to determine the optimal setting for each run. When looking for clusters, the FTPC cluster finder (StFtpcClusterFinder) searches for a signal in a window of a particular pad length and time length. Adjustable parameters include:

- minTimebin: Corresponds to the minimum time bin a cluster from the inner part of the detector (longer drift time) can have.
- minTimebinMed: Corresponds to the minimum time bin a cluster from the

middle part of the detector (medium drift time) can have.

- minTimebinOut: Corresponds to the minimum time bin a cluster from the outer part of the detector (shorter drift time) can have.
- maxTimelength: Corresponds to the maximum length (size) in time bins a cluster from the inner part of the detector can have.
- maxTimelengthMed: Corresponds to the maximum length (size) in time bins a cluster from the middle part of the detector can have.
- maxTimelengthOut: Corresponds to the maximum length (size) in time bins a cluster from the outer part of the detector can have.
- maxPadlength: Same as maxTimelength, except refers to size in pads.
- maxPadlengthMed: Same as maxTimelengthMed, except refers to size in pads.
- maxPadlengthOut: Same as maxTimelengthOut, except refers to size in pads.
- deltaTime: Size of cluster search window in time bins.
- deltaPad: Size of cluster search window in pads.
- minChargeWindow: Minimum charge in the search window required to consider signal a good cluster.

The primary goal of tuning these parameters is to maximize the number of hits on track while trying to reduce the track residuals. An estimate for reasonable values of the minChargeWindow can be acquired from the cluster charge histograms based on the location of the peak. The max ADC histograms are additionally useful for troubleshooting, as they should retain a Landau-like distribution while testing these tuned parameters. If they do not, then the tuned parameters are most likely poor choices.

The cluster finding (and track finding) parameters were originally tuned using d+Au data from Run 4. The deuteron side of the d+Au data is a good place for checking cluster finding. Due to the low multiplicity on the d-side and good vertex finding, one does not encounter track-splitting problems to the extent that exist in pure heavy ion collisions.

The cluster finder parameters are stored in the STAR MySQL offline database in **Geometry_ftpc/ftpcClusterGeom** with beginTime="2004-10-01 00:00:00".

Ideally, these parameters should be checked every year AFTER the FTPC calibration has been completed. To check if the cluster and track finding parameters are still optimal, run the standard reconstruction chain on a physics daq file. The "ftpc_hits" and the "ftpc_tracks" histogram sets will be filled and can be printed out with the bfcread_hist_to_ps.C macro [7]. If there are any signs of trouble in either the MaxAdc or charge histograms in "ftpc_hits" the cluster finding parameters should be re-tuned. These 2 sets of histograms should be saved for reference.

Consult Appendix E for detailed instructions on how to tune the cluster finding parameters.

3.12 Track Finder Tuning

There are several track finding parameters that can be changed to determine the optimal setting for each run. Both cluster and track finding are done in the local FTPC coordinate system, which is ultimately transformed into the global STAR coordinate system (via the TPC coordinate system). The FTPC tracking is done starting from the outer pad plane (farthest from the primary vertex). When a hit is found on a pad plane, the tracker looks to the next pad plane for a hit within a defined window. The tracker iterates through these hits and others on subsequent pad planes until either the track is terminated or the residuals of the fit are minimized ??. One possibility to improve tracking is to restrict this search window.

All FTPC global tracks are also primary tracks because no dca cut is used in reconstruction.

There are 4 defined FTPC tracking types:

1. Main Vertex Tracking: Uses the primary event vertex as another point on all found tracks. These are not automatically primary tracks. The resultant track dca from the global momentum fit determines whether the track comes from the primary vertex.
2. Non-Vertex Tracking (Free Tracking): Does not use the primary event vertex.
3. No Field Tracking: Tracking of straight tracks from the primary vertex (B = 0).
4. Laser Tracking: Optimized for laser analysis.

Tracking for real data is accomplished via "Two Cycle Tracking", which uses the "Main Vertex Tracking", followed by "Free Tracking" on hits not flagged by the "Main Vertex Tracking".

Adjustable parameters include:

- maxVertexPosZWarning (cm): If z-vertex position exceeds this value, a warning is issued.
- maxVertexPosZError (cm): If z-vertex position exceeds this value, tracking is not done.
- maxTrackletLength: Maximum length of tracklets, in terms of hits on track. The tracker creates the tracklets, then propagates them to real tracks.
- minTrackLength: Minimum length of a complete track, in terms of hits on track. For any worthwhile momentum resolution, the FTPC needs at least 5 hits on track.
- rowScopeTracklet: Describes the search window for the next point on a tracklet, in terms of successive padrows.
- rowScopeTrack: Same as above, except for complete tracks.
- phiScope: Describes the search window for the next point in terms of contiguous phi segments.
- etaScope: Same as above, in terms of contiguous eta segments.

- maxDCA: Max DCA of tracks from the primary vertex. This value is currently set to 100 cm, meaning every track incorporates the primary vertex. Vertex cuts are made in offline analysis.

- maxAngleTracklet: Maximum angle between two hits on a tracklet.

- maxAngleTrack: Maximum angle between two hits on a track.

- maxCircleDist: Maximum distance of a new hit compared to the fit (1/r) for that track.

- maxLengthDist: Maximum linear distance of a new hit compared to the fit for that track.

Additionally, there are three adjustable parameters for split tracks:

- maxDist: Maximum distance of a new hit compared to the fit for that track.

- minPointRatio: $\frac{Minimum\ \#\ of\ hits\ on\ track}{Possible\ \#\ of\ hits\ on\ track}$. Currently same as "maxPointRatio", 0.5 (5/10).

- maxPointRatio: $\frac{Maximum\ \#\ of\ hits\ on\ track}{Possible\ \#\ of\ hits\ on\ track}$. Currently same as 'minPointRatio', 0.5 (5/10).

These adjustable track parameters are called **code parameters**. Code parameters are the parameters used by the Maker algorithms. They are considered time-independent. All code parameters are stored in the STAR CVS repository in the StarDb branch; the FTPC track finder code parameters in StarDb/ftpc/ftpcTrackingPars.C.

Before starting to tune the parameters, create a baseline set of the **ftpc_tracks** histograms. Then consult Appendix E for detailed instructions.

3.13 FTPC DriftMapMaker

The StFtpcDriftMapMaker software calculates the FTPC electron drift maps for the selected gas composition and magnetic field setting. Ideally the FTPC gas flow should be regulated to 50% Ar- 50% CO2. If the inner laser beams cannot be reconstructed at the correct location,this may be due to either the gas temperature or the gas composition. If there is a major, permanent change in the gas composition, the drift maps for the new gas composition must be calculated. To calculate a set of drift maps for gas compsitions other than 50%Ar-50%CO2 or 50.3%Ar-49.7%CO2 [10] run the FtpcDriftMapMaker.C macro (see Appendix B).

After the drift maps are calculated for the new gas composition, they must be tested by re-doing the laser analysis to check the reconstructed position of the inner laser beams and by reconstructing a physics run to check the radial step position for both FTPCW and FTPCE. See Appendix C for detailed instructions on how to run the laser analysis using a "local" copy of the drift maps.

Once the drift maps for the new gas compostion has been successfully tested, StFtpcDriftMapMaker must be run for each of the 5 possible magnetic field settings to produce a complete set of drift maps. This complete set must then be uploaded into the MySQL offline database [6]

3.14 Embedding

Embedding is required to determine the track finding efficiency of the FTPC due to the limitations of the tracking algorithm, electronics losses, and the finite acceptance of the detector. To calculate this correction, data is created that contains both real tracks and embedded Monte Carlo tracks. This data is then processed through the FTPC data production software and an association can be made between the number of found Monte Carlo tracks and the total number embedded. This provides an estimate of the track finding efficiency.

The FTPC makes use of the "SlowSimulator" ("$STAR/StRoot/StFtpcSlowSimMaker") software during embedding to create these simulated tracks. One adjustable quantity is the "GasGain", whose optimum setting should be determined from comparing the difference between the Charge and MaxADC distributions for clusters-on-track in real versus simulated data. There is a web page with information on the "GasGain" at, `http://www.star.bnl.gov/public/ftpc/Calibrations/simulations/simulations.html`

The Association Maker does associations between embedded and subsequently found MC tracks.

Official embedding requests should be made through appropriate PWG conveners. The **Embedding Coordinator** is in charge of the actual running of embedding jobs. See `http://drupal.star.bnl.gov/STAR/comp/org/embedding-structure` for more information.

Appendices

A Initializing FTPC Database Tables

The FTPC database macros are located in $STAR/StRoot/StFtpcCalibMaker/macros.

These macros can be run when **$(STAR)/StRoot/StFtpcCalibMaker/macros** is in the **Root.MacroPath:** of your **.rootrc** file.

On an rcas machine

stardev

mkdir Work

cd Work

setenv DB_ACCESS_MODE read

Copy the perfect gain table to your working directory

cp /afs/rhic.bnl.gov/star/users/jcs/public/PerfectGainTable/ftpcAmpSlope.C **.**

Download the newest ftpcCoordTrans table to your working directory (yyyy-mm-dd is today's date)

root4star -b -q 'Db_ReadTable.C("Calibrations_ftpc","ftpcCoordTrans","yyyy-mm-dd 00:00:00")

root4star -b -q 'Db_ReadTable.C("Calibrations_ftpc","ftpcGas","yyyy-mm-dd 00:00:00")

Edit ftpcCoordTrans.C so that

row.observedVertexOffsetX[0] = 0; // [0] = east, [1] = west ;

row.observedVertexOffsetX[1] = 0;

row.observedVertexOffsetY[0] = 0; // [0] = east, [1] = west ;

row.observedVertexOffsetY[1] = 0;

Edit ftpcGas.C so that

row.percentAr = 50;

row.percentCO2 = 50;

row.defaultTemperatureWest = 25.5;

row.defaultTemperatureEast = 25.5;

row.adjustAverageWest = 0.0;

row.adjustAverageEast = 0.0;

After the tables are edited, they should be uploaded back into the MySQL Database where yyyy-mm-dd hh:mm:ss is the timestamp Jerome has announced for the new run year.

```
setenv DB_ACCESS_MODE write
root4star -b -q 'Db_LoadTable.C("Calibrations_ftpc","ftpcAmpSlope","yyyy-mm-dd hh:mm:ss")
root4star -b -q 'Db_LoadTable.C("Calibrations_ftpc","ftpcCoordTrans","yyyy-mm-dd hh:mm:ss")
root4star -b -q 'Db_LoadTable.C("Calibrations_ftpc","ftpcGas","yyyy-mm-dd hh:mm:ss")
```

Only users who have been granted permission can upload tables to the database.

B Calculating New Drift Map Tables

Drift maps are produced by the StFtpcDriftMapMaker using the FtpcDriftMapMaker.C macro. To run StFtpcDriftMapMaker (for example for the 49.7:50.3 Ar:CO2 gas composition):

stardev

mkdir Work

cd Work

ln -s StRoot/StFtpcDriftMapMaker/macros/FtpcDriftMapMaker.C FtpcDriftMapMaker.C

nohup root4star -b -q 'FtpcDriftMapMaker.C(2,1.0,-0.3)' & 47.3Ar.LOG &

where:

root4star -b -q 'FtpcDriftMapMaker.C(map,factor,deltaAr)'

map select magnetic field values Default: = 2

= 1 use constant field values

= 2 use mapped field values

factor field scaling factor Default: = 1.0 (full field positive)

> 0.8 full field positive (FF)

< 0.8 && > 0.2 half field positive (HF)

< 0.2 && > -0.2 zero field (ZF)

> -0.8 half field negative (RHF)

< -0.8 full field negative (RFF)

deltaAr the change (+/-) in the percentage of Ar from 50%

FtpcDriftMapMaker.C may take more than 20 minutes to run. Upon completion, it will write out the set of 6 drift map tables for the selected gas composition:

ftpcEField.C

ftpcDriftField.C

ftpcDeflection.C

ftpcVDrift.C

ftpcdDeflectiondP.C

Since the drift map tables always have the same name, you should create a sub-directory for each gas composition and store the tables in the appropriate sub-directory.

C Testing New Drift Map Tables

To test a new set of drift map tables both reconstruction and laser analysis must be run.

On an interactive rcas machine:

```
stardev
mkdir Work
cd Work
cvs co StarDb/ftpc
```

Copy the ftpcDeflection.C,ftpcVDrift.C,ftpcdDeflectiondP.C,ftpcdVDriftdP.C drift map files for the gas composition+magnetic field combination [10] to be tested to StarDb/ftpc

For example: To test the 50.3% Ar:49.7% CO2 gas composition with a full field positive run copy the files from

/afs/rhic.bnl.gov/star/users/jcs/public/Drift_Maps_10kV_50.3_49.7_25.5C/FullFieldPositive

Check out StRoot/StFtpcCalibMaker and StRoot/StFtpcClusterMaker

```
cvs co StRoot/StFtpcCalibMaker
cvs co StRoot/StFtpcClusterMaker
```

Open StRoot/StFtpcCalibMaker/StFtpcCalibMaker.cxx with an editor. Search for all occurrences of USE_LOCAL_DRIFTMAP and follow the instructions.

Then in StRoot/StFtpcCalibMaker/macros edit lasertest_single.C,lasertest.C and gasTemp.C: change

const char *paramsDB = "$STAR/StarDb";

to

//const char *paramsDB = "$STAR/StarDb";

and change

//const char *paramsDB = "$PWD/StarDb";

to

const char *paramsDB = "$PWD/StarDb";

Open StRoot/StFtpcClusterMaker/StFtpcClusterMaker.cxx with an editor. Search for all occurrences of USE_LOCAL_DRIFTMAP and follow the instructions.

Make the new StFtpcCalibMaker and StFtpcClusterMaker libraries:

```
cons
```

Create links to editted macros:

```
ln -s StRoot/StFtpcCalibMaker/macros/lasertest_single.C lasertest_single.C
ln -s StRoot/StFtpcCalibMaker/macros/lasertest.C lasertest.C
ln -s StRoot/StFtpcCalibMaker/macros/gasTemp.C gasTemp.C
```

Copy the **debug.ini** file into your directory:

```
cp $STAR/StRoot/StFtpcCalibMaker/examples/debug.ini .
```

Rerun the laser reconstruction to produce the root file used by the StFtpcCalibMaker macroswith the drift maps for the new gas composition.

Run the laser analysis to check the reconstructed position of the inner parallel laser.

Reconstruct a physics run to check the radial step positions.

D Analyzing Laser Runs

The following instructions use laser run 10166040, daq file st_laser_10166040_raw_5060001.daq as an example. It is recommended to always redirect output and error messages to a log file.

```
mkdir WORK
cd WORK
stardev
cp $STAR/StRoot/StFtpcCalibMaker/examples/debug.ini .
```

STEP 1 Reconstruct laser run

Reconstruct 200 laser events with the ftpc chain: (From run 10166040 for example)

```
root4star -b -q 'bfc.C(200,"flaser fdbg ftpc db globT detDb tpcDb dbutil in dst event",
"/star/data03/daq/2009/166/10166040/st_laser_10166040_raw_5060001.daq")'
```

This will produce the file run_10166040_laser_test.root which is the input file for the StFtpcCalibMaker macros.

The StFtpcCalibMaker macros are explained in `http://www.star.bnl.gov/public/ftpc/Software/Calibration/StFtpcCalibMaker.html`

STEP 2 Chose the "best" laser sector

Run the StFtpcCalibMaker macro lasertest_single.C for all 3 laser sectors for both

Ftpc West and Ftpc East

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,1,1,3,1,”0”,”0”,0,1)’

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,1,2,3,1,”0”,”0”,0,1)’

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,1,3,3,1,”0”,”0”,0,1)’

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,2,1,3,1,”0”,”0”,0,1)’

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,2,2,3,1,”0”,”0”,0,1)’

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,2,3,3,1,”0”,”0”,0,1)’

Each job produces 6 output files. Use the ”radz” histogram (Lower Left) on page 1 of the *.ps output files to select the sector for both Ftpc West and Ftpc East with the ”best” lasers: Ideally there are 5 laser beams/sector: 3 straight lasers parallel to the beam and 2 inclined lasers.

For Run 10166040, laser sector 2 has the ”best” lasers for both Ftpc West and Ftpc East. In Ftpc West laser sector 2, three straight lasers and one inclined laser are reconstructed. In Ftpc West laser sector 2, two straight lasers and two inclined lasers are reconstructed.

STEP 3 Check laser t0

The next step in the laser analysis is to check the laser t0. To do this run the lasertest_single.C macro for the straight lasers for the selected laser sector:

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,1,2,1,1,”0”,”0”,0,1)’

root4star -b -q ’lasertest_single.C(”run_10166040_laser_test”,2,2,1,1,”0”,”0”,0,1)’

The laser t0 is correct when the the outer laser beam reconstructs at the surveyed position. The laser beam surveyed positions are documented in "Measured Positions of the FTPC Laser Beams" at

`http://www.star.bnl.gov/public/ftpc/Calibrations/laser/laser.html` → measured positions.

The histogram "radpol_l3" (LL on page 5 of the *.ps files)

run_10166040_laser_test-pos_w_lsec_2_g_dt0_dg0_dT0.ps

run_10166040_laser_test-pos_w_lsec_2_g_dt0_dg0_dT0.ps

shows the reconstructed position. If the reconstructed and the surveyed positions are the same, proceed to STEP 4 with the lasertest_single.C *.ps files.

If the outer straight laser does not reconstruct to the surveyed position, either the laser t0 or the gas composition is wrong. Run the lasertest.C macro with delta_t0 = delta_gas = delta_Temperature = 0 to step over a range of laser t0 vs. gas composition values. Then run the laser_2d_pos.C macro to draw the 2-D t0 vs. gas histograms. The correct t0 - gas combination can be read from the LL histogram "Laser III (...,...);nominal 28.51 cm - reconstructed position". (Tip: the histogram is easiest to read when displayed full-screen and magnifyed on a computer terminal) Determine the value of delta_t0 needed to move t0 to 0. Rerun the lasertest.C macro with delta_t0=determined value, delta_gas = delta_Temperature = 0. Then rerun laser_2d_pos.C macro. It may be necessary to repeat this process several times before the outer laser beam is recontructed at the surveyed position. When a new laser t0 is determined it should be entered into the MySQL offline data base in the Calibrations_ftpc/ftpcElectronics table "laserTZero" field. The lasertest_single.C macros must be rerun with delta_t0=new laser t0.

STEP 4 Check gas Temperature and gas Composition

If the gas temperature and the gas compostion are correct the inner straight laser beam will reconstruct to the surveyed position. The histogram "radpol_l1" (UL on page 5 of the *.ps files)

run_10166040_laser_test-pos_w_lsec_2_g_dt0_dg0_dT0.ps

run_10166040_laser_test-pos_w_lsec_2_g_dt0_dg0_dT0.ps

shows the reconstructed position. If the reconstructed and the surveyed positions are the same, the laser analysis is finished. If the reconstructed and the surveyed positions are different for either Ftpc West and/or FtpcEast,either the gas temperature or the gas compostion is wrong. Run the gasTemp.C macro with delta_t0=delta_t0, delta_gas=delta_Temperature=0 to step over a range of gas temperature vs. gas composition values for Ftpc West and/or FtpcEast. Then run the gasTemp_2d_pos.C macros to draw the 2-D temperature vs. compsition histograms. The correct temperature - gas combination can be read from the UL histogram "Laser I (...,...);nominal 11.68cm - reconstructed position". Rerun the gasTemp.C macro with delta_t0=delta_t0, delta_gas=0, delta_Temperature=determined value. Then rerun the gasTemp_2d_pos.C macro. It may be necessary to repeat this process several times before the inner laser beam is recontructed at the surveyed position. When the correct delta_T has been determined for both Ftpc West and Ftpc East, it should be entered into the MySQL offline data base in the Calibrations_ftpc/ftpcGas table adjustAverageWest and adjustAverageEast fields.

E Tuning Cluster & Track Finding Parameters

Since the FTPC track finding parameters are influenced by the cluster quality, it is advisable to tune the cluster and the track parameters at the same time.

On an interactive rcas machine

stardev

mkdir Work

cd Work

cvs co StarDb/ftpc

setenv DB_ACCESS_MODE read

Download the most up-to-date ftpcClusterGeom table

root4star -b -q 'Db_ReadTable.C("Geometry_ftpc","ftpcClusterGeom","yyyy-mm-dd 00:00:00")

where yyyy-mm-dd today's date

mv ftpcClusterGeom.C StarDb/ftpc

cvs co StRoot/StFtpcClusterMaker

Edit StRoot/StFtpcClusterMaker/StFtpcClusterMaker.cxx commenting out the following line of code in the InitRun method

m_clustergeo = (St_ftpcClusterGeom *)dblocal_geometry("ftpcClusterGeom");

and adding the following line of code

m_clustergeo = (St_ftpcClusterGeom *)local("ftpcClusterGeom");

in the Init method right after

m_fastsimpars = (St_ftpcFastSimPars *)local("ftpcFastSimPars");

Then

cons

To test different settings of the ftpcClusterGeom parameters, edit StarDb/ftpc/ftpcClusterGeom.C
To test different settings of the track paramters, edit StarDb/ftpc/ftpcTrackingPars.C
Then run the bfc.C macro with the same reconstruction chain and input daq file you used to obtain the set of reference histograms. When the reconstruction job is completed, produce the new ftpc_hits and ftpc_tracks histograms and compare them to your reference set to see what effect your parameter change had.
Repeat the procedure until the results are optimal. When tuning is finished, upload the new ftpcClusterGeom values into the Geometry_ftpc database with the new timestamp [6] and cvs commit StarDb/ftpcTrackingPars.C.

References

[1] bfc is the TLA (Three Letter Acronym) for big full chain

[2] For information on the STAR Offline Database see `http://drupal.star.bnl.gov/STAR/comp/db`

[3] For an explanation of the STAR Offline Database timestamps see `http://drupal.star.bnl.gov/STAR/comp/db/how-to-user/timestamps`

[4] The standard reconstruction chain changes from run year to run year. Ask Lidia or Jerome for the current chain options. When FTPC data is reconstructed with the standard chain, the event vertex is reconstructed from the TPC data and can be used in the StFtpcTrackMaker.

[5] For an explanation on how to run the FTPC NoiseFinder programs see `http://www.star.bnl.gov/public/ftpc/Software/Calibration/NoiseFinder.html`

[6] To use the FTPC offline database macros, add

$(STAR)/StRoot/StFtpcCalibMaker/macros

to the **'Root.MacroPath** in your **.rootrc** file. For a description of the macros see:

`http://www.star.bnl.gov/public/ftpc/Software/Calibration/FtpcOfflineDatabaseMacros.html`

[7] The macros bfcread_hist_to_pc.C and bfcread_hist_files_add.C are located in $STAR/StRoot/macros/analysis

Either copy the bfcread_hist_to_pc.C macro into your working directory and edit the command line arguments for the hit and the track histograms respectively -

MainFile = “*.hist.root”

MakerHistDir = “ftpc_hits” or “ftpc_tracks”

psFile = “ftpc_hits.ps” or “ftpc_tracks.ps”

PrintList = “Ftpc”

and the issue the command

root4star -b -q bfcread_hist_to_ps.C

OR issue the commands directly

root4star -b -q ’bfcread_hist_to_pc.C(”*.hist.root”, ”ftpc_hits”, ”bfcTree”, ”ftpc_hits.ps”, ””, ”Ftpc”)’

or

root4star -b -q ’bfcread_hist_to_ps.C(”*.hist.root”, ”ftpc_tracks”, ”bfcTree”, ”ftpc_tracks.ps”, ””, ”Ftpc”)’

[8] For reconstructing Ftpc laser tracks only a reduced set of chain options are necessary.

root4star -b -q ’bfc.C(2,200,”fdbg flaser ftpc db globT detDb tpcDb dbutil in dst event”,”/filedirectory/filename.daq”)’

[9] debug.ini has one line and three options, ’filename drawcluterhisto drawvertexhisto’. The ’filename’ relates to the name of the laser output file. For example, if “filename” = “laser_test”, the laser output file will have the name “run_xxxxxxx_laser_test.root”, where “xxxxxxx” is the specific run number. The flags for “drawcluterhisto” or “drawvertexhisto” are either 0 (do not write histograms to *.root file) or 1 (write histograms to *.root file)

[10] The complete set of drift maps (FF,HF,ZF,RHF,RFF) for 50.0%Ar-50%CO2 and for 50.3%Ar-49.7%CO2 are stored on AFS in /afs/rhic.bnl.gov/star/users/jcs/public/DriftMaps.

[11] A perfect gain table is stored on AFS in /afs/rhic.bnl.gov/star/users/jcs/public/PerfectGainTable

[12] If the laserTZero value must be changed to reconstruct the outer parallel laser beam at the correct position, this value must be updated in the MySQL offline database table Calibrations_ftpc/ftpcElectronincs.