

MuCap Run8 Analysis Report — UC Berkeley

Tom Banks
Department of Physics
University of California, Berkeley
tbanks@socrates.berkeley.edu

September 13, 2006

Contents

1	Introduction	4
2	Analysis infrastructure	4
2.1	Software framework	4
2.2	Data organization	4
2.3	Miscellany	5
3	Muon detectors	5
3.1	μ SC (& μ SCA)	5
3.2	μ PC1	9
3.3	TPC	13
3.4	Coincidences	14
4	Electron detectors	15
4.1	ePCs	15
4.2	eSC	17
4.3	Coincidences	18
5	Special studies	23
5.1	Spark cuts	23
5.2	CAENs & Electronics	25
5.3	Identification of muon scatters	26
5.4	Identification of cosmics	30
5.5	Impact parameter	31
5.6	Magnetic field effects on the impact parameter	33
5.7	Identification of high- Z capture events	35
6	Lifetime studies	38
6.1	The muon lifetime histogram	38
6.2	Cosmics bin error adjustments	40
6.3	Rebinning	42
6.4	Fit range	42
6.5	Selection of input tables	44
6.6	Fiducial cuts	47
6.7	Uniformity of rates across gondolas	47
6.8	Data selection	52
6.9	Unique electron condition	55
6.10	Final pre-corrected muon disappearance rate	56
6.11	A note on statistics	56
7	Lifetime corrections	57
7.1	Muon scatters	57
7.2	High- Z impurities	60
7.3	Deuterium	65
7.4	Muon diffusion + pressure vessel scattering + impact cut	68

7.5	Detector inefficiencies	68
8	Conclusions and remaining work	69
8.1	Final blinded muon disappearance rates	69
8.2	Post-unblinding steps	69
8.3	Remaining serious concerns	70
8.4	Possible improvements and further studies	71
A	Shared Berkeley and Illinois analysis settings (common.h)	75
B	Berkeley-specific settings (ucb_common.h)	78
C	List of Berkeley analysis modules	82
D	The muon scatter rate shift error calculation	83
E	Addenda	84
E.1	Update on Run8 deuterium concentrations (September 20, 2006)	84
E.2	Detector geometry	85
E.3	Unsolved mysteries	86
E.4	Comparison of Berkeley and Illinois Run8 μ^- results	87

1 Introduction

This report describes the UC Berkeley analysis of the 2004 MuCap data (Run8). The document was composed in preparation for the MuCap collaboration meeting at UIUC on October 6–8, 2006, where we intend to unblind the secret offset to the 2004 DAQ electronics clock frequency. As such, I present my preliminary, blinded result for the effective μ^- disappearance rate in purified hydrogen, determined from fits to the time spectrum of decay electrons.

2 Analysis infrastructure

2.1 Software framework

The Run8 data was analyzed in parallel and semi-independently by myself (UC Berkeley) and Steven Clayton (UIUC). Our analyses both utilize the MIDAS Analyzer framework [1], which is a natural and convenient choice since the MuCap DAQ is MIDAS-based. In order to explain how the MIDAS Analyzer works, it is necessary to first describe the structure of the raw MuCap data. Each of the 2004 runfiles is divided into *blocks* of data collected during periods of DAQ livetime, and each block is subdivided into *banks* named according to the electronics module from which the data was drawn (TDC400s, CAEN v767s, COMET compressors, etc.). The MIDAS Analyzer uses a so-called “multistage” approach to process these data blocks one by one. In each successive stage a software *module* processes one or more of the data banks, usually performing some specific task. The results of the module’s operations are packaged into one or more new banks, which are passed downstream along with the existing banks. The initial analysis stages typically process and prepare the raw data, while later stages perform more complex coincidence operations and fill histograms.

One advantage of the multistage approach is that it allows for standard collaboration preprocessing stages, followed by customized individual studies. MuCap has adopted this style and integrated the Berkeley and Illinois analyses into a common Analyzer setup. Both analyses use the same set of modules to process the raw data and the μ SC detector data, while subsequent processing is relegated to the individual analysis efforts—namely, the MuCap Berkeley Analysis Software (MBAS) [3] and the MuCap Illinois Analysis Software (MIAS) [4]. Listings of the shared and Berkeley-specific analysis settings are provided in Appendices A and B, respectively.

The MBAS modules (see Appendix C) are written in either C++ or the Muon Query Language (MQL) [5], a high-level language created by Dr. Fred Gray to compactly and efficiently form coincidences and create ROOT [7] histograms and Ntuples from MuCap detector data. MQL supports the m4 macro processor [6], which can be used to make MQL code even more compact.

2.2 Data organization

Analyzing the raw Run8 data is a time-consuming process. When we performed a full pass in March–April 2005, it required approximately six weeks to complete. We decided shortly thereafter (at the May collaboration meeting at UIUC) to break the analysis up into stages in order to expedite future processing. This restructuring required fundamental changes to the analysis software. The three stages we implemented were (in order of operation): skimming, Ntuple-production, and Ntuple-analysis.

In the skimming stage, we process the raw data and identify muon arrivals which are pileup-protected in time by $\pm 25 \mu\text{s}$, since those are the only muons which are ultimately allowed to contribute to the final lifetime histograms. All of the data surrounding the pileup-protected muon events is saved for later

processing. To date we have only needed to perform one skimming pass. I should note that several data integrity checks are also performed in the skimming stage, such as μ SC signal matching, eSC CAEN/COMP signal matching, etc.

In the Ntuple-production stage we process the skimmed data and form coincidences among the muon detectors and electron detectors, respectively. These tables of coincidences are written into muon and electron Ntuples for subsequent processing. In this stage we can also perform searches for evidence of high- Z impurity captures in the TPC data.

In the final Ntuple-analysis stage, cuts are made on the muon and electron Ntuple tables, and the muon and electron data is then joined together to form lifetime histograms.

2.3 Miscellany

For most of the duration of the Run8 analysis, there were two blinding operations in effect: the first was the global DAQ clock unblinding, which has been kept secret from both Steve and me, and is designed to prevent us from inadvertently steering our results towards the expected answer. The second was a relative blinding: whenever Steve and I presented results to the collaboration in our weekly teleconferences, we each added our own secret offset to our respective rates. This too served to prevent us from artificially converging towards a common result. The relative blindings were removed in May 2006; we hope to reveal the 2004 DAQ clock detuning and remove the global unblinding in October 2006.

A note on computer usage is perhaps worthwhile. While UIUC made use primarily of its local computing resources at Illinois, most of the Berkeley analysis passes were conducted remotely on the PSI merlin computing cluster. However, after the recent completion of Run10, Berkeley switched over to the relatively new “Big Cat” CPUs in the MuCap barrack—lion, puma, and cheetah—since they generate results much faster than merlin. Regardless of where the actual processing is conducted, the Berkeley analysis is organized and managed by a MySQL database which incorporates run information of various types, collected from different sources. The largest source of information is probably Francoise Mulhauser’s annotated shift summary, in which she categorized runs by fill and TPC voltage, etc.

3 Muon detectors

In this section I describe the manner in which I process the data from the individual muon detectors, and how I join it all together.

3.1 μ SC (& μ SCA)

The μ SC is a critical detector for our experiment, so in Run8 we recorded several parallel copies of its signal for redundancy. In the analysis we essentially try to reconcile these multiple versions of the μ SC signal in a way that maximizes the pileup protection efficiency and protects against spurious detector data. Below I describe the procedures used to obtain the bank “MUSC” of good μ SC times, and the bank “MUFP” of μ SC and μ SCA times used in pileup protection. I should note that Fred Gray led the development of these somewhat sophisticated comparison algorithms, and that they were modified slightly by the introduction of data skimming.

Two “direct” copies of the μ SC signal, parameter 6001 (copy1) and parameter 6011 (copy2), were sent to separate CAEN TDC modules in separate electronics crates. The signals are processed by the shared module `MMuSCANalysis.MQL` in the following sequence to get a list of muon arrival times:

1. *Raw data selection of μ SC copy1 & copy2, μ SCA:* Entries in the bank “HITS” with parameter numbers 6001 and 6011 are selected into tables “copy1” and “copy2”, respectively. A small time offset is added to the copy2 signal (`muSCCopy2TimeOffset`) to align it with the copy1 signal. Likewise, the μ SCA signals labeled by parameter 6002 are selected into their own table and receive a small timing offset `muSCATimeOffset` for alignment purposes.
2. *Afterpulse clustering of μ SC copy1 & copy2, μ SCA:* The copy1 and copy2 signals each have a inherent deadtime of roughly 29 ns, so we impose an artificial deadtime (that is, a non-transitive clustering) of exactly `kMuSCADInterval=29 ns` on both to eliminate the small amount of afterpulsing noise observed in their natural deadtime regions (see Figure 1). Similarly, an artificial deadtime of `kMuSCAADInterval=90 ns` is applied to the μ SCA veto counter’s signal.
3. *Forming of intersection of μ SC copy1 & copy2:* The two μ SC copies are compared to one another, and only those copy1 hits which have a copy2 partner within $\pm$ `kMuSCSignalSTCoincHalfwidth=7 ns` are accepted. If there are too many discrepancies between the two signals—i.e. if there are too many copy1 hits (`kMuSCMismatchThreshold=3`) without a corresponding copy2 partner—then the block is later cut by the `MMuSCAnalysisC.cpp` module.
4. *Imposition of μ SCA veto:* A veto on μ SCA hits is applied to the copy1 hits from the copy1/copy2 intersection.
5. *Exported bank:* The surviving table of copy1 hits is exported as the “MUSC” bank of good muon arrival times, which serve as the seed times (i.e. $t = 0$) for our lifetime measurement.

The astute reader might have noticed a pronounced ~ 50 MHz periodic modulation in the μ SC copy1 signal’s autocorrelation plot in Figure 1, a feature first observed in 2003 during Run7. At that time we suspected that the modulations were due to structure in the PSI cyclotron beam, but we could not rule out the possibility that the CAENs (or other electronics) were responsible since the DAQ clock was also derived from a ~ 50 MHz signal. To resolve the question, we directed a radioactive source onto an old μ SC scintillator paddle midway through Run8, thereby providing a stream of random hits. This random stream was fed through a CAEN module, so if the random source’s signal turned out to exhibit the 50 MHz periodicity, we would know that the problem was somewhere in the electronics. Fortunately, as Figure 2 shows, the random source’s autocorrelation plot is nicely flat (a feature confirmed by Fourier analysis), and we can confidently conclude that the PSI proton beam’s structure is responsible for the periodicity in the muon arrivals. Our method for addressing this unavoidable feature will be discussed later.

The second bank produced by the μ SC analysis is “MUFP,” which is intended for use in pileup protection. It is created in the following manner:

1. *Processing of the μ SC routed signal:* Entries are selected from the bank “HITS” which have parameter numbers between 6006 and 6009, corresponding to the μ SC signals from the router. These router channels are assigned “port” numbers 0–3, respectively, and a time offset is added to each signal according to its port number.
2. *Afterpulse clustering of the μ SC routed signals:* An artificial deadtime of `kMuSCADInterval=29 ns` is applied to each router port to remove double-pulsing.
3. *Merging of all μ SC and μ SCA signals:* The artificial-deadtime-modified μ SC copy1 and copy2 signals are merged with the μ SC routed and the artificial-deadtime-modified μ SCA signal into a single table.

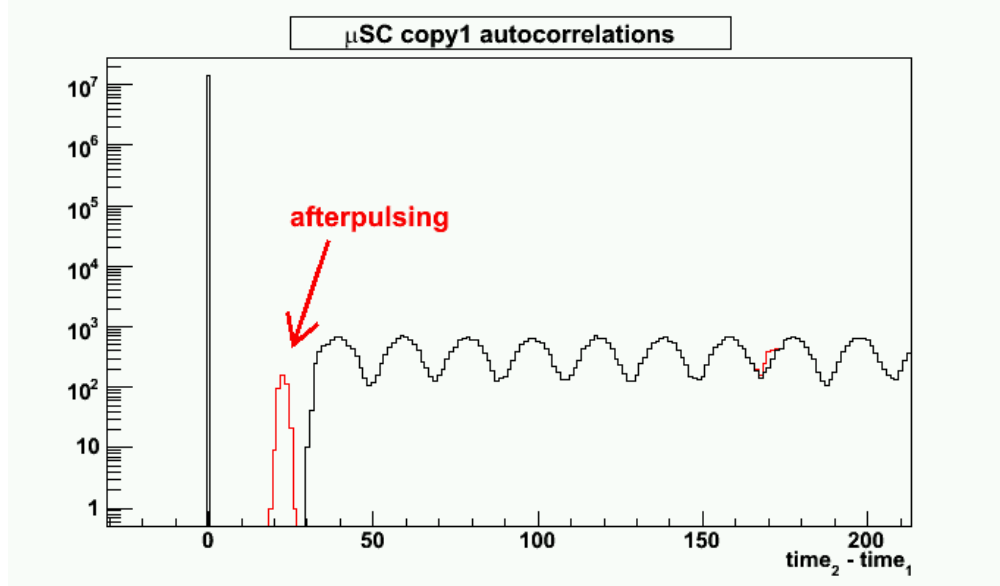


Figure 1: Autocorrelation plots of μ SC times before (red) and after (black) the imposition of a 29 ns artificial dead-time, which eliminates the indicated afterpulsing peak.

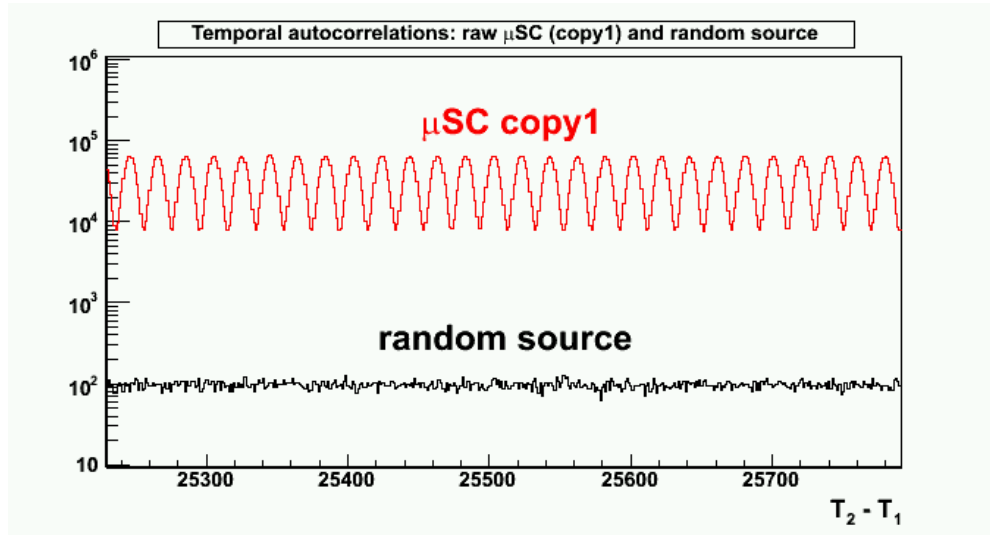


Figure 2: Autocorrelation plots of data from the μ SC copy1 signal and the random source. The random source was implemented in Run8 to test whether the ~ 50 MHz oscillations visible in the μ SC signal were due to beam structure or electronics (i.e. CAEN) effects. The absence of any such structure in the random source autocorrelation reassuringly confirms that the oscillations arise from the PSI cyclotron's proton beam structure.

(Also included in the union are μ SC copy1, copy2, routed, and μ SCA hits which had been previously set aside by the skimmer.)

4. *Clustering of proximate hits:* To prevent the multiple signals from a single muon from cancelling each other out in the later pileup protection—the signals may be slightly staggered in time due to cable lengths variations, etc.—closely-spaced hits ($\pm$ kMuSCSignalSTCoincHalfWidth=7 ns are clustered together. It is important to note that, although an artificial deadtime of 29 ns has been applied individually to each of the constituent μ SC signals, the resulting union of signals has a much shorter effective deadtime of $\sim$ 7 ns (Figure 3), largely thanks to the router. This is advantageous as it implies that the union table will be more effective at pileup protection than the copy1 or copy2 signals alone would be.
5. *Exported bank:* The clustered union of μ SC and μ SCA hits is exported as bank “MUFP” for later use in pileup protection.

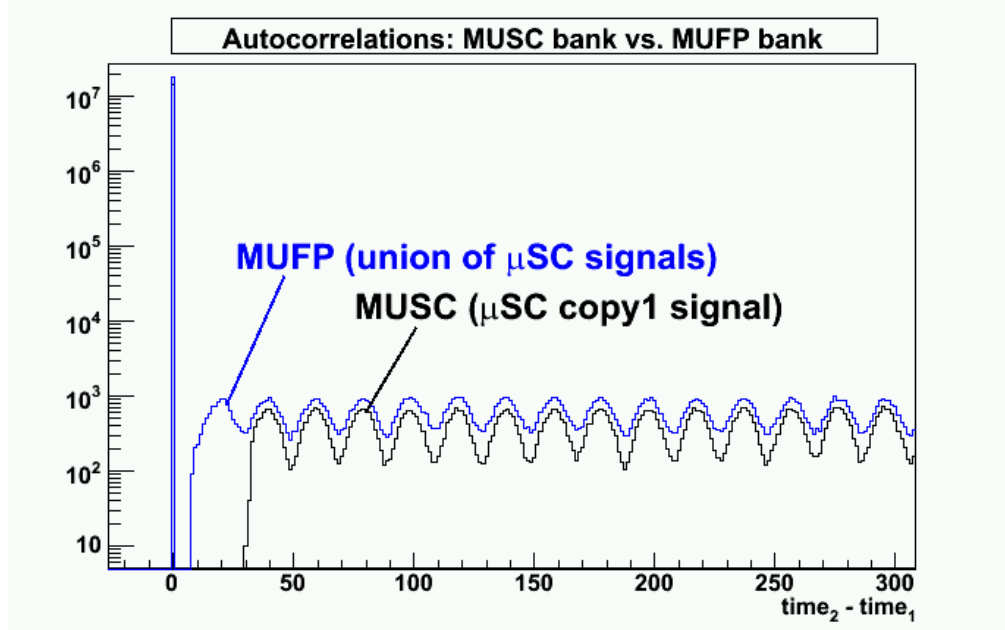


Figure 3: Autocorrelation plots of data from the MUSC and MUFP tables. The MUSC bank, which is drawn from μ SC copy1 signals, has a deadtime of 29 ns, whereas the MUFP bank, which is comprised of the union of all μ SC signals (most notably the router) has an effective deadtime of approximately 7 ns.

Some final notes on the μ SC analysis:

- There existed two additional μ SC signals that were not utilized in the Run8 analysis: parameters 6003 and 6004, corresponding to the gated B and $\bar{B}$ hits.
- There was occasionally evidence of double-pulsing in the μ SC copy1 signal. Fortunately, the existing processing algorithm handles that behavior properly.
- All of the μ SC and μ SCA timing offsets reside in the MIDAS online database (ODB), rather than in the `ucb_common.h` header file.
- The μ SC data processing is the last stage shared by both the Berkeley and Illinois analyses. Beyond this point the two analyses diverge, and subsequent detector processing and coincidence formation is performed according to the discretion of the individual analyzers.

So, from this point on, the detector analysis descriptions are exclusive to the Berkeley software.

3.2 μ PC1

The Berkeley μ PC1 processing sequence is described below. All of it takes place in the module `MMuPC1AnalysisMQL.MQL`, which first processes the X- and Y-planes separately to identify hits, and then the joins the two planes together (if desired) to form a two-dimensional (X,Y) coincidence.

1. *Raw data selection:* μ PC1 wire hits are selected from the raw CAEN data in bank “HITS” (produced by `MCAenCompProcessRaw.cpp`) into either the X- or Y-plane table according to their parameter label. Per-wire autocorrelations formed from this raw data (Figure 4) reveal some unphysical afterpulsing, especially in the X-plane.
2. *Spark cuts:* Once the X- and Y-plane hits have been selected, a search for spark events is conducted. For each plane, clusters are formed of hits within $\pm 1 \mu\text{s}$ of one another. If any of these clusters are larger in size than the `kMuPC1XSparkSizeCutoff=21` threshold in `ucb_common.h`, the earliest time in the cluster is flagged as a spark time, and all μ PC1 hits within the next `kSparkCutDeadtime=50000 ns` are cut. The post-spark-cut X- and Y-plane autocorrelations are shown in Figure 5; it is clear that the spark cuts do not remedy the afterpulsing problem.
3. *Per-wire afterpulse clustering:* To eliminate the afterpulsing noise, “artificial-deadtime” (AD) clustering is performed separately on each wire: namely, all hits within `kMuPC1XClusterInterval=260 ns` of an initial hit are clustered together. The resulting autocorrelation plots (Figure 6) demonstrate that this AD clustering operation eliminates the afterpulsing noise.
4. *Clustering across wires:* Once the individual wires’ signals have been “cleaned up” by the AD clustering, a clustering *across* wires is performed for each plane. All hits within $\pm \text{kMuPC1XClusterInterval}$ and where $(\text{abs}(\text{Xwire}_1 - \text{Xwire}_2) - 1) \leq \text{kMuPC1XClusterMaxGap}$ are grouped into a single hit. The cluster’s assigned time is the earliest time in the cluster, and the position is assigned as the average of the contributing wires. The resulting profiles of the X- and Y-planes are shown in Figure 7.

5. *Joining of X- and Y-plane clusters:* The X- and Y-plane tables are first joined together in time over a wide interval. Based upon the corresponding time difference plot, shown in Figure 8, an X- and Y-plane hit are considered coincident if ($\text{abs}(\text{XYtdiff}) \leq 95$). The resulting two-dimensional image is shown in Figure 9.
6. *Exported tables:* Four banks are exported downstream—the X-plane hits (“MP1X”), the Y-plane hits (“MP1Y”), the XY-coincident hits (“MPC1”), and the spark times (“M1SP”).

The individual X- and Y-plane tables are later merged into a single $\mu\text{PC1XY-OR}$ table, while the coincident table is considered $\mu\text{PC1XY-AND}$. These two tables provide slightly different levels of pileup protection, as will be discussed later. The spark times are used downstream in μSC spark cuts.

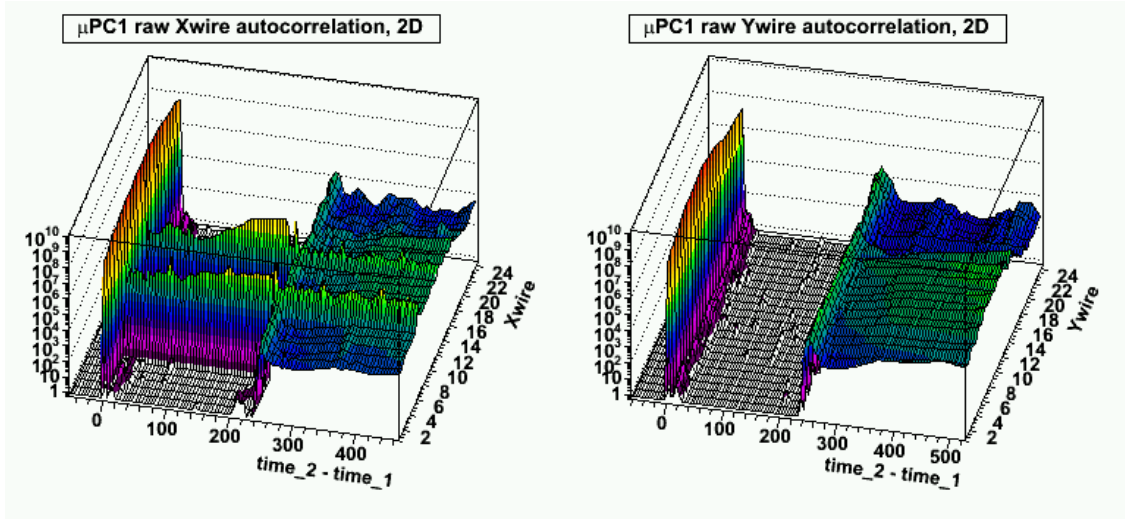


Figure 4: Two-dimensional autocorrelation plots of raw hits in the μPC1 X and Y planes.

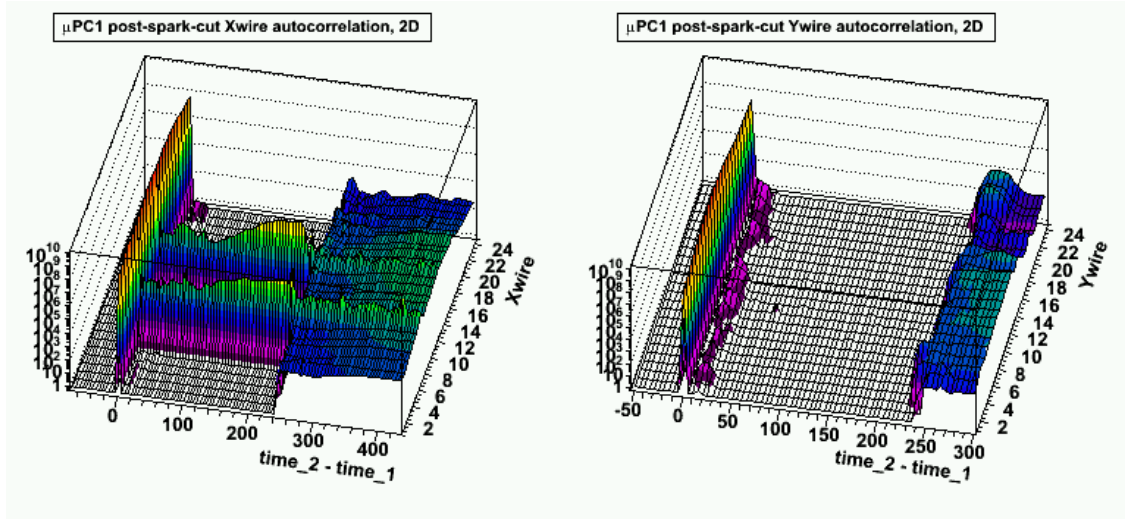


Figure 5: Two-dimensional autocorrelation plots of hits in the μ PC1 X and Y planes after spark cuts.

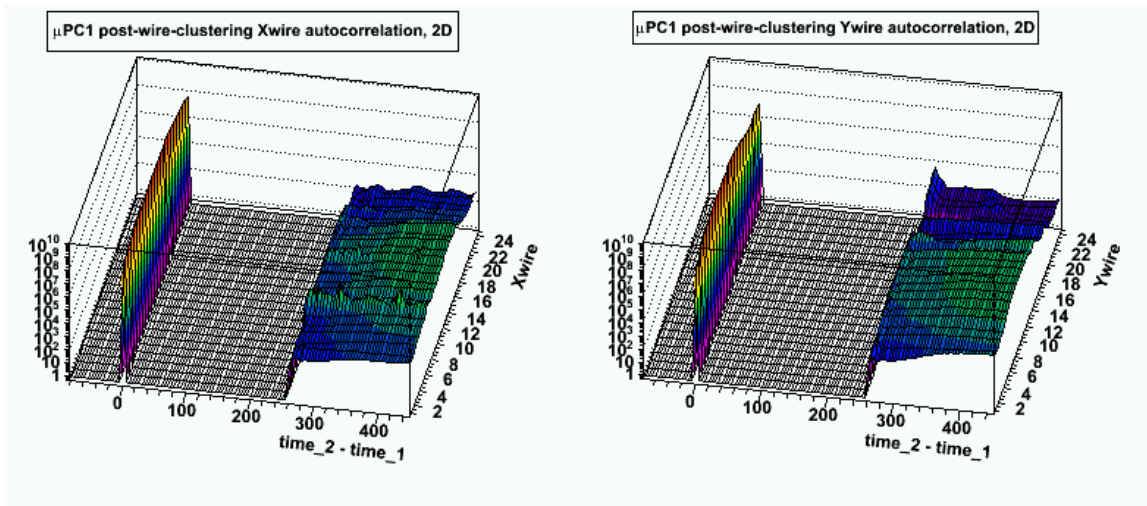


Figure 6: Two-dimensional autocorrelation plots of hits in the μ PC1 X and Y planes after spark cuts and artificial deadtime clustering.

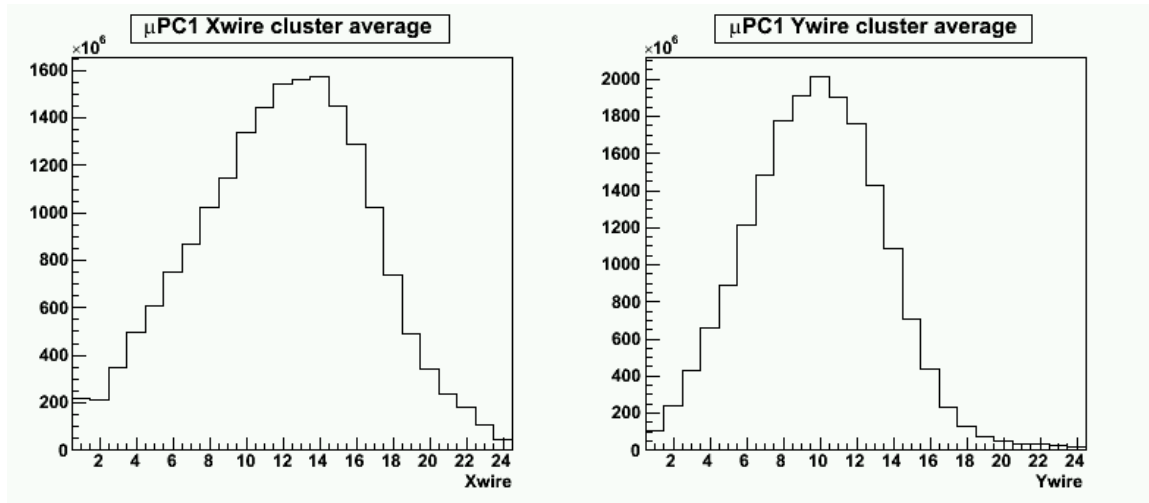


Figure 7: Profiles of hits in the μ PC1 X and Y planes after spark cuts and clustering. The position of a hit cluster is calculated as the average of the contributing wires.

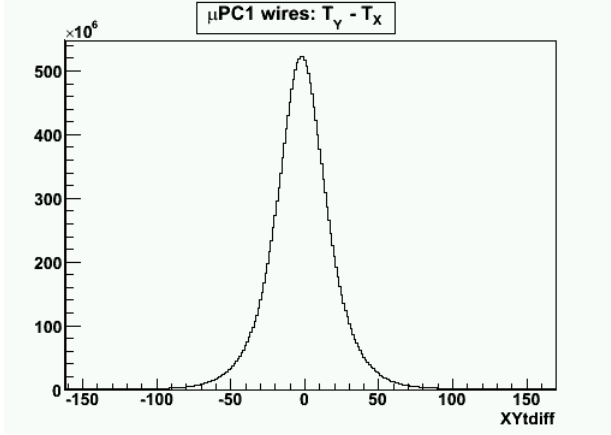


Figure 8: Time differences between hit clusters in the μ PC1 X and Y planes.

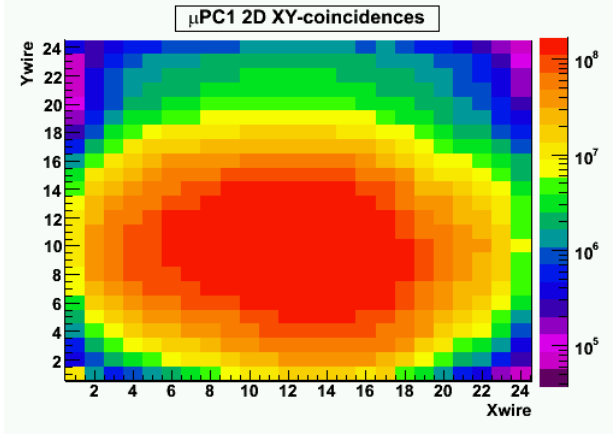


Figure 9: Final distribution of μ PC1 two-plane hits, after selecting temporal coincidences between the X and Y planes in the central peak in Figure 8.

3.3 TPC

In earlier versions of the Berkeley analysis (prior to August 2005), the identification of muon stops and the application of some fiducial cuts took place before the TPC data was joined with the μ SC or μ PC1 data. Unfortunately, this early, upstream specification of the TPC cuts constrained the analysis and prevented us from studying their effects on the fitted decay rate. We therefore decided in August 2005 to reconfigure the code to first gather information about muon stop candidates, and to postpone fiducial cuts until just prior to the creation of lifetime histograms. The items below thus describe the TPC mustop candidate information-gathering steps only; the fiducial TPC cuts will be discussed later.

1. *Raw data selection:* The module `MTDC400ProcessRaw.cpp` processes the TDC400 data in banks “TDC1” through “TDC7” into a bank “STCK,” so named because it is conceptualized as a time-ordered, three-deep stack of pixel hits from the EL, EH, and EVH thresholds. This module also produces a bank “TSPK” of TPC spark candidates, which was implemented to notify the skimmer to save data around such events.
2. *Anode EH segment identification:* The module `MTPCSegmentSniffer.cpp` searches through the “STCK” bank for groups of nearly contiguous anode EH pixels (also referred to as anode EH “segments”, or “islands”), allowing a gap of one pixel only. The boundaries of the EH segments are ascertained, and the information is placed into the bank “AND1.”
3. *Gathering of information about anode EH segments:* Using the “AND1” bank’s anode EH segments as starting points, the module `MTPCTrackAnalysisC.cpp` gathers information about the anode EL and EH pixels surrounding each anode EH segment. The module employs the same “sniffing” algorithm as `MTPCSegmentSniffer.cpp` to collect this information, but this time a gap of two pixels in either direction is allowed. The software counts anode EL and EH pixels and ascertains the extent of pixels in all directions. The software also collects concurrent strip EL and EH information, both along the length of the track (so that x-escapees can be identified) and immediately above the anode EH pixels (to get the x-position of the mustop candidate). All of this information is passed downstream in the bank “TRKS,” so that cuts can be performed later on these mustop candidates.

3.4 Coincidences

In forming coincidences among the muon detectors, I have adopted what I call an “approval” philosophy: I form all possible (realistic) coincidences among the detectors, make appropriate cuts, and then reduce the surviving coincidences down to a unique set of μ SC times for use in the lifetime histograms. Below I outline the algorithm I have implemented in the module `MCoincidenceCalcs.MQL` to construct coincidences among the μ SC, μ PC1, and TPC detectors.

1. *Preparation of μ SC hits:* I first prepare the μ SC data by taking the following actions.

- (a) *Bookending:* In the so-called “bookending” procedure I cut any μ SC hits within $\pm k\text{MuSCBookendWidth}=40500$ ns of the edges of the data block. This action is necessary in order to obtain a flat background in the lifetime histograms which are created later.
- (b) *Local spark and CAEN trailing edge cuts:* So-called “local” sparks indicate that a single detector has sparked in some fashion and is likely to need some recovery time (local sparks in the μ PC1, ePC1, ePC2, and eSC detectors are considered here; the TPC is *not*). Similarly, μ LAN studies discovered that the appearance of CAEN trailing edges—when the CAENs are operating in leading-edge-only mode—indicates a memory buffer overflow and probably a corresponding data loss for a short period of time. Rather than take the draconian action of cutting the entire data block containing local sparks or trailing edge events, I cut any μ SC hits within a time interval of $[-5000, k\text{SparkCutDeadtime}=50000]$ ns around them (the spark cut interval of $50\ \mu\text{s}$ was chosen on Dr. Peter Kammel’s recommendation, based upon wire chamber recovery timescales). I should note that, in addition, I bookend the μ SC data around the electron detector spark cuts and the electron CAEN trailing edge cuts, for the same reasons that I bookend the μ SC data on the edges of the data block.
- (c) *$25\ \mu\text{s}$ pileup-protection* From the surviving μ SC hits, I select those which are pileup-protected in time from all other μ SC hits (recall bank “MUFP”) by $\pm 25\ \mu\text{s}$.

2. *Construction of μ SC and μ PC1 coincidences:* This sequence of operations is somewhat complex, so I will give only a general description. Basically, I identify μ SC hits which have a corresponding (temporally coincident) μ PC1 hit, and then I use any remaining, “unassociated” μ PC1 hits for pileup protection. This latter operation is of great importance, as the μ SC is known to have suffered through periods of low efficiencies (perhaps due to misalignment), in which case the μ PC1 rescues the pileup protection to a large extent (see Figure 10).

I ultimately create two sets of μ PC1-processed μ SC hits: one set where I have only used μ PC1 data where there was an X- and Y-plane coincidence ($\mu\text{PC1XY-AND}$), and one set where the μ PC1 X- and Y-planes are treated as independent detectors ($\mu\text{PC1XY-OR}$). The $\mu\text{PC1XY-OR}$ usage provides somewhat more rigorous pileup protection, as will be discussed later.

3. *Join of μ SC/ μ PC1 data with TPC data:* I perform a generous temporal join of the μ SC/ μ PC1 data with the TPC data over the interval $[-7000, 29000]$, which is somewhat wider than the actual TPC drift interval. The reason for this is that I postpone the fiducial cuts until later downstream in the Ntuple analysis, where I can study their effects on the fitted lifetime. I should emphasize that all of the muon detector coincidences are purely temporal, and no spatial coincidences (e.g. μ PC1/TPC track matching) are performed.

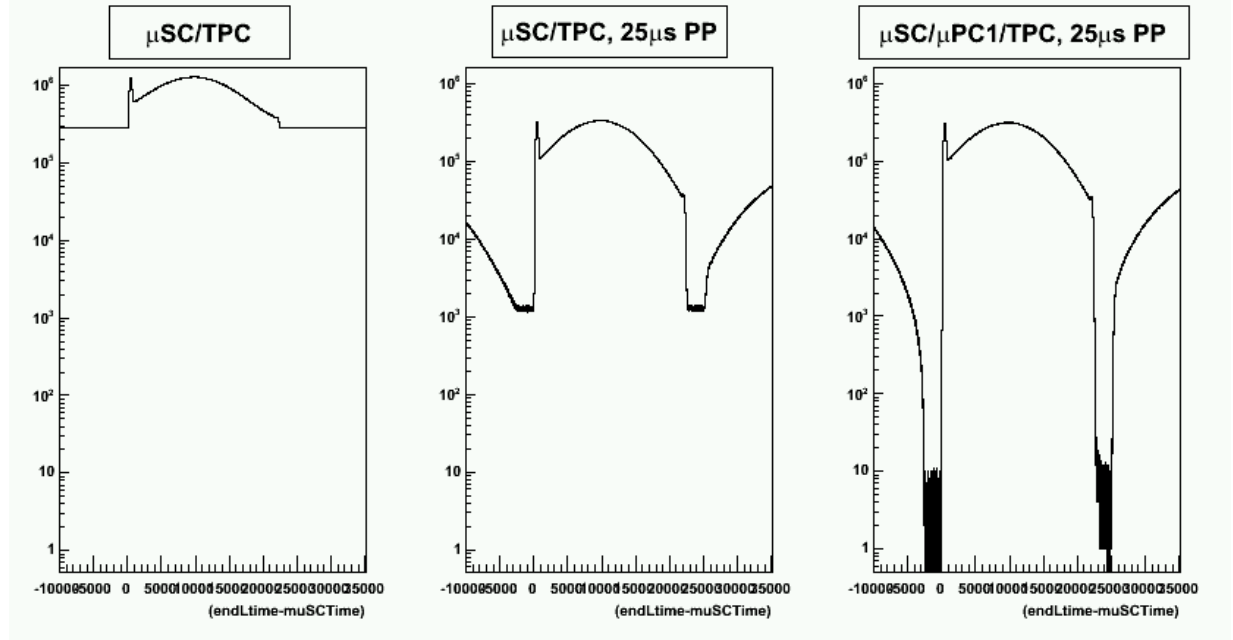


Figure 10: These TPC drift plots—time-integrated over the Run8 data—illustrate the important role that μPC1 plays in pileup protection. The leftmost plot does not involve any pileup protection, the middle plot uses only the μSC for $25\ \mu\text{s}$ pileup protection, and the rightmost plot uses both the μSC and μPC1 for $25\ \mu\text{s}$ pileup protection. Notice the dramatic improvement in pileup protection when μPC1 is incorporated. The remaining accidental background visible between the central peak and the pileup edges is due to time-independent inefficiencies in the MuCap detectors, and possibly TPC noise masquerading as muon stops.

4. *Exported tables:* I export two Ntuples of $\mu\text{SC}/\mu\text{PC1}/\text{TPC}$ hits: one constructed from the $\mu\text{PC1XY-AND}$ coincidences, and one involving the $\mu\text{PC1XY-OR}$ coincidences. The relative merits of these two tables will be discussed later.

At this point the muon detector data is ready to be joined with the electron detector data.

4 Electron detectors

4.1 ePCs

The two ePCs (ePC1 and ePC2) are processed in identical fashion in the Berkeley analysis code, although the exact parameter settings can vary between them (see the `ucb_common.h` specs in Appendix B). The data sets from each of the ePCs’ three “planes” (anode, inner cathode, outer cathode) are also processed in a nearly identical manner before coincidences are formed among them. Thus, in the steps below I first describe the general procedure for processing the data for an individual ePC plane, and then I describe how the planes are joined together in coincidence. The processing is carried out by the modules `MePC1AnalysisMQL.MQL` and `MePC2AnalysisMQL.MQL`.

1. *Raw data selection:* The ePC wire hits are selected from the raw CAEN data in bank “HITS” (produced by `MCAenCompProcessRaw.cpp`) into anode, cathode-inner, and cathode-outer tables.

2. *Spark cuts (SC)*: Once the anode and cathode data tables have been created, a search for spark events is conducted. For each plane, clusters are formed of hits within `kePC*ADInterval` of one another, where `*`=Anode, InnerStrip, or OuterStrip. (There are two versions of the clustering intervals, which I introduced to check for their effects on the final result.) If any of the clusters are larger in size than the `kePC*SparkSizeCutoff` threshold specified in `ucb_common.h`, the earliest time in the cluster is flagged as a spark time, and all of the plane's hits within the next `kSparkCutDeadtime=50000` ns are cut.
3. *Per-wire afterpulse clustering*: Even after spark cuts have been performed, there will still be some occasional afterpulsing noise. To prevent hot wires from biasing the center-of-gravity during the the clustering across wires in the next stage, I perform "artificial-deadtime" (AD) clustering separately on each wire: namely, all hits within `kePC*ClusterInterval` of an initial hit are clustered together into a single hit whose assigned time is the earliest time in the cluster.
4. *Clustering across wires*: Once the individual wires' signals have been "cleaned up" by the AD clustering, a clustering across wires is performed for each plane. All hits within $\pm \text{kePC*ClusterInterval}$ where $(\text{abs}(X_{\text{wire}_1} - X_{\text{wire}_2}) - 1) \leq \text{kePC*ClusterMaxGap}$ and where $\text{cluster_size} \leq \text{kePC*ClusterMaxSize}$ are grouped into a single hit. The cluster's assigned time is the earliest time in the cluster, and the position is assigned as the average of the contributing wires.
5. *Joining of anode and cathode planes*: (The following description is somewhat abbreviated. For a more detailed and complete explanation of how the three ePC planes' clusters are joined together, see my Run7 ePC analysis memo from February 2004.) First, the anodes are separately joined with the inner strips (AI) and with the outer strips (AO) based upon temporal coincidences, and (ϕ, z, t) information is assigned to each pairing. The physical offsets of the planes are tuned so that the AI and AO z-distributions are properly centered around zero. Next, the triple plane coincidence AIO is formed by joining together the AI and AO tables using common anode (ϕ, t) information as the bridge between the two cathode planes. Finally, a cut is made to select those three-plane coincidences where the AI z-position and the AO z-position are within 12 mm of one another. The results after this operation are presented in Figure 11.
6. *Exported tables*: Several banks of various detector combinations are exported downstream:
 - (a) anode only ("EP1A")
 - (b) inner strip only ("EP1I")
 - (c) outer strip only ("EP1O")
 - (d) anode/inner strip coincidence, 2 versions ("E1I1", "E1I2")
 - (e) anode/outer strip coincidence, 2 versions ("E1O1", "E1O2")
 - (f) inner/outer strip coincidence ("E1IO")
 - (g) anode/inner/outer strip coincidence, 2 versions ("E1V1", "E1V2")
 - (h) union of planes' sparks ("E1SP")
 - (i) intersection of planes' sparks (chamberwide) ("E1CS")

Some of these are used merely to calculate detector efficiencies, while others contribute to the final lifetime histograms. The anode/inner and anode/outer coincidence banks are later merged into ePC-cathode-OR tables, while the anode/inner/outer coincident events are designated ePC-cathode-AND.

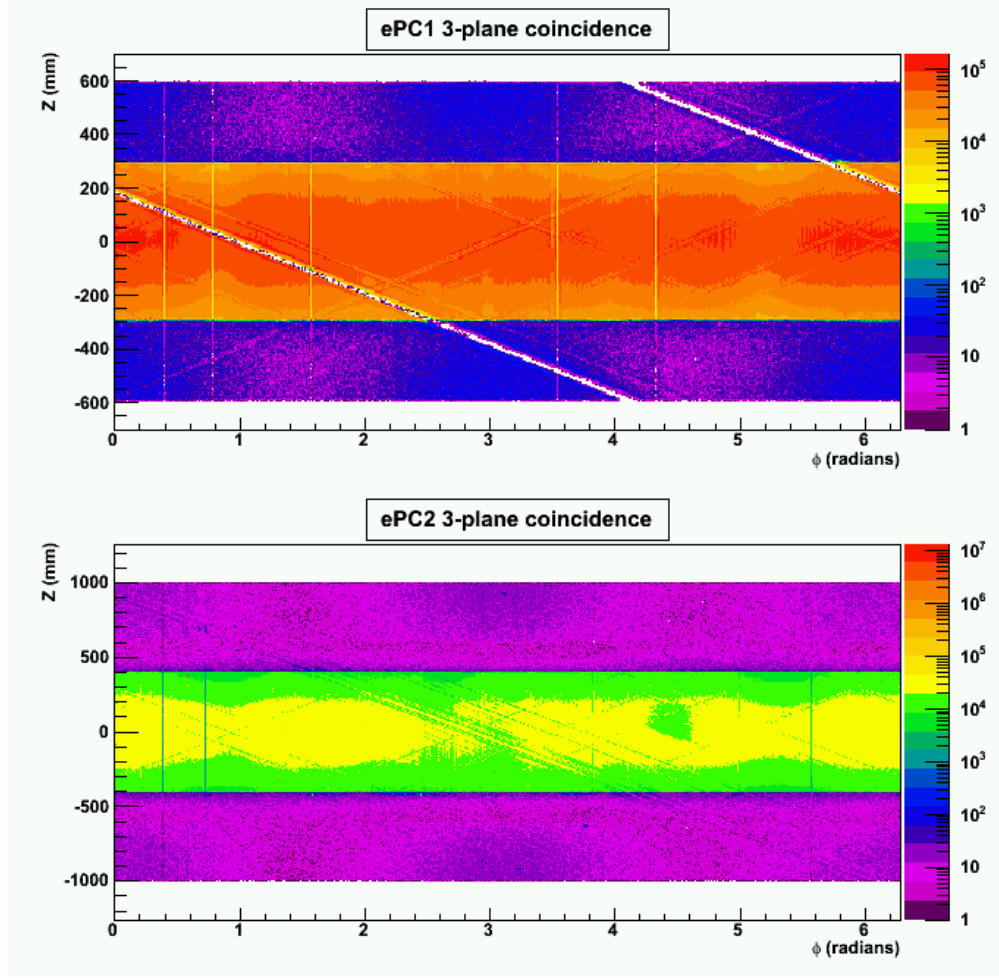


Figure 11: The *ePC1* and *ePC2* three-plane-coincident (ϕ, z) plots. I should note that the ϕ coordinates here are still in the “internal” coordinate space of the *ePCs*, in which anode 1 is defined to be around $\phi = 0$. Offsets will later be introduced to each to bring the *ePCs* into alignment with the global coordinate system of the experiment.

4.2 eSC

Below is the sequence of steps followed by the `MeSCAnalysisMQL.MQL` in processing the eSC data:

1. *Raw data selection*: The eSC hits are selected from the raw CAEN data in bank “HITS” (produced by `MCaenCompProcessRaw.cpp`). It should be noted that copies of the eSC signal were sent to both the CAENs and COMPs, and that those two sets of signals are checked upstream in the skimmer for discrepancies. If discrepancies are found, the block is cut.
2. *Spark cuts (SC)*: Spark events are identified from the raw data by clustering together all eSC hits within $\pm 1 \mu\text{s}$ of one another. If any of the resulting cluster sizes are greater than or equal to the `keSCSparkSizeCutoff` threshold in `ucb_common.h`, the earliest time in the cluster is flagged as a spark time, and all eSC hits within the next `kSparkCutDeadttime=50000 ns` are cut. All spark times are also saved for export.

3. *Per-phototube afterpulse clustering*: To eliminate afterpulsing noise, “artificial-deadtime” (AD) clustering is performed separately on each gondola phototube: namely, all hits within `keSCClusterInterval` of an initial hit are clustered together into a single hit whose assigned time is the earliest time in the cluster.
4. *Identification of inner gondola coincidences*: For each gondola, the inner upstream and inner downstream hits are joined together within a ± 100 ns time interval. Next, a timing offset is introduced for each gondola to center the time differences $T_{\text{upstream}} - T_{\text{downstream}}$ about zero (see Figure 12; the timing offset is applied to the upstream time, $T_{\text{upstream}} \rightarrow T'_{\text{upstream}}$). Finally, those coincidences within a 4σ interval around the central peak at zero are selected as good coincidences, and assigned the time (without any loss of generality) $T_{\text{inner}} = (T'_{\text{upstream}} + T_{\text{downstream}})/2$.
5. *Identification of outer gondola coincidences*: Identical procedure as for the inner gondolas.
6. *Join of inner/outer gondola coincidences*: For each gondola, the good inner and good outer coincidences within 100 ns of each other are joined together. Time offsets are introduced yet again to center the inner/outer timing for each gondola about $\Delta t = 0$ (Figure 13). Finally, the inner/outer coincidences centered around zero are selected to obtain the final set of good, fourfold gondola hits (Figure 14). These fourfold coincidences are assigned the time (without any loss of generality) $T_{\text{4fold}} = (T_{\text{inner}} + T'_{\text{outer}})/2$.
7. *Exported tables*: Two important banks are exported from the module: a table of good, fourfold gondola hits (“ESCG”), and a table of eSC sparks for later cuts (“ESSP”).

It should be noted that all of the eSC timing offsets are located in the MIDAS ODB, rather than in `ucb_common.h`, for historical reasons. I should also point out that all of the eSC timing alignments described above only synchronize the times for within an individual gondola—there may still be absolute timing offsets among the 16 gondolas when plotted against, say, the μ SC time. Fortunately these timing differences are comparatively small, only 5 ns at most, so in the fit regions of interest (starting around 100 ns) the gondolas’ lifetime histograms are all nicely exponential.

4.3 Coincidences

Just as with the muon detectors, I have adopted an “approval” philosophy towards using the electron detectors to identify good eSC times for inclusion in the lifetime histograms. I form all possible ePC1/ePC2/eSC coincidences (subject to certain reasonable criteria), apply cuts as desired (often in conjunction with the muon detector data), and then reduce the surviving electron coincidences down to a unique set of eSC times. In this way I avoid making any restrictive choices about how to associate different detector hits with one another, since, in my opinion, we simply lack sufficient information to properly make such judgements. Given the multiplicities which can arise from combining the numerous ePC planes, it does not make sense to me to make exclusive pairings of eSC times and ePC1/ePC2 tracks.

Here I will describe the algorithm used in MBAS module `MeDetCoincidenceCalcs.MQL` to construct ePC1/ePC2/eSC coincidences. The following procedures are performed in identical fashion for both the ePC-cathode-AND and ePC-cathode-OR data sets.

1. *Temporal join of ePC1 and ePC2 hits*: Temporally coincident ePC1 and ePC2 hits are joined together.

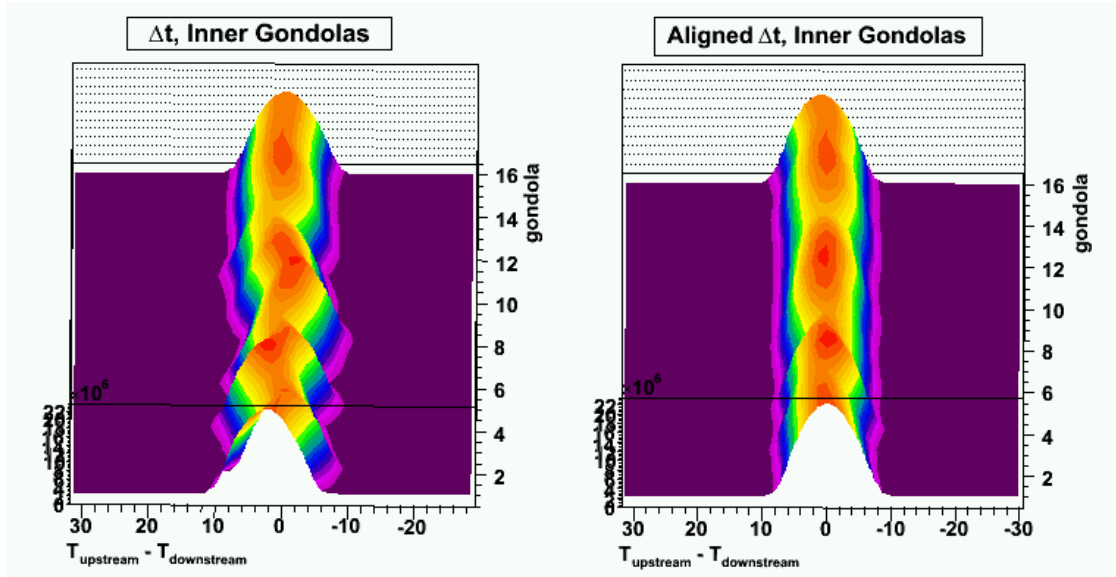


Figure 12: (Left) The initial per-gondola time differences between inner upstream and inner downstream hits. (Right) The per-gondola inner time differences after adjustments to eliminate timing offsets.

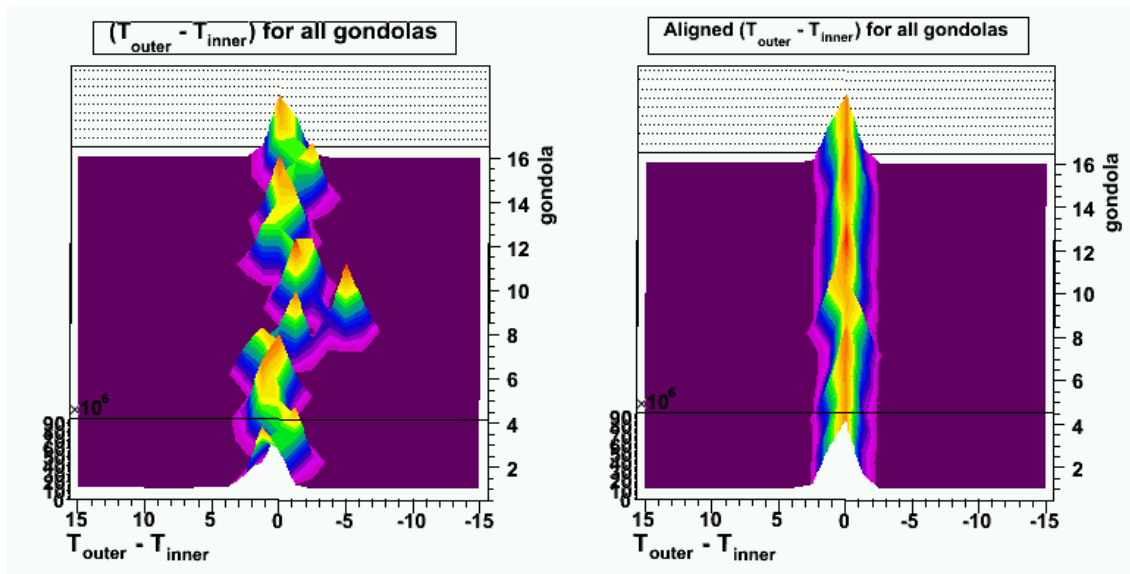


Figure 13: (Left) The initial per-gondola inner and outer time differences. (Right) The per-gondola inner and outer time differences after adjustments to eliminate timing offsets.

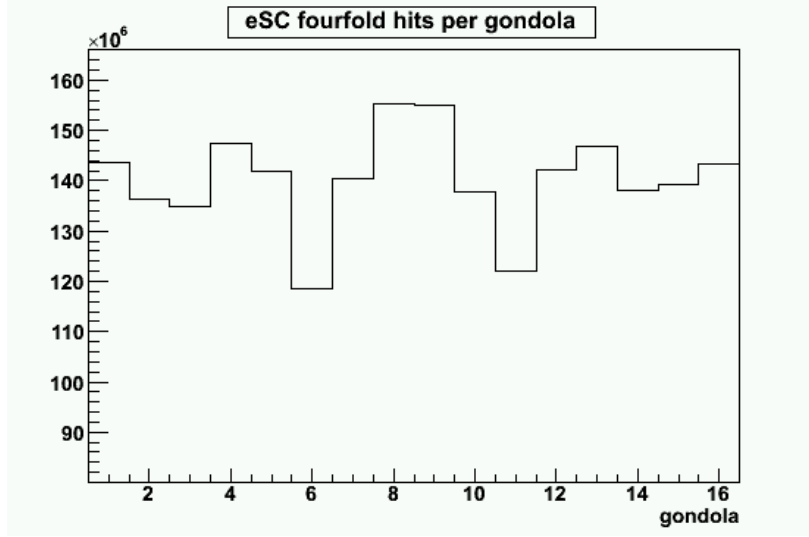


Figure 14: Fourfold eSC hits per gondola. Note the pronounced dip in counts for gondolas 6 and 11, which are screened to a significant extent from decay electrons by the TPC supports.

2. *Selection of ϕ -coincident ePC1/ePC2 pairs:* I select the subset of temporally coincident ePC1/ePC2 pairs where the two hits are within ± 0.35 radians of each other (the angle ϕ here refers to the cylindrical coordinate that runs around the circumference of the ePC drums). For any larger $\Delta\phi$, the ePC1/ePC2 track does not vector back to the pressure vessel.
3. *Temporal join of ePC1/ePC2 tracks with eSC hits:* The eSC hit must be temporally coincident with both the ePC1 and the ePC2 hit in the ePC1/ePC2 pair. At this stage I also apply a hard cut on the physical ePC1 and ePC2 z -limits; that is, I cut any ePC1/ePC2 pairs where an ePC z -value lies beyond the physical extent of that ePC's active region.
4. *Selection of ϕ -coincident ePC1/ePC2/eSC pairings:* I select the subset of temporally coincident ePC1/ePC2/eSC groupings where the ePC1/ePC2 vector roughly points to the eSC gondola. I say "roughly" because I allow for some slop: the ePC1/ePC2 track can point into the adjacent gondola up to, at most, 1/5 of its width. The final results are plotted in Figure 15. Note that I do not use any z -information when comparing the ePC and eSC data. In principle, timing differences between a gondola's upstream and downstream phototubes enables the calculation of the hit's z -position. However, the resolution is very coarse: the BC-404 eSC scintillator plastic has index $n = 1.58$, which means that we would naïvely expect light to take $\Delta t_{\max} = 90 \text{ cm}/(c/1.58) \sim 4.74 \text{ ns}$ to propagate along the entire length of a single gondola, while the CAEN timing resolution is only 1.25 ns (and nonuniform!). In reality the width Δt is somewhat wider, as can be see in Figure 12, but I decided that it was still not worthwhile to attempt to calculate and use eSC z information.
5. *Export ePC1/ePC2/eSC Ntuples:* Once the final ePC1/ePC2/eSC coincidences have been formed, I write the data out as ROOT Ntuples for processing in the subsequent Ntuple-analysis stage. In the most recent Ntuple-production pass I created four such Ntuples: cathode-AND and cathode-OR, at two artificial deadtime settings v1 and v2.

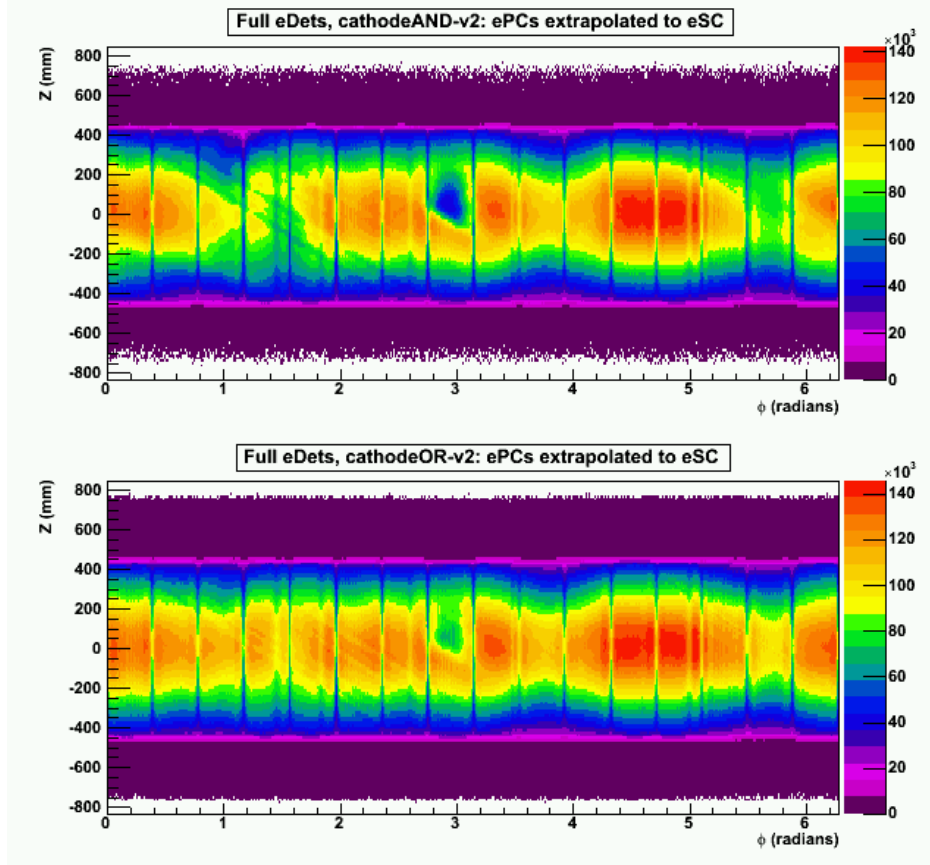


Figure 15: Plots of the full electron detector (ePC1/ePC2/eSC) coincidences. The upper plot is constructed from ePC cathode-AND data, while the lower plot is constructed from ePC cathode-OR data. The plotted (ϕ, z) points are the ePC1/ePC2 tracks extrapolated to the eSC detector’s “plane.” Note that the contrast is greater in the upper cathode-AND plot, which is constructed using more stringent coincidence criteria. Note too the pronounced inefficiency at roughly 2.9 radians, which is due to a dead spot in ePC2. The vertical lines arise from coverage inefficiencies from the small gaps between the eSC gondolas.

At this point the electron detector data is ready to be joined with the muon detector data.

Before moving on, it is worth noting that we can learn something about the individual electron detector efficiencies by looking at the ratios of (ϕ, z) histograms which involve different combinations of detectors (see Figures 16, 17, and 18). I should point out that the numerical efficiencies in the plots are not reliable, since they are consistently higher than unity (I have restricted the plots’ ranges from 0–1). The reason is most likely because of combinatoric enhancement from multiplicities in the wire chamber hits. The actual efficiencies are closer to 92.5% for ePC1, 88.2% for ePC2, and 95% for the eSC. Thus, the ePC efficiency plots are best used as a way to view efficiency variations across a detector.

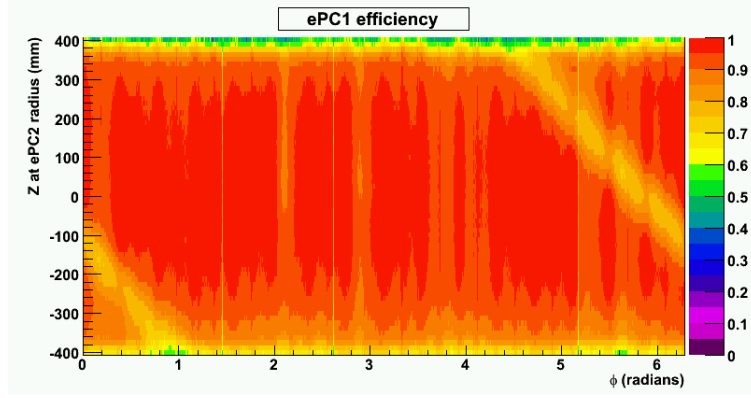


Figure 16: The $ePC1$ efficiency in (ϕ, z) , calculated via the histogram division $(ePC1+ePC2+eSC)/(ePC2+eSC)$. Note that the $ePC1$ efficiency here is by necessity plotted at the $ePC2$ radius.

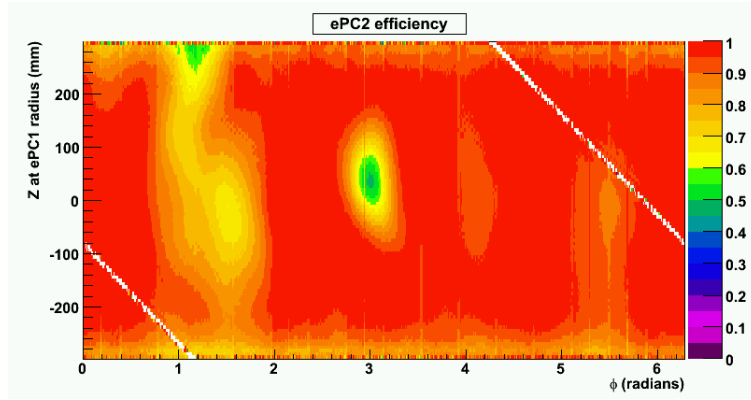


Figure 17: The $ePC2$ efficiency in (ϕ, z) , calculated via the histogram division $(ePC1+ePC2+eSC)/(ePC1+eSC)$. Note that the $ePC2$ efficiency here is by necessity plotted at the $ePC1$ radius. Note too the presence of inefficiency “deadspots” in $ePC2$.

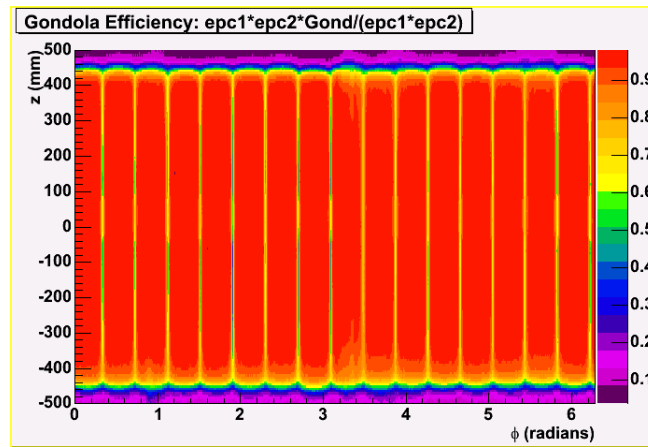


Figure 18: The eSC efficiency in (ϕ, z) , calculated via the histogram division $(ePC1+ePC2+eSC)/(ePC1+ePC2)$. (This image courtesy of Steve Clayton.)

5 Special studies

The following studies play some role—whether directly or indirectly—in constructing the final lifetime histograms.

5.1 Spark cuts

The Berkeley analysis recognizes two categories of detector sparks: local and global. Local sparks are events where a larger-than-normal number of contiguous wires simultaneously in a single detector. (In the case of μ PC1 and the ePCs, this applies to the individual detector planes.) Global sparks are occasions where “local” sparks appear in nearly every MuCap detector at the same time. The exact reasons for such occurrences remain unknown.

I have found that an effective method for identifying global sparks is to look for instances when all six ePC detector planes exhibit simultaneous local sparking (see Figures 19 and 20). If this occurs, the entire

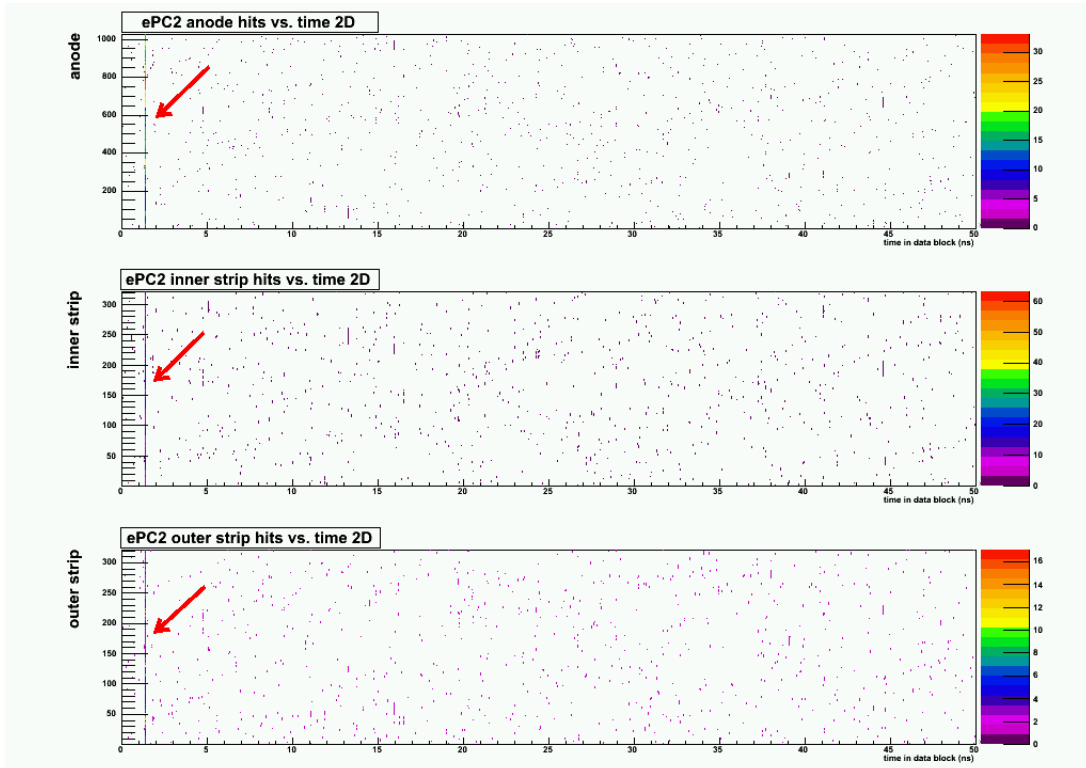


Figure 19: Example of a chamberwide ePC2 spark. The plots show ePC2 anode, inner cathode, and outer cathode plane hits over the course of a single data block. The spark, indicated by the red arrows, is easily recognized by the high multiplicity of simultaneous hits across all ePC2 wires.

data block is discarded by the MGlobalSparkCut module. However, the global spark cut alone is not sufficient to eliminate all sparking-related effects, since most of the detectors still suffer from individual, local sparks. For example, the ePCs often spark in preamp card groups (Figure 21). To protect against these frequent events, I discard all data within a $[-5 \mu\text{s}, 50 \mu\text{s}]$ time interval around the local spark. After

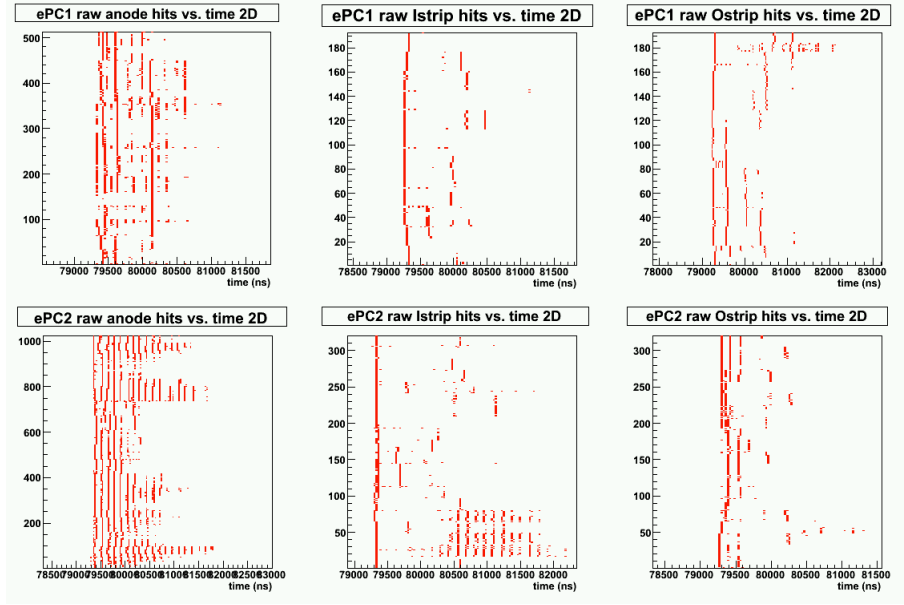


Figure 20: This plot shows a spark in all six ePC planes, which I use as a means to identify global sparks. These plots are similar to the ones presented in Figure 19, but with a zoomed-in time scale.

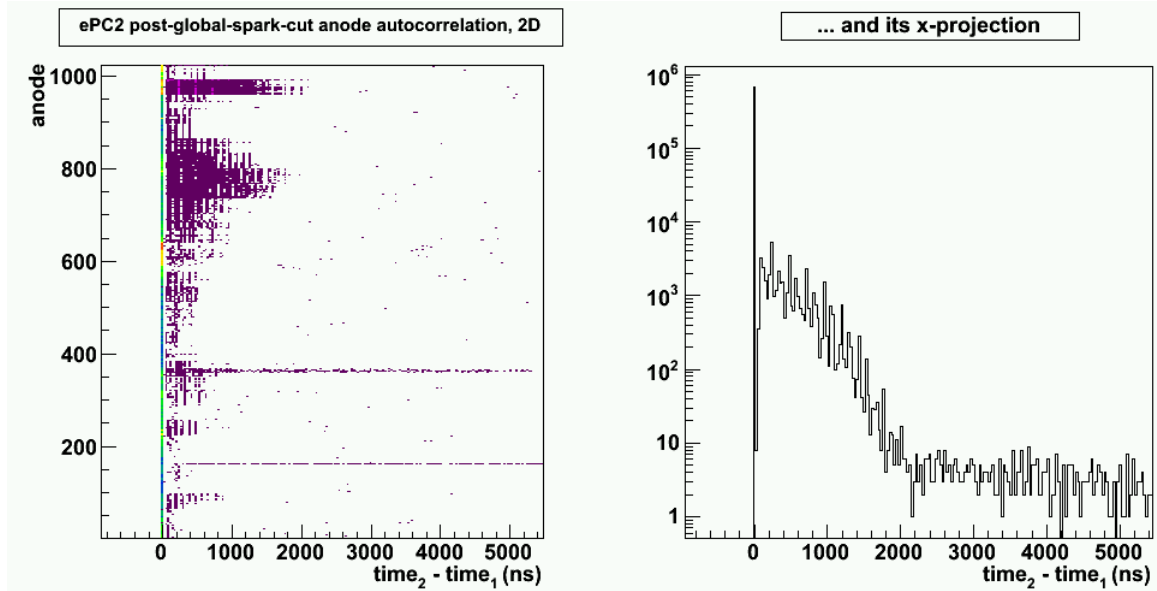


Figure 21: Autocorrelation plots for the ePC2 anode plane, revealing the existence of local sparking **after** global spark cuts have been performed . Sparking in the ePC2 preamp card groups is clearly evident in the 2-dimensional plot. (The 1-d plot is simply the x -projection of the 2-d plot.)

that, artificial deadtime clustering is used to suppress any residual hot wires or channels. This last step is important for the μ PC1 and ePCs, as it prevents hot wires from biasing the spatial clustering’s center-of-gravity.

5.2 CAENs & Electronics

We have struggled with the CAEN v767 TDC modules and their deficiencies for years. A complete recounting of our history of CAEN problems is beyond the scope of this paper, but, through a combination of preventive action and intelligent cuts in the analysis (with some help from our MuLan colleagues), we believe that we have minimized CAEN-related artifacts as much as possible. In this section I will describe how we did so.

During Run8 we employed three CAENs. To avoid the possibility of cross-talk, two of the CAENs were dedicated to muon detector signals, and one was dedicated to electron detector signals. Furthermore, to protect against CAEN misbehavior—or to provide us with the necessary tools to identify it later in the analysis—we took the following steps:

- We used a 25 MHz signal as the CAEN external clock input. Previous experience had demonstrated that faster clock signals, though declared acceptable in the CAEN documentation, produced erratic behavior.
- We sent 2.5 kHz “rollover” clock pulses into each CAEN, to provide us with the means to later check the synchronization among the CAEN TDCs in the analysis.
- We sent separate copies of the crucial μ SC signal into different CAEN modules so that they could later be compared in the analysis.
- We recorded DAQ error flags, and the CAEN modules’ own error flags.

At various stages in the analysis, we perform several kinds of consistency and error checks on the CAEN data:

- When processing the raw CAEN data in module `McaenCompProcessRaw.cpp`, we check for both DAQ-generated and CAEN-module-generated error datums. If an error datum of either kind is encountered, the data block is rejected.
- The `McaenCompProcessRaw.cpp` module also counts the number of trailing edges per block, and records their times in bank “CAET,” in the form (time, caen#).¹ If the number of trailing edges from a single CAEN within a single block exceeds 100, the entire block is cut. Otherwise, it is up to Steve and me to decide how to handle the trailing edge information in our individual analyses. I chose to treat them in exactly the same manner as local sparks cut the surrounding data, in case they are responsible for a temporary suppression in the acquisition of “real” data. In fact, I merge the CAEN

¹At the April 27-30, 2005 UIUC MuLan meeting, MuCap was informed of the pernicious effects of CAEN trailing edge hits. We had been aware that trailing edges still occasionally appeared even when we operated the CAENs in leading-edge-only mode, but the Boston group provided some further information about the nature of the problem and its possible effects. In particular, Kevin Lynch found that the CAEN trailing edges typically appear in bursts of simultaneous, multichannel, contiguous hit clusters. Kevin surmised that the problem may be related to a fault in the clock distribution network such that when several clock bits change at once, trailing edge hits are generated; the more bits involved in the flop, the bigger the trailing edge burst. The primary concern is that these large bursts of trailing edges could lead to loss of real leading edge hit data. A limited inspection of some MuCap Run8 data files essentially confirmed the behavior discovered by MuLan.

trailing edges in with the spark times, so they are all dealt with by the same local cut algorithm in module `MCoincidenceCalcs.MQL`. (Trailing edges from muon CAENs 0 and 2 are lumped in with muDetector sparks, and electron CAEN 1 is merged with eDetector sparks).

- If any discrepancies are found among the CAENs' 2.5 kHz rollover clock pulses, the data block is rejected. (Module `MRolloverCheckMQL` fills comparison histograms and the `MRolloverCheckC` module inspects them, cutting the block if necessary.)
- Comparison of the various μ SC signal copies provides further protection against TDC desynchronization. If there are too many mismatches, the data block is cut.
- Comparison of the eSC CAEN and compressor (COMP) copies also provides some protection against CAEN losses. If there are too many discrepancies between the two sets of data, the block is cut.

From the unskimmed production pass in spring 2005, I estimated that CAEN errors were responsible for the rejection of $\sim 1\%$ of the total data. All data integrity, error, and spark cuts add up to a 2% cut of the total data.

The only remaining CAEN issue at this point is the possible interplay between the DAQ clock and the 50 MHz cyclotron beam structure: beating effects between the two will most likely contribute an entry in our final systematics error table. Fred has worked on Monte Carlo simulations of this effect, and may have something to say on the subject at our October 6–8, 2006, meeting. It should be noted that an accurate simulation of the situation requires knowledge of the DAQ clock frequency, so the error assessment may have to be repeated after the unblinding.

5.3 Identification of muon scatters

On rare occasions, a muon enters the TPC volume and scatters hard off of a proton. The recoiling proton usually deposits enough energy in the detector to trigger the EH threshold, and as a result the scatter event can be misinterpreted as a muon stop. The identification of muon scatters is complicated by the fact that the scattered muon often does not leave behind a robust ionizing track. It is important for us to get a handle on this behavior because muon scatters into the surrounding high- Z detector materials can increase the fitted decay rate, pulling it away from the value for stops in pure hydrogen.

After grappling with the problem for some time, I eventually developed two methods for identifying muon scatter events, which I refer to as “TLS” and “MWPC”:

- TLS is an acronym for “Total-Least-Squares.” In this method I inspect the region downstream of an ostensible EH Bragg stop and look for evidence of a track of EL pixels leading away from the EH cluster, which is an indication of the departure of a scattered muon. If I find more than four EL pixels, I perform a total-least-squares fit to the points and check the results. A group of EL pixels is considered to be a scattered muon's track if (1) the spread of points around the best-fit line is < 2.2 mm, (2) the best-fit line points back towards the EH cluster (impact < 8 mm), (3) the points cover a distance of at least 18 mm, and (4) the points are sufficiently spread out along the best-fit line.
- MWPC is an acronym for the “MultiWire Proportional Chamber” region of the TPC. When an ionizing particle passes through the bottom of the TPC, it usually triggers the EH threshold. Thus, if a μ SC hit can be associated with an apparently good TPC stop in the fiducial drift interval, but there is *also* an EH cluster in the time interval [0,850] ns after the μ SC hit, then there is a good chance that the muon scattered through the bottom of the TPC.

In Figure 22 I present a muon scatter event which would be recognized by both my TLS and MWPC scatter criteria.

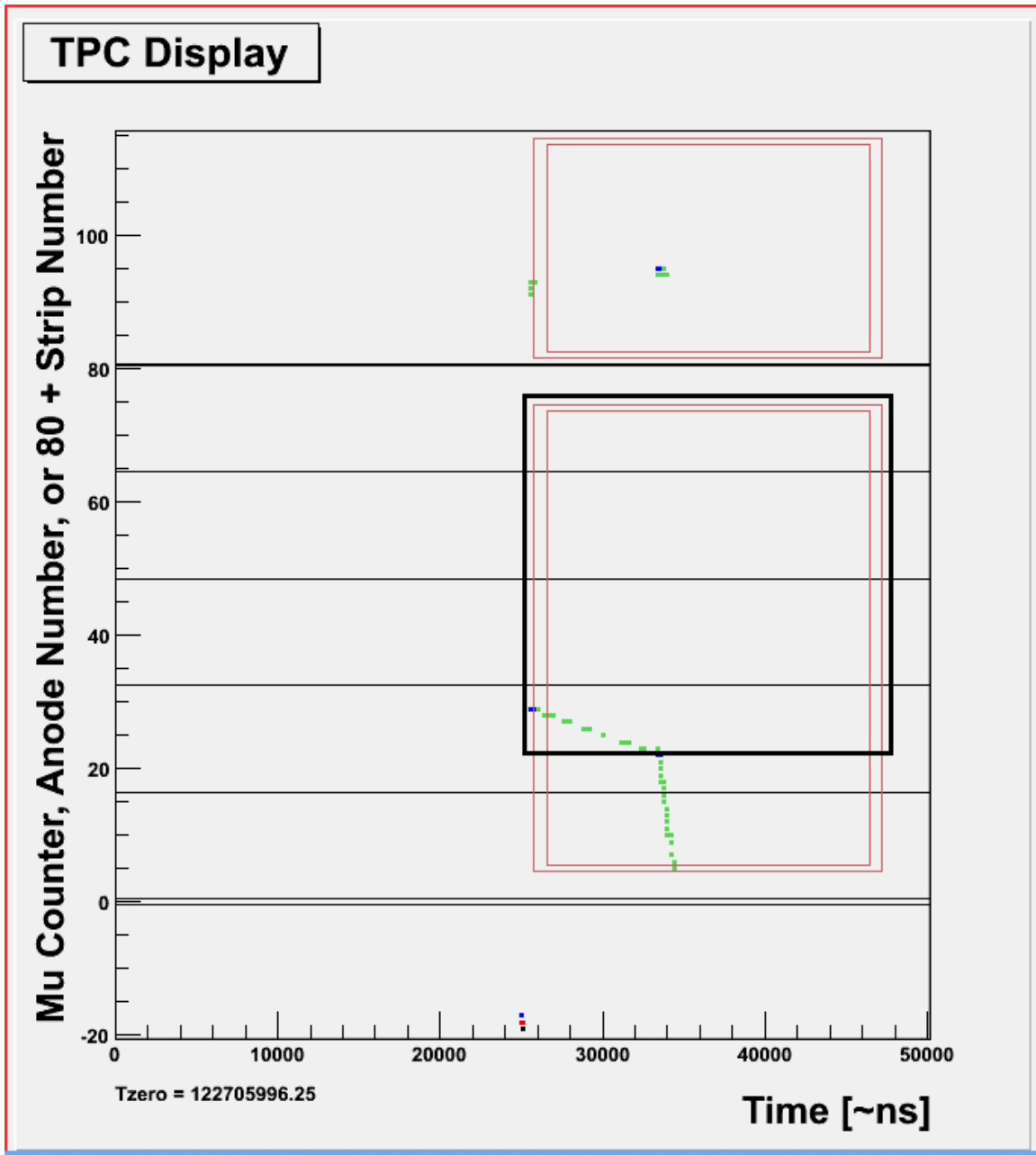


Figure 22: An example of a probable muon scatter, taken from Steve Clayton's event display. In this case, the muon appears to have scattered downwards through the bottom of the TPC and thus would be identified by both my TLS and my MWPC criteria.

My analysis of the clean fill data has consistently yielded muon scatter fractions around 1.2×10^{-4} . However, *a priori* there's no way of knowing how accurate this result really is, because I don't know the

efficiency of my scatter finder—am I catching 100% of all scatters, or 1%? To answer that question, we need some way of estimating the fraction of muons which scatter off of a proton and deposit enough energy to mimic a muon stop. A useful resource in this regard is the SRIM (“Scattering and Range In Materials”) simulation software. It allows one to simulate the trajectories of muons in hydrogen gas under MuCap-like conditions, and obtain information about muon-proton scatters.

Now, a “scatter” is an event where the muon collides with a hydrogen atom within the TPC’s fiducial volume, but then ultimately stops outside of the TPC’s active region. The energy of the recoiling atom is relevant, because scatter events are only problematic if the recoil energy is large enough to trigger the EH threshold and thereby mimic a muon stop. Figure 23 shows the SRIM “escape scatter” fractions as a function of the recoil energy. At this point we need to know which “recoil energy lower cutoff” corresponds to the

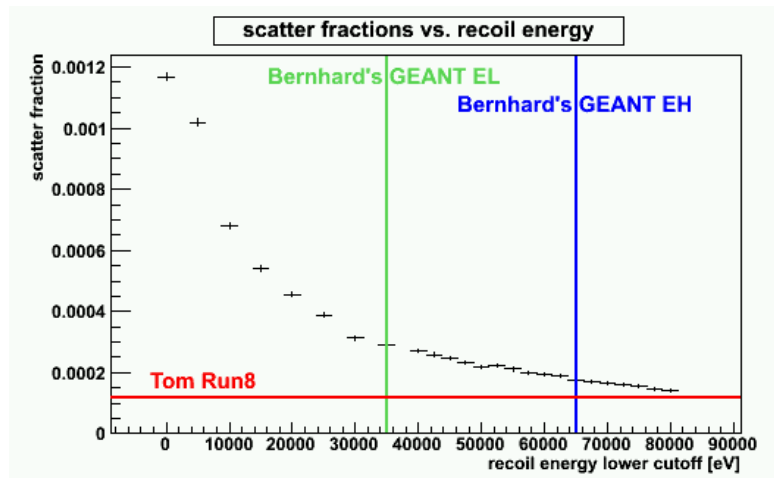


Figure 23: The SRIM escape scatter fractions as a function of the recoil energy cutoff. For reference I have drawn the EL and EH threshold values as determined by Bernhard Lauss using GEANT MC simulations, as well as the scatter fraction I obtained in my analysis of the Prod-50 clean fill.

actual Run8 EH threshold. Bernhard Lauss’ GEANT settings (EL=35 keV, EH=65 keV, EVH=260 keV) were tuned so that his MC muon tracks resembled the Run8 data in the event display. However, we need a better way to connect the SRIM results with the Run8 data, because although the GEANT settings provide a nice rule-of-thumb, we can’t use one MC to calibrate another. So how can we find where the Run8 EH threshold sits on the SRIM recoil energy scale?

Fortunately, there is another observable which can help us to establish a SRIM/Run8 correspondence: the scatter events’ angular distribution in theta, which is a function of the recoil energy cutoff (Figure 24). By finding the SRIM scatter-angle distribution that most closely resembles the Run8 scatter-angle distribution (i.e. which has a similar shape and peak-to-peak spread) we can estimate which SRIM recoil energy cutoff corresponds to the Run8 EH threshold setting, as shown in Figure 25. Unfortunately, the peak-to-peak spread does not steadily increase with the recoil energy cutoff. Nevertheless, from the Figure 25 plots I think it is safe to say that the Run8 EH threshold is between the SRIM energies 50–80 keV; in fact, one could argue that the SRIM 70 keV cutoff is most appropriate. The scatter fractions for SRIM energies greater than 50 keV are shown in Figure 26. It thus appears that Bernhard’s EH setting at 65 keV is in fact a reasonable point for the SRIM recoil energy cutoff, in which case my scatter-finding algorithm is roughly $(1.21 \times 10^{-4} / 1.76 \times 10^{-4}) = 69\%$ efficient.

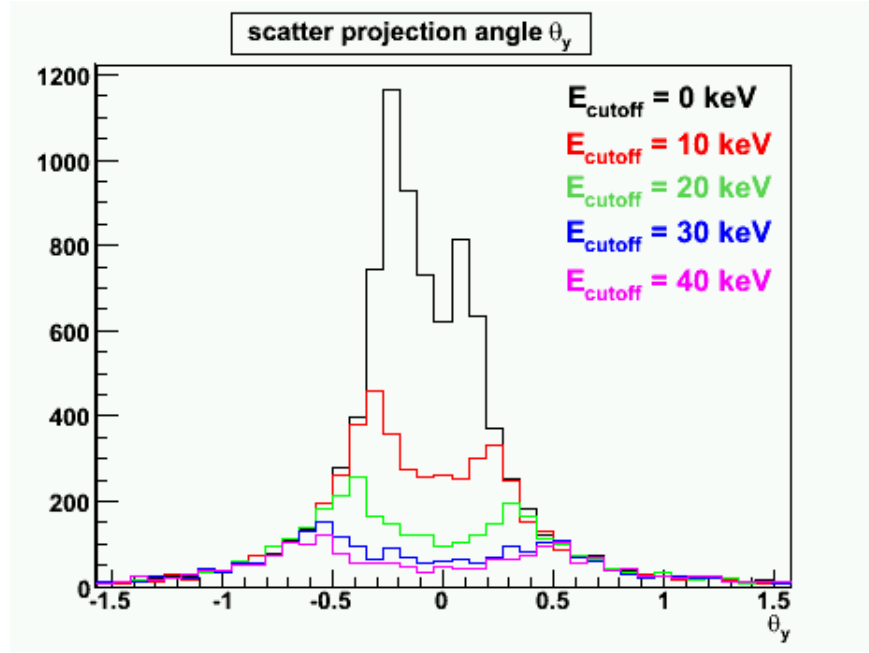


Figure 24: The angular distribution of SRIM muon scatters in hydrogen vs. the recoil cutoff energy. The scattering angle θ is calculated with respect to the z -axis of the experiment, and the left peak is larger than the right peak because the SRIM muon beam was pointed downwards, just like the Run8 beam.

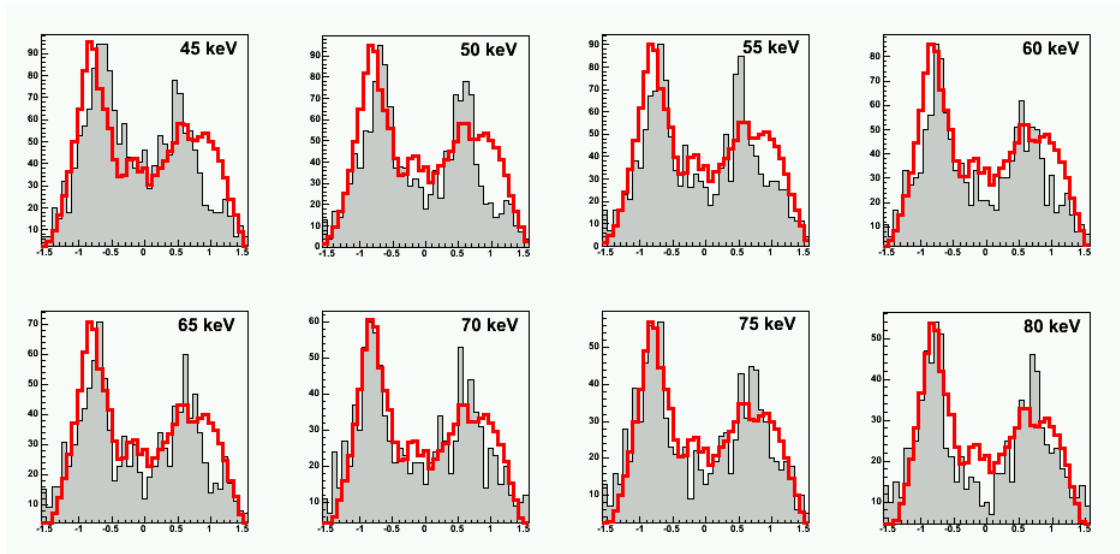


Figure 25: The Run8 Prod-50 clean fill TLS scattering distribution in θ (red) is laid over SRIM angular scatter distributions involving different recoil energy cutoffs. This is a way to determine which SRIM recoil energy cutoff is closest to the Run8 EH threshold which led to the observed scattering distribution.

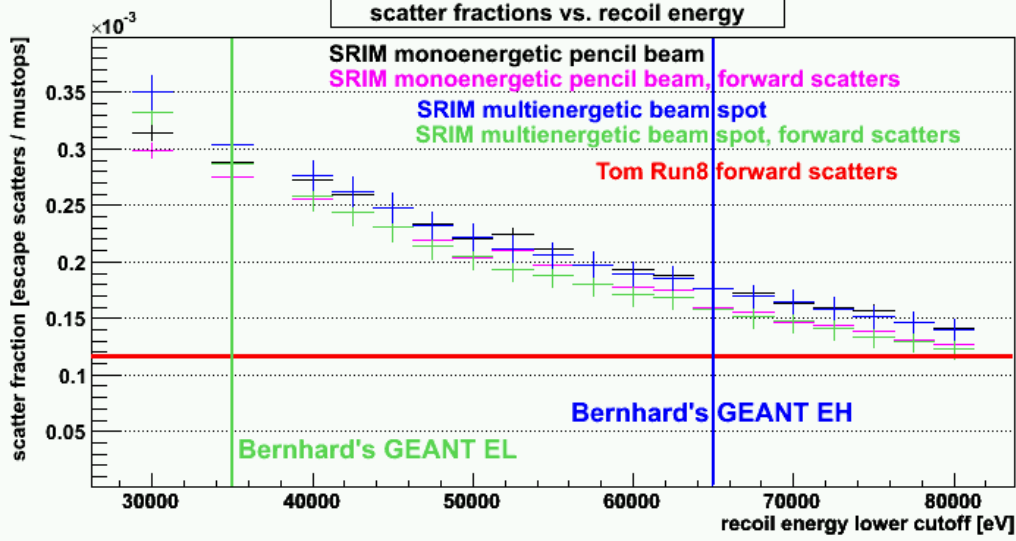


Figure 26: A close-up view of Figure 23, for better comparison of the muon scatter fraction predicted by SRIM vs. what I actually find with my analysis of the Prod-50 data.

5.4 Identification of cosmic

To identify throughgoing (muon) cosmic, I look for simultaneous electron detector track pairs that lie along roughly the same line and are oppositely directed. Here are the steps in my cosmic identification algorithm:

1. I form the set of temporally coincident ePC1/ePC2/eSC track pairs ($|\Delta t| \leq 15$ ns).
2. For each pair of tracks, I enforce an antiparallel condition where the angle between the two vectors (calculated using the law of cosines) must be $\geq 3\pi/4$. I also enforce an identical antiparallel condition on the vectors' projection onto the electron detectors' cylindrical (r, ϕ) plane. (The reason for this seemingly redundant cut is to prevent against a small number of pairs which lie along roughly the same anode and thus ultimately yield unphysical cosmic vectors.)
3. I require that the distance of closest approach between the two tracks' extended lines must be ≤ 26 mm, a number based upon the distribution of histogrammed impacts.

If a pair of tracks passes through all of the preceding cuts, they are considered to be part of a single cosmic event, and the (ϕ, θ) coordinates describing the cosmic vector are calculated from the two tracks' ePC2 points. The resulting angular distribution of cosmic is plotted in Figure 27; its basic features and statistics conform with expectations.

I would like to point out one cosmic-related curiosity for the sake of completeness. The faint vertical bands visible in the ePC backgrounds in Figure 11 at $\phi_{\text{relative}} = 0$ and π are likely due to the “unphysical” reconstructions of (near) vertical throughgoing cosmic—that is, the cathode hit from one cosmic strike coincident with the anode hit from the cosmic's other strike. The edge of the unphysical band is especially visible in the ePC2 (ϕ, z) plot. Both ePC1 and ePC2 were intelligently designed such that the unphysical band does not overlap in z with the physical band. Thus, as long as z cuts are performed on the ePC data, these unphysical hits should not produce any contributions.

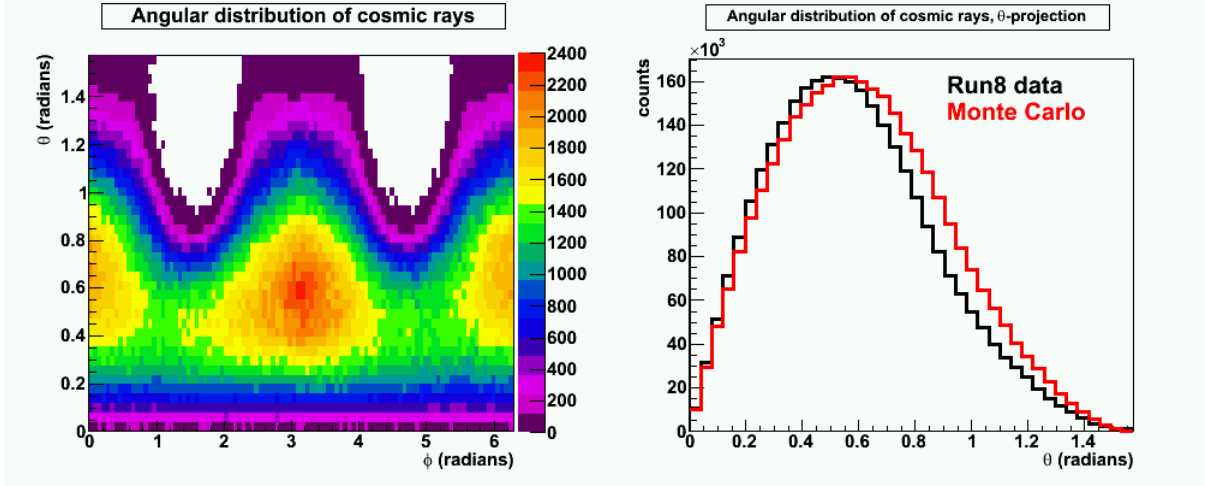


Figure 27: (Left) The angular distribution of identified cosmic rays, in spherical coordinates. The azimuthal angle $\phi = 0$ points along the $+x$ -axis, while the zenithal angle $\theta = 0$ points upward along the $+y$ -axis (see Figure 55 in Appendix E.2 for a diagram of our conventional definitions for the MuCap experimental coordinate system). The two semicircular holes in the (ϕ, θ) plot correspond to the openings at the ends of the eSC cylinder, which is the limiting detector when it comes to solid angle coverage. The asymmetry in the size of the peaks at $\phi = 0$ and $\phi = \pi$ can probably be attributed to spatial variations in the detector efficiencies. (Right) The θ -projection of the cosmic distribution. Notice that the Run8 θ -distribution data has roughly the same shape as predicted by my fast Monte Carlo software, which performs a purely geometrical simulation of the effects from cylindrical detectors and a $\cos^{1.8} \theta$ cosmic distribution. There are some small discrepancies here between reality and the MC, but they can probably be attributed to the aforementioned electron detector efficiency variations, as well as shielding asymmetries (e.g. in the arrangement of the $\pi E3$ area’s concrete, and in the site’s surrounding geographical features).

5.5 Impact parameter

In MuCap analysis parlance, the term “impact parameter” has generally been used to refer to the closest distance between a muon’s stopping position and the line which describes the outgoing decay electron’s trajectory. Cuts on the impact parameter play an important role in the Run8 analysis: First, they dramatically lower the background in the decay spectrum by eliminating a large fraction of uncorrelated muon/electron pairs, and thus improve our decay signal; Second, impact-parameter-cut-related studies have enabled us to calculate the deuterium concentration ratio between the clean fill and the natural hydrogen calibration fill, confirming the result from external measurements of hydrogen samples.

To calculate an impact parameter, we must first establish the muon’s stopping position in (x, y, z) , as well as the outgoing decay electron’s three-dimensional trajectory. Determining the x and z coordinates of a muon stop is relatively straightforward: the z -coordinate is calculated from the last anode in the TPC EH cluster that represents the Bragg peak, and the x -coordinate is calculated from the average of the EL cathode hits that are coincident with the EH anode cluster. Determining the muon stop’s y -position is far more complicated, since the TPC convolves space and time in the y -dimension. To reconstruct the y stopping position, we need another counter to provide the muon’s arrival time—in our case, the μ SC fills this role, and enables us to calculate the drifttime $\Delta t_{\text{drift}} = t_{\text{TPC}} - t_{\mu\text{SC}}$ (I should note that t_{TPC} is selected as the earliest time at the end of the muon stop). Converting this drifttime into an accurate μ stop y -position is tricky, because the TPC is divided into two sections: the MWPC amplification region and

the sensitive volume (Figure 28). Based upon Peter Kammel’s GARFIELD studies of TPC drift prop-

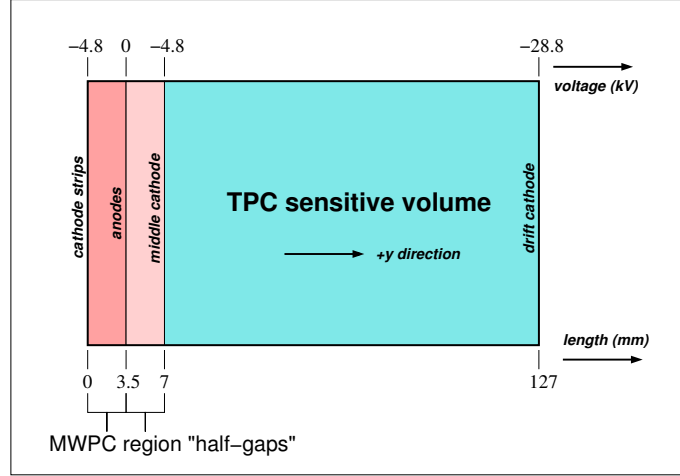


Figure 28: Schematic of the TPC detector’s different regions. (The chamber is turned on its side).

erties (MuCap Note #21), I believe that the large reflection peak near 475 ns in the TPC drift distributions in Figure 10 corresponds to the edges of the TPC’s MWPC region. Once I take into account the time smearing due to the 200 ns TDC400 resolution of the TPC data, I determined that the two TPC regions correspond to the following drift intervals: MWPC=[0,kTPCMWPCoffset]=[0,375] ns; sensitive volume (SV)=[kTPCMWPCoffset,kTPCMaxDriftTime]=[375,22420] ns. The formula for converting drifttime into y -position is then

$$y = \left[\frac{\Delta t_{\text{drift}} - 375 \text{ ns}}{22420 \text{ ns} - 375 \text{ ns}} \right] \times 120 \text{ mm} . \quad (1)$$

The stability of the TPC drift distribution during Run8 is a possible concern. After all, the TPC’s amplification and drift voltages were reset or changed several times during Run8—sometimes incorrectly—and this could have affected the drifttime. We always tried to preserve a sensitive volume voltage difference of 24 kV so that the drifttime interval remained constant, but it was not immediately clear if we were successful in this regard. (For example, we think that there was a period of at least one day, Nov. 4, when the drift voltage was accidentally set to 29.8 kV instead of the customary 28.8 kV). Fortunately, inspection of the drift distribution’s width over time revealed that the drift interval was stable to within one TDC400 200 ns clocktick for the duration of Run8, which is better than 1%, or 1.1 mm.

The electron vector is comparatively easy to compute, being formed simply from the line between an ePC1 point and an ePC2 point. The impact parameter b can then be calculated using standard geometrical formulas.

In Figure 29 I have plotted the x , y , and z components of the time-integrated Run8 clean fill impact parameter, as well as the impact itself, $b = \sqrt{x^2 + y^2 + z^2}$. The x and y distributions are narrow because the ePC anodes are closely spaced and yield good resolution; the z distribution is somewhat broader due to the wider spacing of the ePC cathodes. In order to center the y and z impact distributions, I had to introduce small offsets to account for the displacement of the TPC relative to the coordinate origin of the electron detectors. The angular ϕ offsets for ePC1 and ePC2 (relative to the eSC) were also tuned until the location of the impact distribution’s peak was minimized.

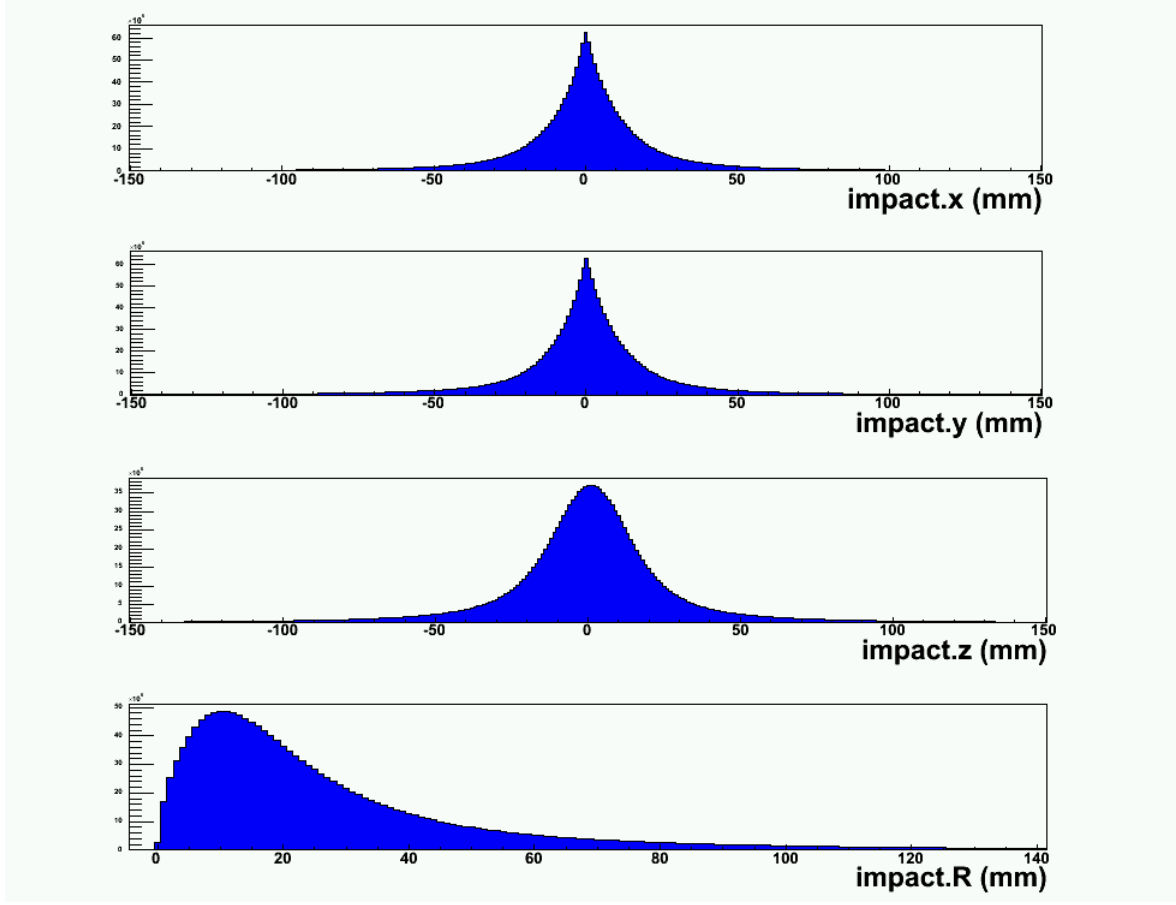


Figure 29: Impact parameter histograms from the Run8 Prod-50 clean fill. The x , y , and z components are plotted, as well as the total impact parameter $b = \sqrt{x^2 + y^2 + z^2}$.

Monte Carlo studies have confirmed that the impact parameter resolution is limited primarily by scattering from the pressure vessel. That is, most of the broadening in the bottom impact plot in Figure 29 is due to electron scattering from the vessel walls. This creates an extended tail, which has subtle implications for any impact cuts that are made: namely, the time-dependent process of muon diffusion couples with scattering in such a way that performing an impact cut—even at a large b value, such as our customary 120 mm—will increase the observed decay rate. The distribution of these rate offsets as a function of the impact parameter cut is presented in Figure 30. Ultimately we will have to perform a correction to the fitted rate to account for this impact-parameter-cut-generated offset, which is around 3 Hz for our standard “disk” impact cut $b \leq 120$ mm.

5.6 Magnetic field effects on the impact parameter

The μ^+ SR saddle coil magnet was left running for most of Run8, during both μ^+ and μ^- data taking, for the sake of maintaining stable experimental conditions. The magnet suffuses the interior of the pressure vessel with a (nearly) uniform ~ 50 gauss magnetic field, designed to induce a controlled precession in positive muons (the presence of a magnetic field is irrelevant for negative muons). Unfortunately, the magnetic field

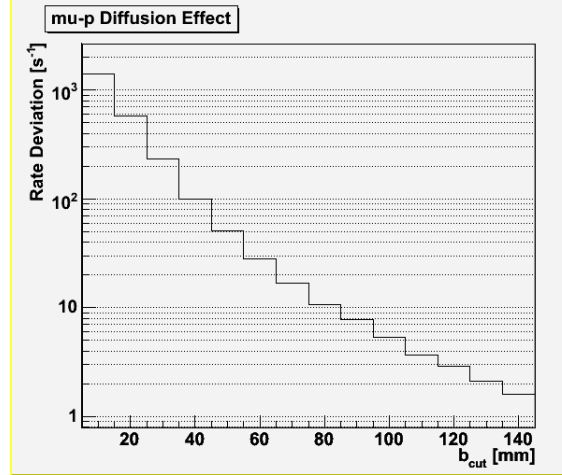


Figure 30: Decay rate deviations vs. impact cut, due to the coupling between μp diffusion and electron scattering. This plot was created by Steve Clayton, who simulated the effect using the Run8 impact distribution in conjunction with a simple model of thermal μp diffusion with constant $k = 0.4875 \text{ mm}/\sqrt{\mu\text{s}}$.

has an undesirable side effect: it deflects the trajectories of outgoing decay electrons from both muon species to some degree. If the deflections are sufficiently large, they could compromise the integrity of the impact parameter and any related cuts. It is therefore important to check how large these deflections are for MuCap.

First, let us survey the underlying physics. When a charged particle moves through a uniform magnetic field, it follows a helical trajectory: the particle moves with constant speed parallel to the field while executing SHM in the plane perpendicular to the field. The radius of a decay electron's circular orbit is given by

$$r = \frac{p_{\perp}}{(0.3B)}, \quad (2)$$

where $p_{\perp}$ is the momentum component perpendicular to B in GeV/c, B is in Tesla, and r is in meters. In Run8, the MuCap magnetic field was approximately 50 Gauss=0.005 Tesla. For a 30 MeV decay electron moving (in the worst case) perpendicular to the B-field, Formula 2 gives a radius of $r=20 \text{ m}$, so naively one would not expect the magnetic field deflections to be significant over most of the Michel spectrum.

To confirm this, I implemented magnetic-field-induced helical electron trajectories in the Monte Carlo framework I had originally designed for studying diffusion and scattering effects. The software recreates the muon stopping distribution in the TPC, then emits a decay electron in a random direction, with an energy sampled from the Michel spectrum. I next iteratively step through the electron's helical trajectory inside the pressure vessel's 50 Gauss field. When the electron trajectory intercepts the pressure vessel wall, I calculate the tangential velocity vector at that point, and from there the corresponding impact parameter.

The Monte Carlo impact parameter distribution arising solely from magnetic field deflections is plotted in Figure 31 alongside the actual Run8 impact distribution. The magnetic field deflections are clearly a small perturbation atop the pressure-vessel-scattering-dominated impact distribution, so we need not worry about magnetic field effects.

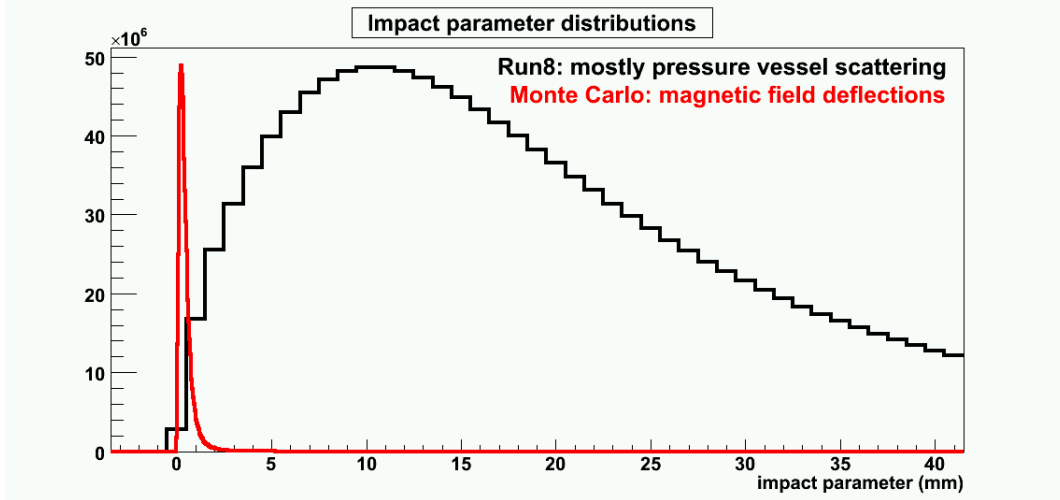


Figure 31: *Impact parameter distributions from my Monte Carlo simulation of magnetic field deflections, and from the actual Run8 data, which is dominated by scattering from the pressure vessel.*

5.7 Identification of high- Z capture events

When a μp atom encounters a high- Z impurity in the hydrogen gas, the muon will preferentially and irreversibly transfer to the heavier atom due to its stronger nuclear binding energy. This behavior is a concern for MuCap, because high- Z atoms have high nuclear capture rates $\Lambda_Z \gg \Lambda_S$, and their presence increases the effective muon disappearance rate. It is therefore important for us to be able to determine the impurity levels in our hydrogen gas. During Run8, we had two means of doing so: (1) external measurements of gas samples, and (2) *in situ* monitoring of capture events in the TPC data.

To identify muon capture events in the TPC data like the one shown in Figure 32, I wrote an analysis module (MTPCImpurityCaptureSearch) which inspects the pixels in the time interval following a muon stop for evidence of a capture. Namely, the module searches rightwards for EH or EVH TDC400 pixels within the 1–10 μs time interval after the stop, inside a ± 3 anode interval around the stopping anode. As a control, the capture search is also performed in the opposite direction (leftwards), in the time preceding the muon stop. The resulting distribution of capture yields over time during Run8 is shown in Figure 33. These capture yields—and the corresponding gas measurements—will later play a crucial role in performing corrections to the fitted decay rate due to effects from high- Z impurities.

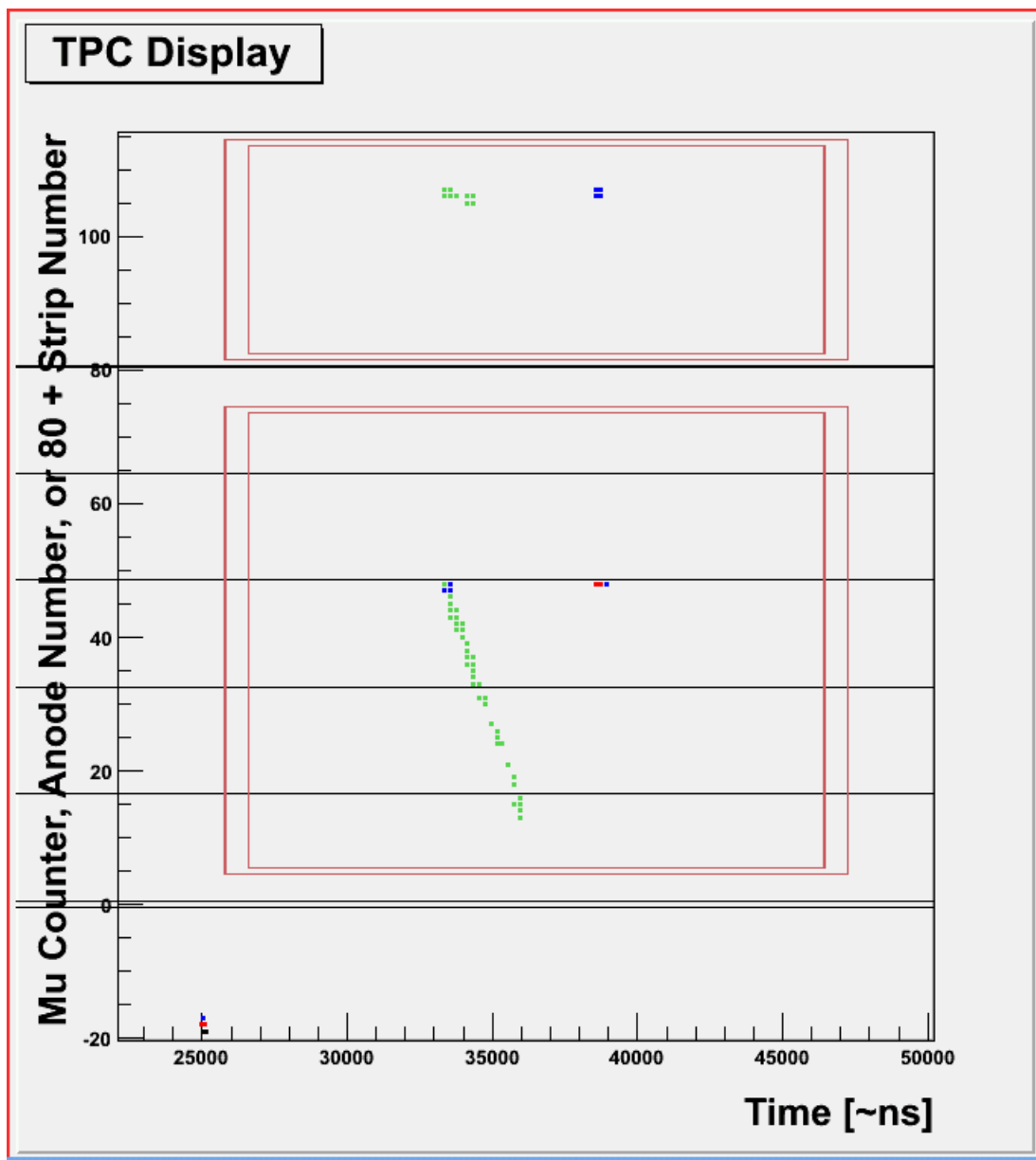


Figure 32: Example of a probable high- Z impurity capture event, from run 14630 in the N_2 -doped fill (CalibN2).

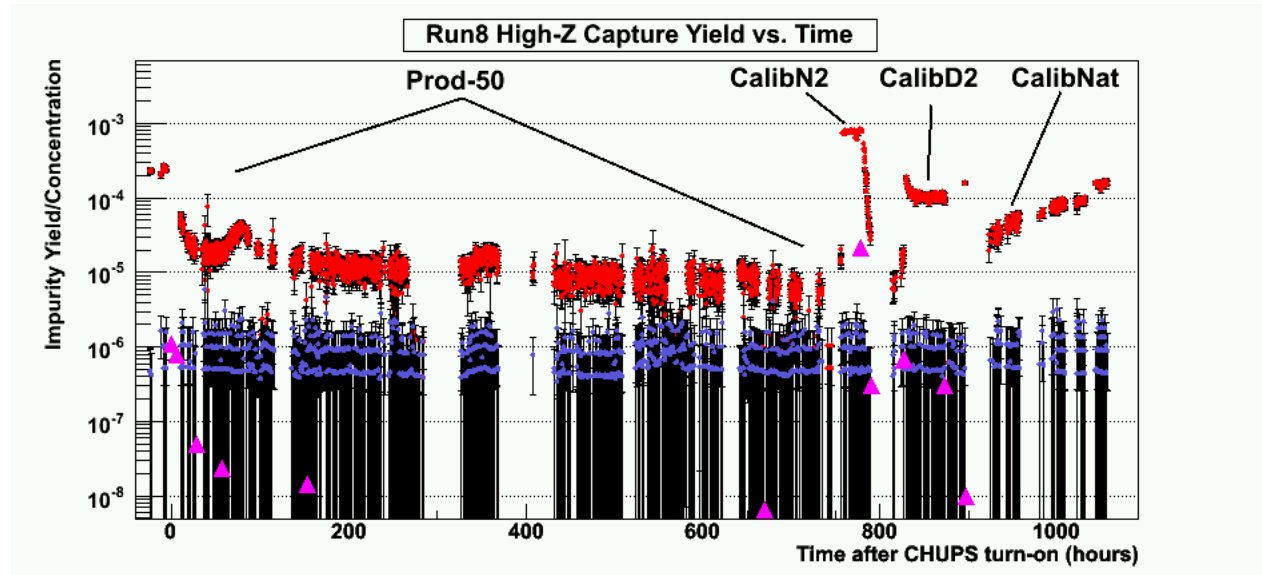


Figure 33: The high- Z capture yield vs. time during Run8. The red points are EVH captures, and the blue points are EVH control captures. (The reason the control points lie in bands is because the statistics are so low.) Note that the EVH capture yields are well above the flat noise from the control search. The EVH capture points, as well as the gas chromatography measurements of the nitrogen concentration in ppm (purple triangles), track nicely with the expected exponential cleaning effect from CHUPS. We believe that the residual difference between the EVH capture and control points is due to outgasing from the TPC materials. The four primary gas fillings are labeled; note the wide variation in high- Z impurity levels between them.

6 Lifetime studies

6.1 The muon lifetime histogram

Our immediate goal in MuCap is to extract the effective muon disappearance rate in hydrogen from the experimentally observed time spectrum of decay electrons $N_e(t)$, which is proportional to the surviving muon population $N_\mu(t)$:

$$N_e(t) = - \left. \frac{dN_\mu}{dt} \right|_{\text{decay}} = \lambda_0 N_\mu(t) . \quad (3)$$

Histograms of the decay time spectrum are created by the module `MNtupleAnalysisMQL.MQL`, which joins together muon and electron detector data which are temporally coincident within a $[-40 \mu\text{s}, 40 \mu\text{s}]$ interval, makes appropriate cuts, reduces the remaining data down to a unique set of μSC and eSC times, and then histograms the resulting time differences $\Delta t = t_{\text{eSC}} - t_{\mu\text{SC}}$ into 1.25-ns bins. The histograms are written out to ROOT files for later study.

When forming these lifetime histograms, it is desirable to minimize background contributions as much as possible, and thereby obtain a stronger decay signal. In particular, we want to correctly associate each decay electron with its parent muon. There are several ways to improve the signal-to-noise ratio. First, we can protect against muon pileup in the TPC. That is, we only consider muons whose arrival is separated in time from other muons by a $\pm 25 \mu\text{s}$ buffer. Since the TPC drift volume takes approximately $23 \mu\text{s}$ to completely “flush,” the pileup protection requirement ensures that (to a high degree) there is only one muon track in the TPC at a time, and therefore there is no ambiguity in assigning a μSC arrival time to a muon stop in the TPC. The $25 \mu\text{s}$ pileup protection requirement unfortunately cuts a large fraction of the data ($\sim 69\%$ of all muons from our $\sim 21 \text{ kHz}$ Run8 muon beam), but it is an absolutely essential step: if we use the TPC to identify muon stops, then we *must* perform pileup protection to ensure that the lifetime histogram has a flat background [9]. By using a pileup protection window, we also protect against rate effects in the detectors.

A $25 \mu\text{s}$ pileup-protected lifetime histogram is presented in Figure 34, alongside a non-pileup-protected lifetime histogram for comparison. Notice that a large portion of the background has been “dug out” from around the decay signal in the pileup-protected lifetime histogram. We can identify three components in the pileup-protected lifetime histogram:

1. The exponential decay electron signal.
2. The flat accidental background (ABG) directly beneath the decay signal. The ABG is comprised of time differences between pileup-protected muons and uncorrelated electrons. It receives contributions from cosmics, unseen muon arrivals, and probably detector noise to some degree.
3. The pileup background at the edges arises from the time differences between electrons and muons outside of the $25 \mu\text{s}$ pileup protection buffer.

We can deepen the pileup protection trough even further by using additional electron detectors, and by performing cuts on the muon/electron impact parameter. The successive improvements in the signal-to-background ratio are presented in Figure 35 ².

²There is, in principle, an alternative to our $25 \mu\text{s}$ “global” pileup protection procedure. MuCap once considered the possibility of enforcing “local” pileup protection, where each decay electron is uniquely matched with its parent muon based upon the three-dimensional intersection of the two particle tracks (so-called “vertex matching”). This is an exceedingly difficult procedure, fraught with all sorts of potential systematic distortions from rate effects and track overlap topologies (e.g. the so-called muon arrival

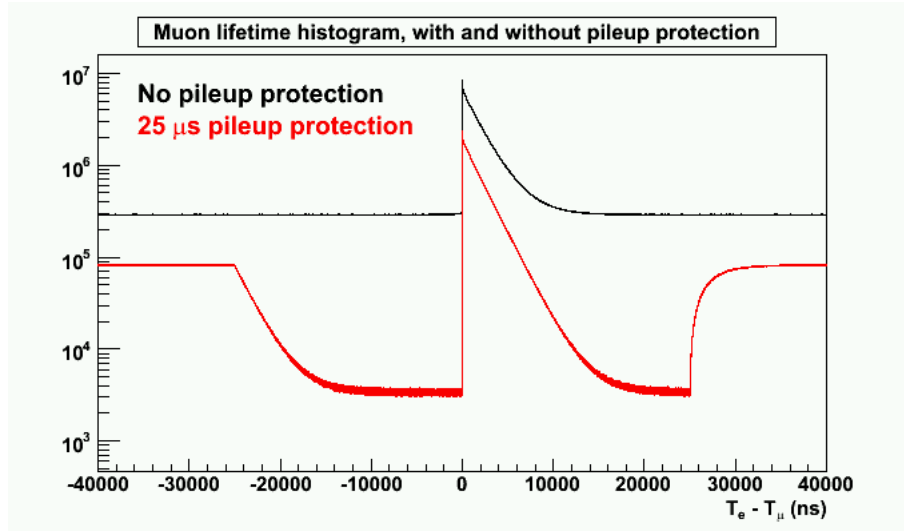


Figure 34: Muon lifetime histograms constructed from μ SC and eSC detector data, with and without pileup protection.

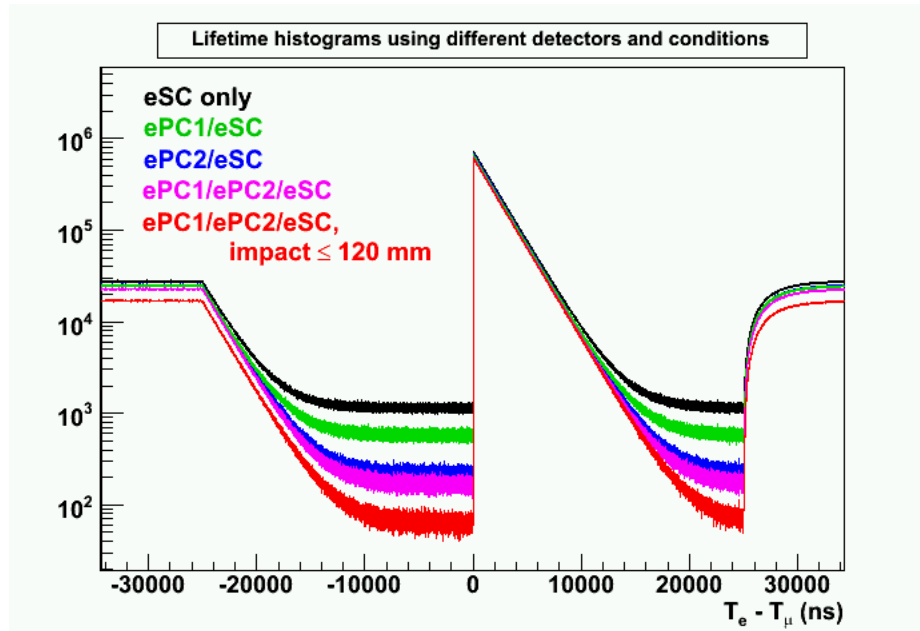


Figure 35: Successive improvements in the signal-to-background ratio are made as more electron detectors are used, and with the enforcement of a reasonable impact cut at 120 mm. Ultimately, we achieve a four-orders-of-magnitude difference between the accidental background level and the decay signal peak.

We use MINUIT via ROOT to perform fits to the lifetime histograms. Our fit function is a single exponential of the form

$$f(t) = N\lambda e^{-\lambda t} + B, \quad (4)$$

where B is the accidental background, N is the overall scaling factor, and λ is the effective disappearance rate. The repeated appearance of λ serves to reduce the off-diagonal term in the covariance matrix that links λ and N , thereby improving fit convergence; this procedure should not change the fitted value or uncertainty in λ . In order to use the fit function in Equation 4, we must confine our fits to the interval $[0, 25] \mu\text{s}$, inside of the pileup protection background. Although we can describe the basic shape of the pileup background in mathematical terms, we do not bother to include it in our fits because its edges are “messy”, containing multiple exponential components.

Note too that Equation 4 describes a simple exponential decay signal. In reality, the time spectrum has a far more complicated shape due to effects from impurities (both elemental and isotopic) and molecular formation. However, the impurity concentrations are sufficiently small that the resulting perturbations to the effective decay rate are linear, and we correct for them later. Similar considerations apply to the effects from molecular formation, although its uncertainties are more considerable and must be taken into consideration *after* the unblinding.

6.2 Cosmics bin error adjustments

Once throughgoing cosmics have been identified (see Section 5.4), the question remains: How can we use this information in the analysis of lifetime histograms? One possibility is to subtract off the portion of accidental background attributable to the identified cosmics (Figure 36). However, I am wary of performing any cosmics subtractions on the decay spectrum, for two reasons: (1) it is not essential to remove cosmics, because they only contribute a flat background; (2) I am concerned about the possibility that imperfections in my cosmics identification procedure could introduce a time-dependent effect to the decay spectrum. In fact, as Figure 36 shows, a small residual muon decay component is just barely visible in my ostensibly cosmics-only background, so my identification is manifestly imperfect.

A more attractive alternative was proposed by Steve Clayton: rather than subtract off those lifetime entries where the eSC hit has been attributed to a cosmic, one can simply enlarge the errors on the decay spectrum bins to adjust for the double-counting produced by cosmics. That is, the error on a bin with N counts is generally taken as $\sqrt{N}$, but this value is too small because it assumes that all N events are independent, whereas some of them are actually pairs of hits from a single cosmic event. So how should the error be properly calculated? To answer that, let us write the number of counts N in a given bin as the sum of signal and accidental background terms,

$$\begin{aligned} N &= S + B \\ &= S + B_{\text{cosmics}} + B_{\text{other}}, \end{aligned}$$

where B_{cosmics} is the portion of total background B due to cosmic events. Since the cosmic events generate pairs of hits, the corresponding error is

$$\sigma_{B_{\text{cosmics}}} = 2\sqrt{\frac{B_{\text{cosmics}}}{2}} = \sqrt{2B_{\text{cosmics}}},$$

“double-kill,” etc.). Furthermore, it is perhaps impossible to achieve with our current experimental setup, since scattering from the pressure vessel significantly broadens the impact distribution and prevents accurate vertex matching. In all likelihood, local pileup protection will most likely never be realized in MuCap, and we will stick with the far simpler global pileup protection.

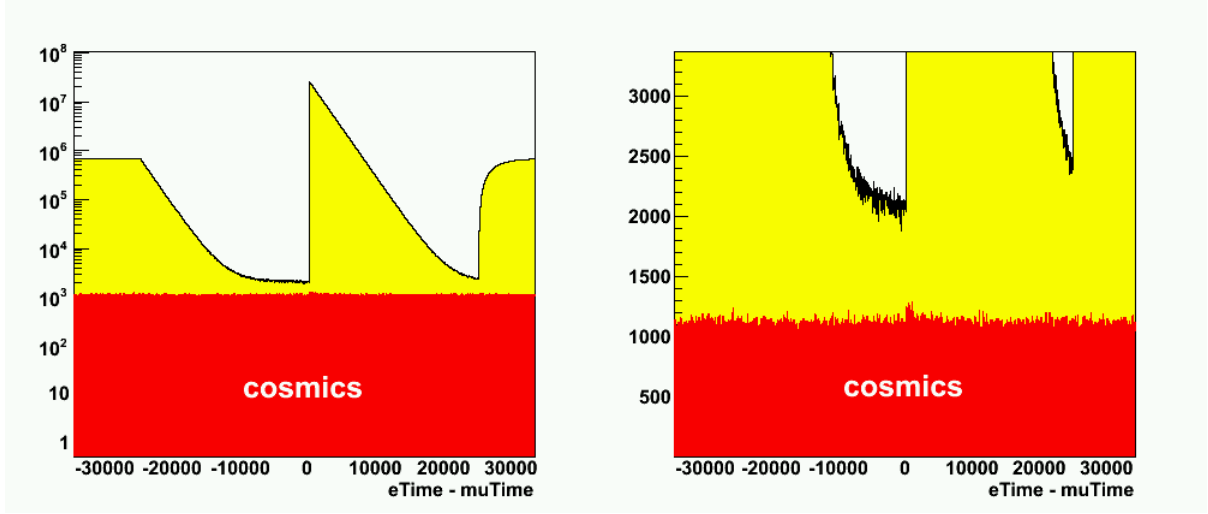


Figure 36: These plots show the contribution from cosmic events (red) to the lifetime histogram's accidental background. The plot on the left is a log-scaled wide view of the decay time spectrum, while the plot on the right is a zoomed-in linear-scale view of its accidental background. In this latter plot it is clear that cosmic events account for approximately half of the accidental background. Note too that there is a small residual decay curve visible in the cosmics histogram at $t = 0$, further reason to not cut cosmics from the main decay spectrum.

while the error for the “other” background contribution is simply

$$\sigma_{B_{\text{other}}} = \sqrt{(B - B_{\text{cosmics}})}.$$

If we plug all of these formulas into the expression for the bin error, we get

$$\begin{aligned} \sigma_N^{\text{corrected}} &= \sqrt{\sigma_S^2 + \sigma_{B_{\text{cosmics}}}^2 \sigma_{B_{\text{other}}}^2} \\ &= \sqrt{S + (2B_{\text{cosmics}}) + (B - B_{\text{cosmics}})} \\ &= \sqrt{S + B + B_{\text{cosmics}}} \\ &= \sqrt{N + B_{\text{cosmics}}}. \end{aligned}$$

According to Figure 36, $B_{\text{cosmics}} \approx (B/2)$. So, to adjust the bin errors for cosmics, I first perform an initial fit to get an estimate for the background B . I then change each bin's error from $\sqrt{N} \rightarrow \sqrt{N + B/2}$ and perform a second fit to obtain the final result. I have found that this two-fit procedure generally produces a dramatic improvement in the χ^2 of the adjusted fit (e.g. from $\chi^2=1.15$ to 1.05), but only lowers the fitted rate very slightly, by ~ 0.25 Hz at most.

Except when analyzing the unique electron lifetime histograms, I always assume that cosmics account for half of the accidental background, and adjust the bin errors according to the procedure described above. I should note, however, that I have established that cosmics are half of the background for a single set of lifetime histogram conditions only. Nevertheless, it seems reasonable to assume that cosmics account for roughly half of the accidental background under other conditions as well. In any case, it has a negligible effect on the fitted rate and primarily improves the χ^2 .

6.3 Rebinning

Each of the CAEN modules has a built-in interpolator that subdivides the externally provided 40-ns-period (25 MHz) clock signal into 32 time bins, thus yielding an ostensible time resolution of 1.25 ns. However, the interpolator subdivision is egregiously nonuniform, with variations that are far larger what is statistically allowed, as can be seen in Figure 37. In order to eliminate the potential for effects from the interpolators’

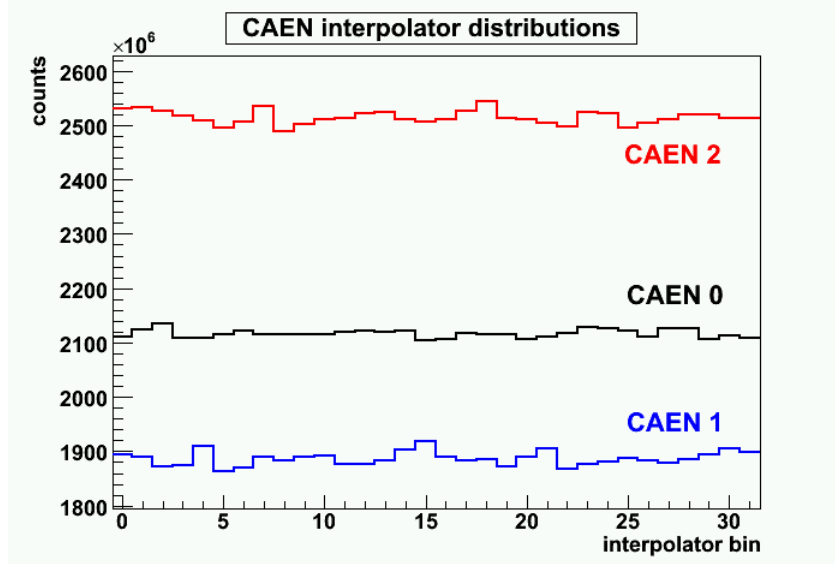


Figure 37: *Blah.*

nonuniformity, we rebin the 1.25-ns-bin-width lifetime histograms by 32. The resulting fitted decay rates are within ~ 3 Hz of the unbinned rates, and the χ^2 values are comparable. Rebinning by 64 gives nearly identical rates as rebinning by 32, but with slightly better χ^2 values.

There is also the question of where to set the 32-rebinning boundary. Our lifetime fits generally start after 100 ns, in order to allow full depopulation of the μp triplet state. Consequently, we should perhaps be careful about setting the rebinning boundary—we don’t want the first 40-ns-wide bin to include too much information from the period before the fit start time. To check the sensitivity of the fitted decay rate (and corresponding χ^2) to the location of the 32-rebin boundary, I wrote a custom rebinning function which allows me to specify the location of the rebinning boundary (the built-in ROOT `Rebin()` function does not have that versatility), and ran fit scans for boundaries at locations between 30–110 ns, in 1.25 ns steps, with a fit start time of 110 ns. The results are plotted in Figure 38, which shows that there is an approximately 2 Hz variability over the range of boundary locations. There is greater volatility in the χ^2 values. For my standard conditions, $t_{\text{rebin}} = 105$ ns and $t_{\text{start}} = 110$ ns, the χ^2 is better than 1.

6.4 Fit range

We know that the start time of our fits should be greater than 100 ns to allow the μp triplet state to depopulate, and that the fit stop time should be less than 25 μ s in order to avoid overlapping into the pileup background. But how stable are the fitted rates for variations within these boundaries?

Start and stop time fit scans for the cathode-AND and cathode-OR histograms are presented in Figures 39

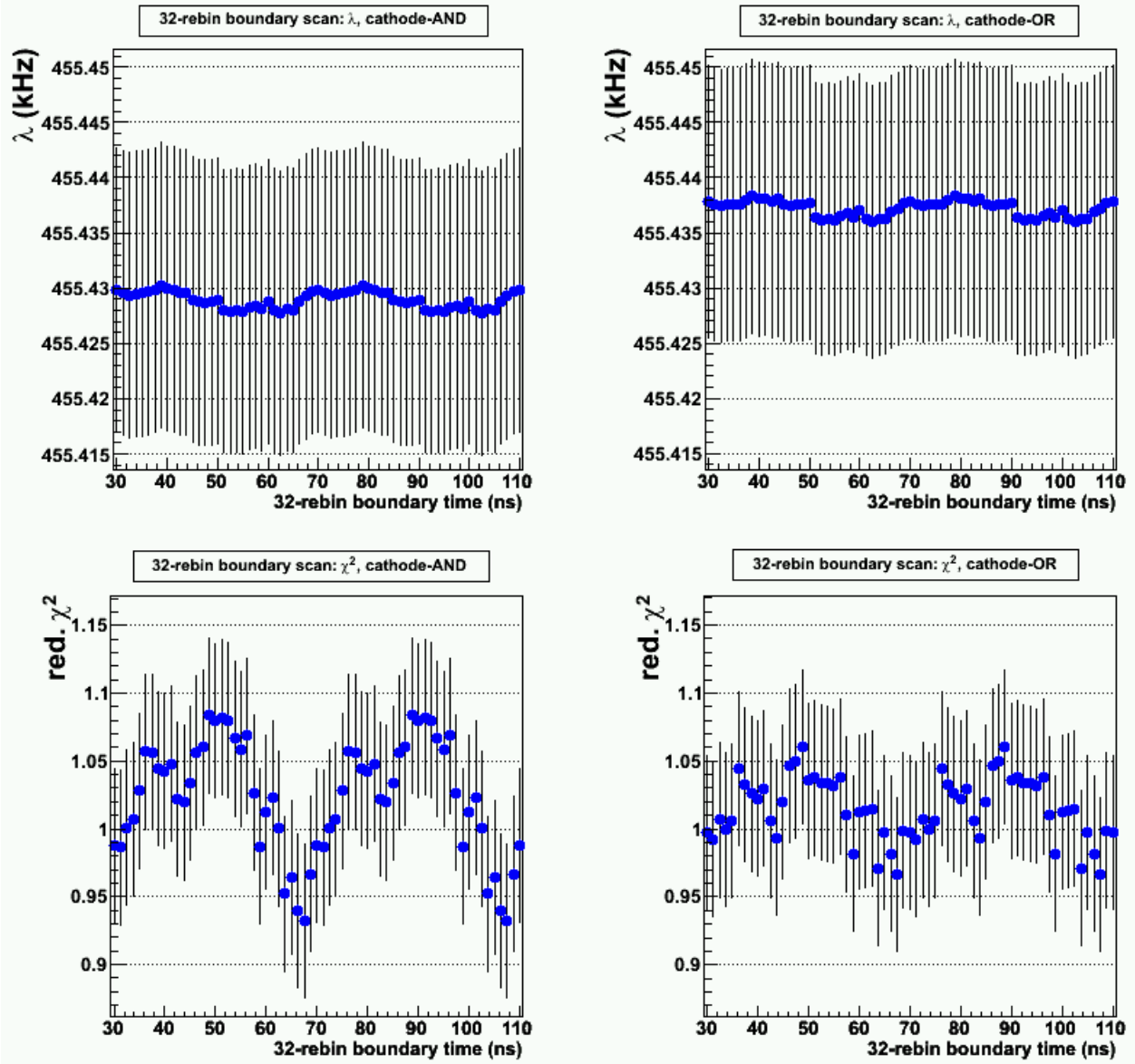


Figure 38: Fit results from time scans over the location of the 32-rebin boundary, from 30–110 ns in 1.25 ns steps. The fits were performed over the interval [110,24900] ns. Notice that a periodic structure is evident in these scans, which is to be expected: you should see the same results for rebin boundaries at x ns and $(x + 40)$ ns.

and 40. The start time scan runs from 50–2000 ns in 25 ns intervals, with the fit stop time fixed at 24900 ns. The stop time scan runs from 22500–24950 ns in 25 ns intervals, with the fit start time fixed at 100 ns. The rebin boundary was fixed at 95 ns. I have drawn the so-called Kawall bands around the start and stop time scans’ fitted rates, to indicate the 1σ statistical deviation expected among the correlated fit results to data subsets ($\sigma = \sqrt{\sigma_1^2 - \sigma_2^2}$). The greater volatility in the start time scans’ fitted rates is due to the fact that there is more information in the decay spectrum at early times; in contrast, the stop time scan includes a much smaller fractional variation in histogram counts, and so the λ values appear much more stable. Overall, the fitted rates are completely consistent with the Kawall envelope, so the time scans exhibit no problems.

I mentioned earlier that the 16 eSC gondolas all have slightly different timing offsets relative to one another. This is illustrated in Figure 41, where I have plotted the lifetime histograms for the sixteen gondolas, and zoomed in on the lifetime “turn-on” region near $\Delta t = 0$. In effect, we are using the μ SC as a reference time in Figure 41 so that we can observe the spread in eSC times. It is clear from the lifetime histograms that the gondola times are offset from one another by 5 ns at most. Furthermore, the “latest” gondola’s turn-on is complete by $\Delta t = 10$ ns. In light of these circumstances, I chose to set my standard fit start time at 110 ns, with the rebinning boundary at 105 ns; the standard fit stop time is 24,900 ns.

6.5 Selection of input tables

When Steve and I unblinded our relative rate offsets in May 2006, we found that our results were separated by ~ 16 Hz—far larger than the allowed statistical variation. Subsequent studies revealed that our different μ PC1 and ePC cathode treatments were responsible: Steve used a μ PC1XY-OR treatment, while I used μ PC1XY-AND, and Steve used a ePC cathode-OR treatment, while I used ePC cathode-AND. Ultimately, Steve and I both observed that different μ PC1 and ePC treatments in our analyses produced different fitted rates, although Steve’s variations were all within statistics, while mine were approximately twice as large.

This raised the question of which muon and electron tables should be used to form the lifetime histograms which I fit to get my final result. I ultimately chose to use the μ PC1XY-OR table rather than the μ PC1XY-AND table, since the former imposes a more stringent pileup protection on muon arrivals. It is more difficult to choose between the ePC cathode treatments, however: the cathode-AND condition provides better tracking, but is probably more susceptible to deadtime effects than the cathode-OR treatment. Furthermore, the rate differences between the two approaches (~ 8 Hz) are much larger than what one would expect from statistical variation (~ 3.7 Hz), which suggest that at least one of them is being skewed from some unknown effect. Unfortunately, at present I have no way of knowing which is superior, so I have chosen to present a result for both treatments for the time being. In Table 1 I present fitted rates for different lifetime conditions, all using the μ PC1XY-OR treatment and a standard fiducial TPC volume cut. Notice the larger-than-statistical variations between the eSC-only rate and the no-impact-cut eDet treatments. This is a source of some concern.

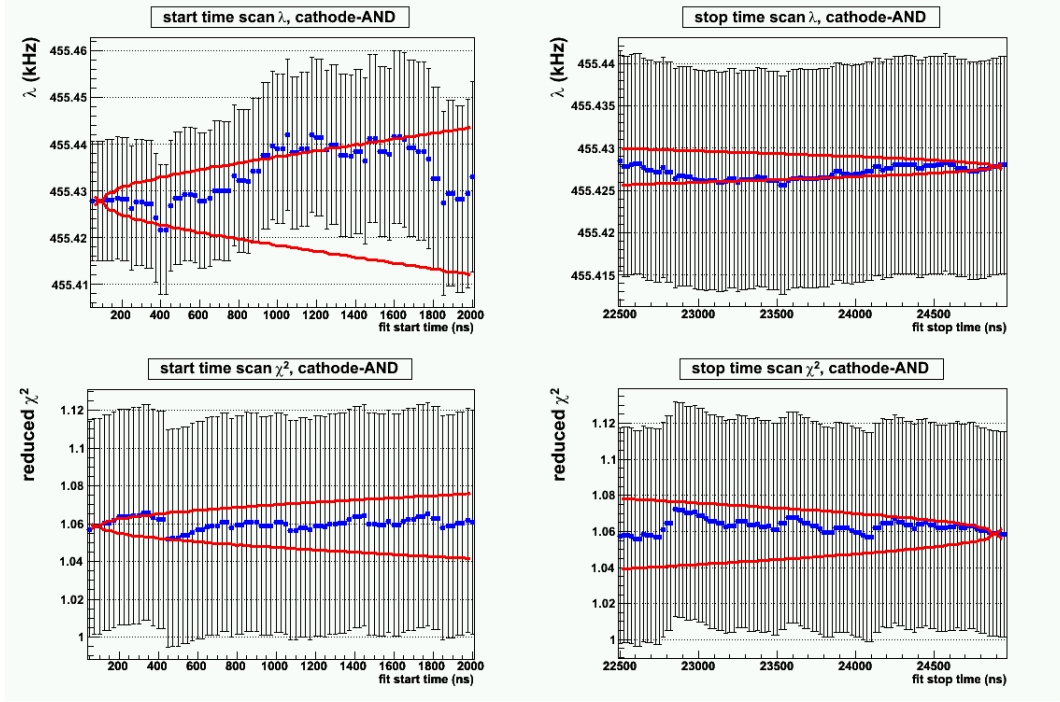


Figure 39: Start and stop time scan fit results for the cathode-AND lifetime histograms. The custom ROOT functions `plotKawall` and `plotCSKawall` were used here.

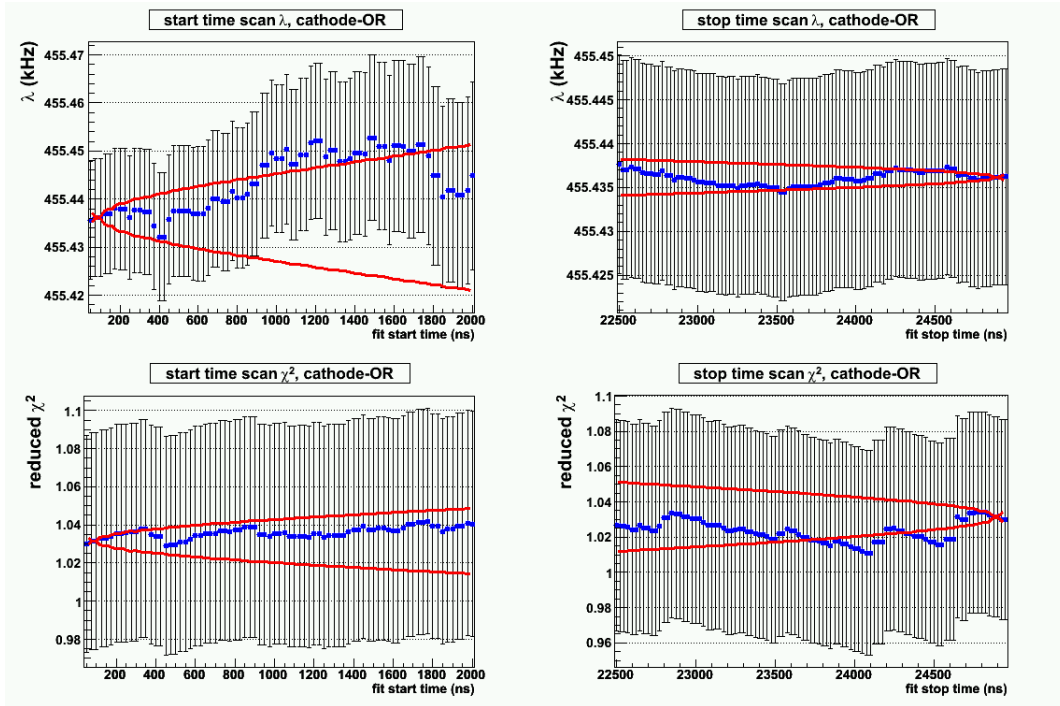


Figure 40: Start and stop time scan fit results for the cathode-OR lifetime histograms. The custom ROOT functions `plotKawall` and `plotCSKawall` were used here.

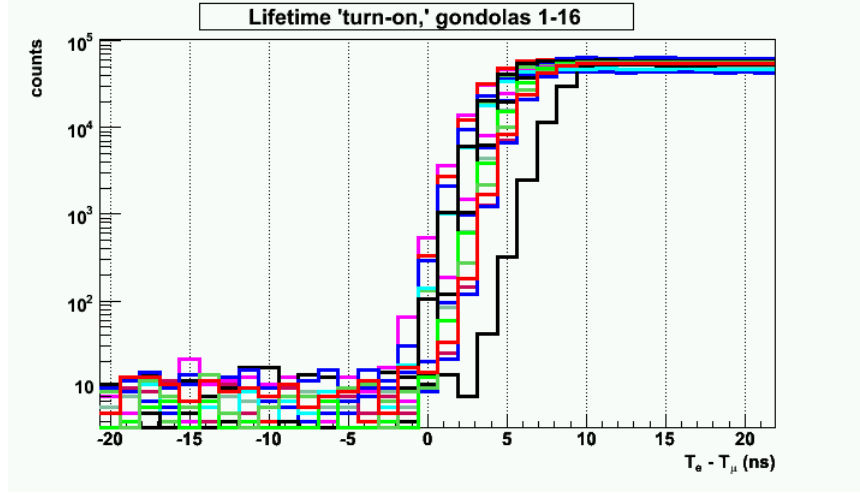


Figure 41: Lifetime histograms for the sixteen eSC gondolas, zoomed-in around the time region $\Delta t = t_e - t_\mu = 0$. The experimental timing offsets among the gondolas are evident in the staggered “turn-on” times of their lifetime signal.

Fill	eDet table	impact cut	λ (s ⁻¹)	σ_λ (s ⁻¹)	χ^2/d
Prod-50	eSC-only	—	455,433.94	12.86	0.956
	cathode-AND	—	455,417.09	12.96	0.974
	cathode-AND	120 mm	455,428.06	12.86	0.964
	cathode-OR	—	455,425.66	12.42	0.999
	cathode-OR	120 mm	455,436.29	12.33	0.998

Table 1: Fitted rates for different Prod-50 lifetime histograms, using different detector treatments and conditions. The variations among the eSC-only, cathode-AND, and cathode-OR treatments are, unfortunately, much larger than what is expected based upon the “allowable” set/subset deviation $\sigma = \sqrt{\sigma_2^2 - \sigma_1^2}$.

6.6 Fiducial cuts

Recall that I postpone my TPC fiducial volume cuts until just prior to the creation of lifetime histograms, which allows me to explore the effects of fiducial volume cut variations on the fitted rate. I will pursue this topic further in the following subsection, but first I should introduce the basic features of my fiducial cuts. My standard TPC fiducial volume is defined as:

```
define(STANDARD_FIDTPC_CUT, '  
(MIN(lowstrip, (kTPCNumStrips-highstrip+1)) > 2) &&  
(MIN(BraggLowstrip, (kTPCNumStrips-BraggHighstrip+1)) > 3) &&  
((mostLtime-muSCTime) > (kTPCMWPCOffset +  
5.0*(kTPCSVDriftInterval/kTPCYDimension))) &&  
((mostRtime-muSCTime) < (kTPCMaxDriftTime -  
5.0*(kTPCSVDriftInterval/kTPCYDimension))) &&  
((EHminLtime-muSCTime) > (kTPCMWPCOffset +  
15.0*(kTPCSVDriftInterval/kTPCYDimension))) &&  
((EHmaxRtime-muSCTime) < (kTPCMaxDriftTime -  
15.0*(kTPCSVDriftInterval/kTPCYDimension))) &&  
(EHendanode > kTPCFirstAmpAnode) && (EHendanode < 68) &&  
(ELdistUS >= 5) &&  
((EHendanode+ELdistDS) < kTPCLastAmpAnode) &&  
(EHcountDS == 0)')
```

In the x -dimension (TPC cathodes 1–35), I require cathode hits in the strip interval [4,32], coincident with the anode EH Bragg stop. I also require that these cathode hits cannot extend outside of the interval [3,33], in order to protect against muon escapes in that direction.

In the y -dimension, my cut has two “walls”: an outer y -boundary for EL track pixels, and an inner y -boundary for EH Bragg stop pixels. A two-walled cut is necessary because, as Steve Clayton has demonstrated, decay-electron-generated EL pixels can interfere with the mustop identification algorithm.

In the z -dimension, I enforce a variety of conditions. The interval of functional, amplifying anodes is [5,73]. I require that the mustop’s endanode (that is, the furthest anode in an EH cluster) lie within the (inclusive) interval [6,67]. However, due to constraints upon the length of EL “lead-up” pixels (>5), the endanode interval effectively becomes [10,67]. A maximum anode of 67 is chosen to allow for the identification of muon scatters and escapes in the z -direction. In a similar vein, I require an absence of EH pixels beyond the ostensible stop, to make sure this EH cluster is a stop, and I require that any EL pixels downstream of the ostensible stop cannot extend all the way to the last anode ($kTPCLastAmpAnode=73$).

The efficacy of my muon stop identification and fiducial cut algorithms is apparent in the start time scans in Figures 39 and 40: If my fiducial cuts were not providing adequate protection against wallstops—or if there were gross and irremediable inefficiencies in the muon detectors, such as malfunctioning TPC cathodes—then there would be evidence of high- Z lifetime components in the start time scans. The absence of any such obvious effects leads me to believe that wallstop contributions are minimal. In fact, in the next chapter I will perform a small correction for wallstop effects, based upon my muon scatter studies.

6.7 Uniformity of rates across gondolas

An issue which has occupied much of our attention over the past year is the uniformity of the fitted rates across the sixteen eSC gondolas (Figure 42). Ideally, the rate variations among the gondolas should be purely statistical in nature, and thus a ROOT `pol0` (constant) fit across the gondolas’ rates should give a reduced χ^2 value consistent with 1. However, in the initial stages of our analysis (circa 2005), we observed

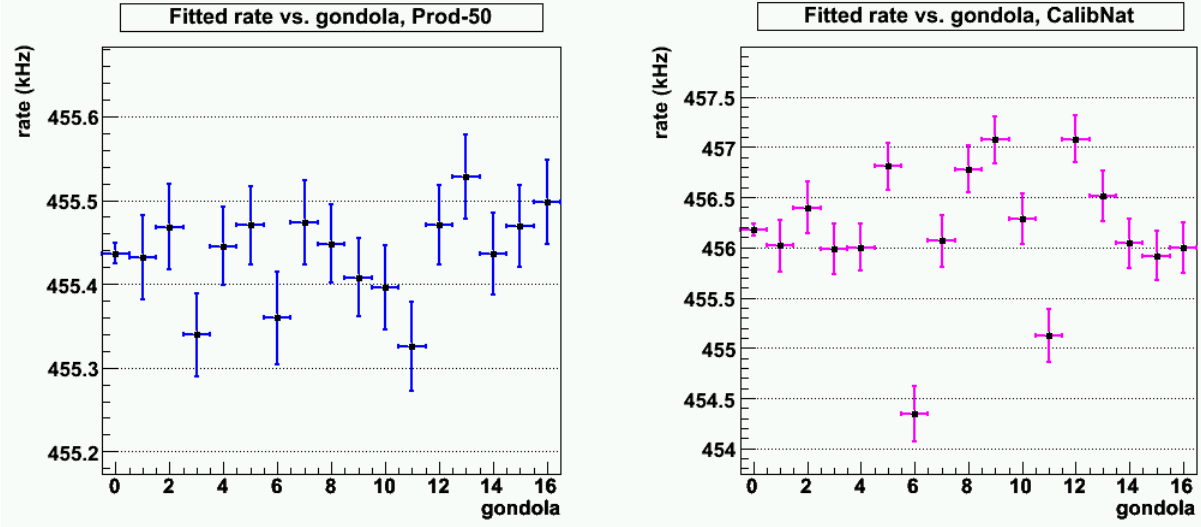


Figure 42: Examples of fitted rate vs. gondola for both the Prod-50 and CalibNat fills (gondola “0” here is the histogram which sums over all sixteen gondolas). For this particular Prod-50 lifetime histogram, the reduced χ^2 from a constant pol0 fit across gondolas 1–16 is ~ 1.2 , which corresponds to a probability is $\sim 25\%$ for a 15-d.o.f. fit. Notice that the per-gondola nonuniformity is significantly worse for the CalibNat fill. As explained in the text, this is a consequence of the rapid μd diffusion in hydrogen gas, coupled with electron scattering from the TPC frames.

marked nonuniformities in our fitted rates across the gondolas, reflected in egregiously high pol0 χ^2 values. At the time we simply dubbed this phenomenon the “gondola effect,” as we did not understand its origins.

After much investigative effort, we have come to believe that “gondola effects”—that is, nonuniformities in the fitted rates across the gondolas—can be created by two separate mechanisms:

1. Decay-electron-generated EL pixels can interfere with the identification of muon stops. Steve demonstrated the existence of this TPC response phenomenon, which is emission-direction dependent, and showed how his initial mustop identification algorithm was especially sensitive to electron effects.
2. Coupling between the processes of rapid μd diffusion and scattering from TPC frames conspires to produce a “gondola effect” which scales with increasing deuterium concentration. I was able to convincingly reproduce this effect with Monte Carlo simulations.

We have no control over the μd -diffusion-driven “gondola effect”—it is an inescapable feature of our experimental setup. Fortunately it appears to have a negligible effect in the deuterium correction’s zero-extrapolation procedure.

The TPC-response-related mechanism is more problematic. We must be careful to employ muon stop identification criteria that are not systematically affected by the occasional appearance of time-delayed, decay-electron-generated EL pixels. This is the primary reason why Steve and I use “double-walled” fiducial cuts in the y -direction. In order to check the sensitivity of the fitted rates to the fiducial cuts, as well as the corresponding uniformity across gondolas, I created lifetime histograms with incremental variations in the cut specifications. The results of these scans over fiducial cuts are plotted in Figure 44 (cathode-AND) and Figure 45 (cathode-OR). The scans are performed over (1) the length of the EL lead-up track to the Bragg stop, (2) the distance of the maximum allowable endanode from the last anode, and the (3) position of the

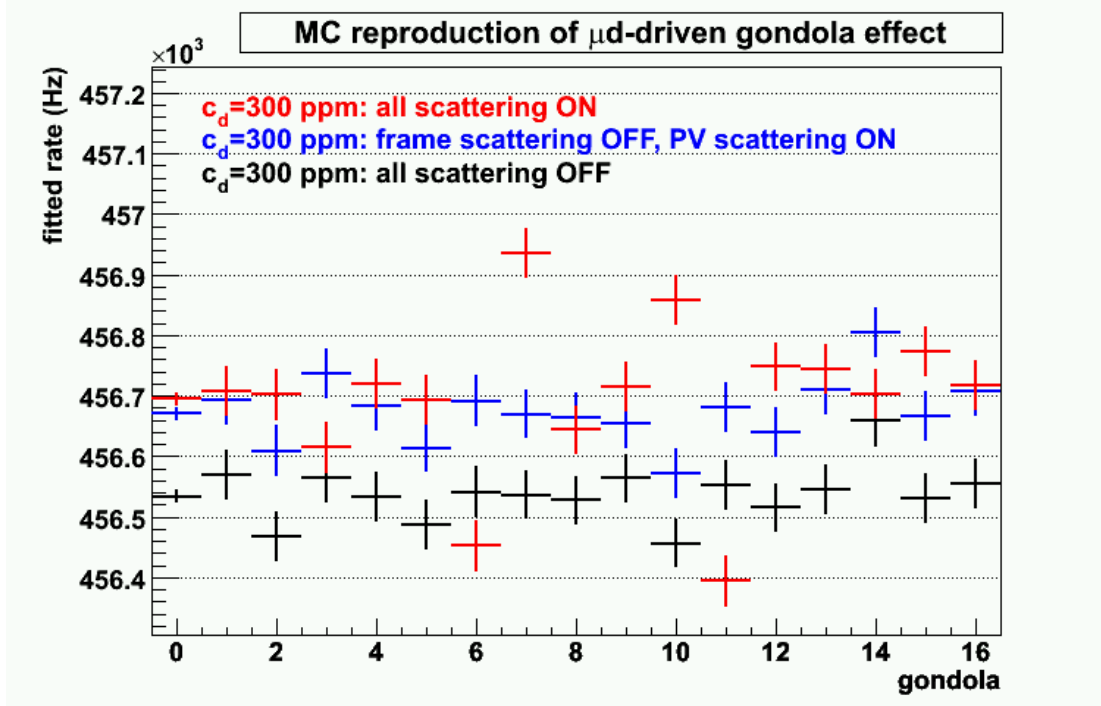


Figure 43: Reproduction of the deuterium-driven “gondola effect”, created using my fast Monte Carlo software. The three sets of rate vs. gondola distributions I have presented clearly demonstrate that this particular gondola effect is intimately connected with scattering from the TPC frames. Note how the “all scattering ON” rates in gondolas 6 and 11 are pulled downwards relative to the others by this effect, just as we see in the Run8 CalibNat data in Figure 42. I should note that I had to use a nominal MC concentration of $c_d = 300$ ppm to achieve results similar in scale to what we see in Run8 at $c_d = 120$ ppm.

EH y -boundary, which is stepped inwards away from the outer EL boundary. The Kwall bands are drawn relative to those fiducial cuts which are closest to my standard cut (scan indices 15 and 35). The plots for the lead-up and endanode cuts look fairly stable and unremarkable. The y -boundary cut is of the most concern, since it is the most sensitive to the appearance of electron-generated EL pixels. Indeed, when the EL and EH boundaries are coincident (scan indices 10 and 30), the “gondola effect” is highly pronounced. As the EH wall is moved inwards, the per-gondola χ^2 steadily decreases to around 1.2, where it appears to stabilize. My standard cut seems to be reasonable.

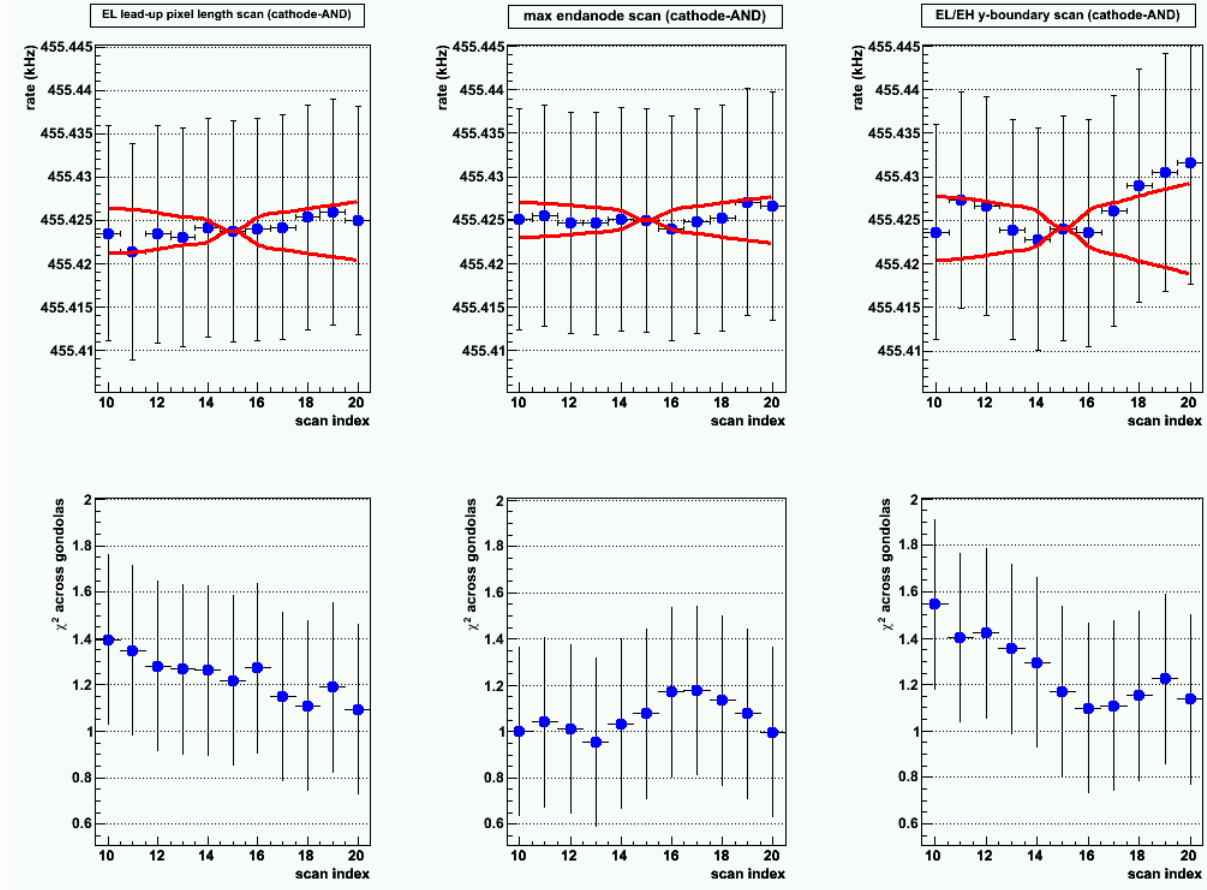


Figure 44: Fiducial cut scans for lifetime histograms involving the cathode-AND electron table. For the EL lead-up study, scan indices 10–20 correspond to lead-up lengths 0–10; for the endanode study, indices 10–20 correspond to a maximum anode 72–62; for the EH boundary study, indices 10–20 correspond to an EL/EH boundary separation of 0–20 mm in 2 mm increments (the EL boundary is 5 mm from the edges of the sensitive volume).

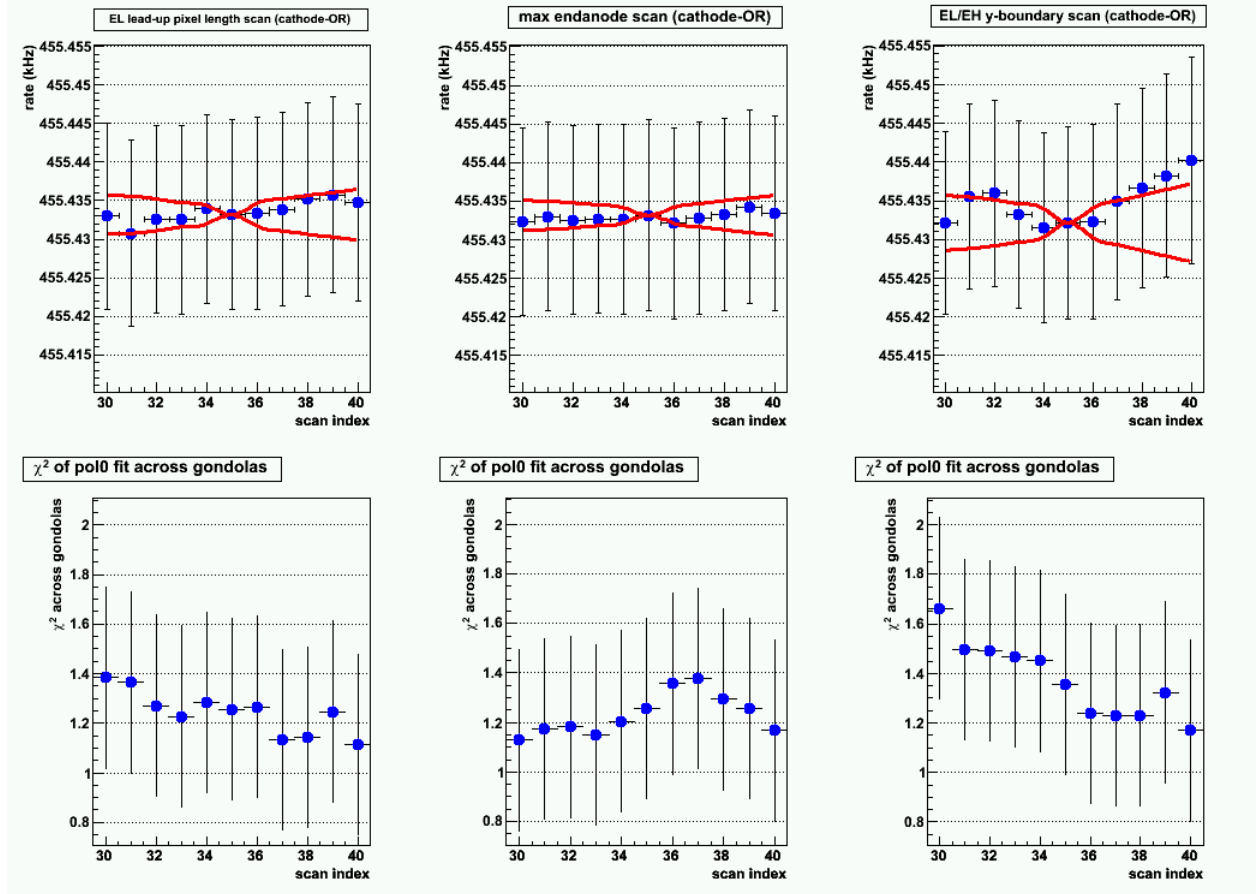


Figure 45: Fiducial cut scans for lifetime histograms involving the cathode-OR electron table. For the EL lead-up study, scan indices 30–40 correspond to lead-up lengths 0–10; for the endanode study, indices 30–40 correspond to a maximum anode 72–62; for the EH boundary study, indices 30–40 correspond to an EL/EH boundary separation of 0–20 mm in 2 mm increments (the EL boundary is 5 mm from the edges of the sensitive volume).

6.8 Data selection

In the Run8 μ^- data analysis, there are four primary run periods of interest, corresponding to different hydrogen gas target compositions (Table 2). Prod-50 forms our main clean production data set and provides

Designation	Description	TPC voltage (V)	Run numbers	Statistics
Prod-50	Clean fill production data	5017	11650–14140	1.56×10^9
CalibN2	Nitrogen-doped fill, $c_N \sim 11$ ppm	4817	14530–14648	7.77×10^7
CalibD2	Deuterium-doped fill, $c_d \sim 20$ ppm	4817	14738–15036	2.13×10^8
CalibNat	Deuterium-doped fill, $c_d \sim 120$ ppm	4817	15072–15220	6.24×10^7

Table 2: Run8 fills of interest in my μ^- analysis. The “designation” label is from Dr. Francoise Mulhauser’s organization of the Run8 data subsets. Much of the information in her summary was incorporated into a MySQL database on dolphin, which we then used to organize and track analyses of the Run8 data.

the starting point for our decay rate measurement. The other three “Calib” fills are used to inform corrections to the Prod-50 fitted rate for effects from high- Z and deuterium impurities. Note the variation in TPC voltage between the Prod-50 and calibration fills. An operating voltage of 4.8 kV is believed to have little effect on the identification of muon stops, but the decreased gain does lower the efficiency for the identification of scatter events and impurity captures. We chose to play it safe and use only the more reliable Prod-50 from the clean fill, and ignore the Prod-48 data. The shortcomings of a 4.8 kV operating voltage have less relevance for the calibration fills, where the statistics are lower and effects from impurities dominate.

We must consider the question of which runs to include when compiling data for the lifetime histograms, especially for the clean Prod-50 data. There were a number of variations in experimental conditions during this period—replacement of the μ SC, the PSI power outages, etc.—and we must check if they had any effect on the fitted rate.

First of all, only those runs marked as “good” in Dr. Mulhauser’s run summary were analyzed. Next, we excluded a small number of runs which exhibited obvious problems with the electron detector efficiencies. Of greater concern are the muon electron detector efficiencies: there were enormous fluctuations in the μ SC efficiency over the course of Run8 (Figure 46). Nevertheless, there were two periods of concern in our primary clean fill Prod-50 interval, indicated in Figure 46, where the μ SC inefficiency jumped dramatically. The first period, beginning at run 13067, corresponds to post-power-outage misbehavior. The second period corresponds to a time when the muon CAEN was malfunctioning. Fortunately, there are many mitigating factors that help to remedy troublesome μ SC behavior: First, a lowered μ SC inefficiency is in some ways a self-correcting problem, since the detector ceases to contribute significant statistics to lifetime histograms. Second, the presence of μ PC1 rescues the overall pileup protection efficiency to a large extent—especially if the μ PC1XY-OR condition is used—which can be seen by inspecting the counts in the pileup-protected TPC drift plot’s accidental background (Figure 47). In fits to lifetime histograms which included (or excluded) the problematic run periods (Table 3), there were no egregious deviations from the overall summed rate, so the problematic runs do not appear to pose a threat to the final result. Also, there is no obviously bad behavior in the per-run lifetime trend plots.

I should note that Steve has reported large variations in the fitted rate for different run groupings of the Prod-50 data, so the issue is not completely closed, and further study may still be required. However, I don’t consider this a “showstopper” issue.

Lastly, I should mention that Fred Gray at one point checked the μ^- data run-by-run for μ SR oscillations,

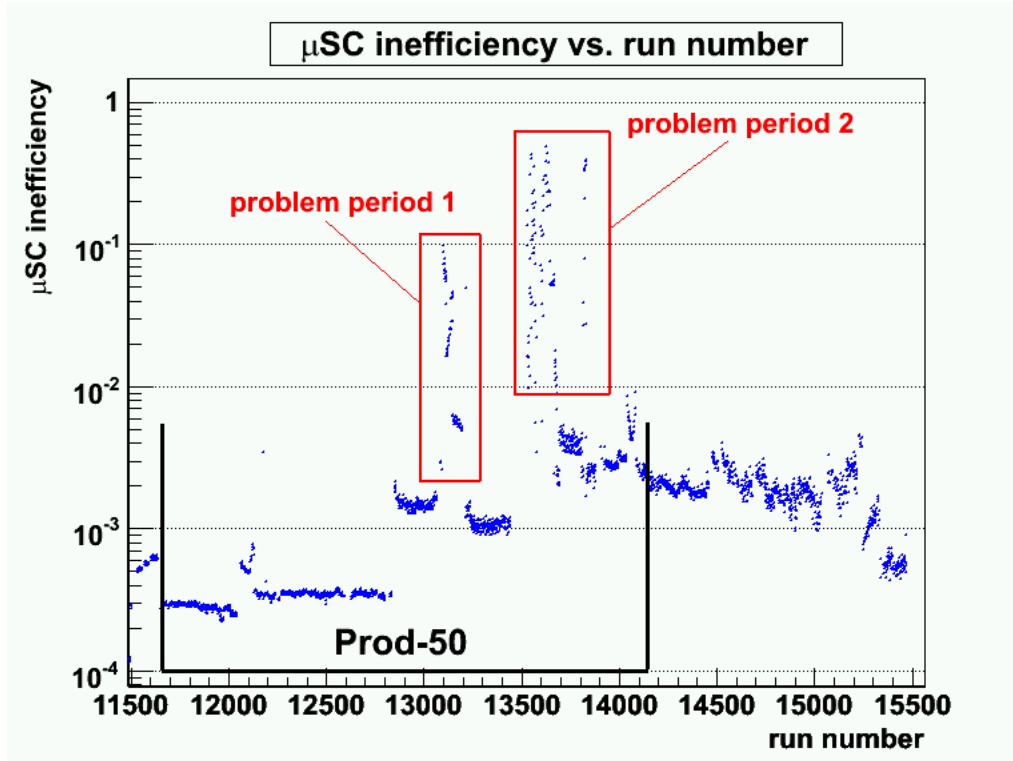


Figure 46: The μ SC inefficiency vs. run number during Run8. There are two periods of special concern where the μ SC inefficiency jumps up dramatically. The jump in the inefficiency around run 12850 can be attributed to the replacement of the 500 μ m scintillator with the 250 μ m scintillator. I should note that the numbers here are about an order-of-magnitude higher than they should be because of a flaw in my algorithm; the overall trend is correct, however.

but found no evidence of misclassified μ^+ runs.

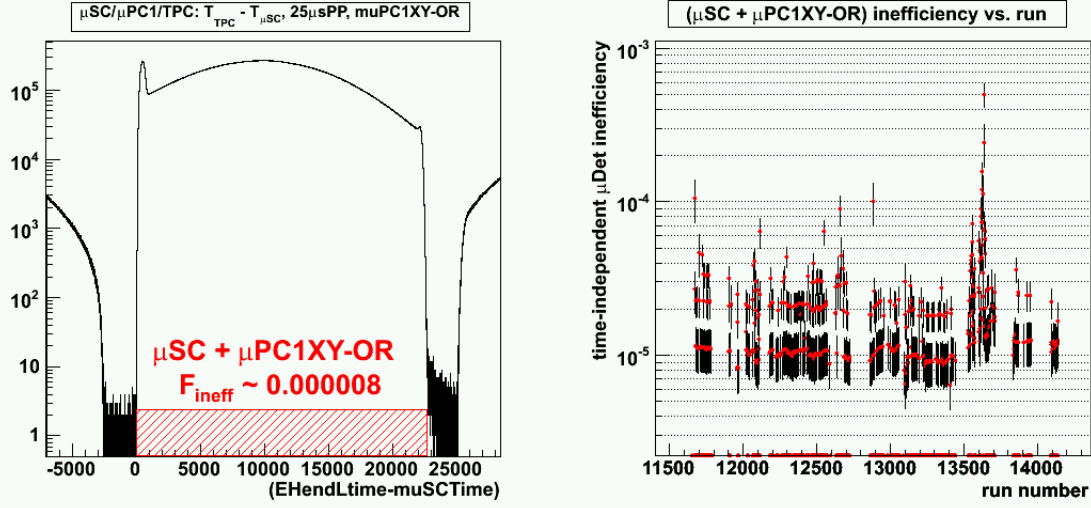


Figure 47: The Prod-50 full μ Detector TPC drift plot (using μ PC1XY-OR treatment), and the per-run μ Detector inefficiency. The detector inefficiency can be estimated by looking at the accidental background level in the TPC drift plot, and this is how I determined the per-run inefficiencies. Notice that, in comparison to Figure 46, the inclusion of μ PC1 improves the situation significantly.

lifetime histogram	Prod-50 μ SC data	λ (s^{-1})	σ_λ (s^{-1})	χ^2/d
cathode-AND	all	455,428.06	12.86	0.964
	bad period 1 only	455,415.46	69.50	1.012
	bad period 2 only	455,449.20	52.30	0.944
	bad periods 1,2 excluded	455,426.94	13.51	0.953
cathode-OR	all	455,436.29	12.33	0.998
	bad period 1 only	455,402.63	66.57	1.024
	bad period 2 only	455,479.97	50.06	0.959
	bad periods 1,2 excluded	455,434.40	12.97	0.981

Table 3: Fit results for different run groupings which include or exclude the problematic μ SC run periods illustrated in Figure 46. The bad run periods do not exhibit outlandish fitted rates, and their inclusion or exclusion appears to have little effect on the overall summed rate, so I see no reason to exclude them.

6.9 Unique electron condition

One potentially interesting lifetime histogram is where only one electron detection is allowed within a specified time interval around the muon arrival. That is, if two or more electrons can be associated with a μ SC hit, it is rejected. An example of the lifetime histogram which results from enforcement of this “unique electron” condition over the time interval $[-5, 25] \mu\text{s}$ is presented in Figure 48. Fits to the unique electron

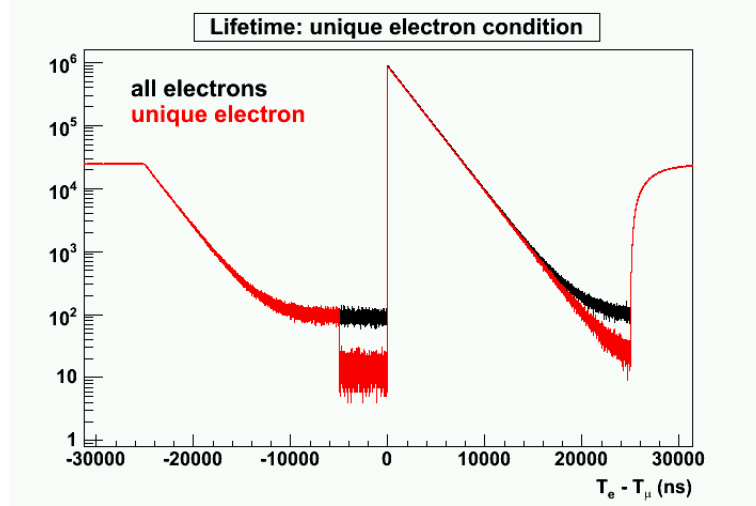


Figure 48: Pileup-protected lifetime histograms with and without the unique electron condition. The black spectrum involves all detectors, with a $25 \mu\text{s}$ pileup protection interval, an impact cut $b \leq 120 \text{ mm}$, and all scatters removed. The red spectrum is the same, but with a unique electron condition enforced over the interval $[-5, 25] \mu\text{s}$.

histograms are perhaps best compared to the cosmics-bin-error-adjusted fits, since in both cases steps have been taken to avoid effects from double-counting. The fitted rates for the $25 \mu\text{s}$ pileup-protected Prod-50 lifetime histograms, involving all detectors, using my standard fiducial, $b \leq 120 \text{ mm}$, and scatter cuts, for fit interval $[110, 24900] \text{ ns}$ (with 32-rebin boundary at 105 ns) are given in Table 4. If we compare the unique

ePC table	lifetime histogram	$\lambda \text{ (s}^{-1}\text{)}$	$\sigma_\lambda \text{ (s}^{-1}\text{)}$	χ^2/d	$\sigma (\chi^2/d)$
cathode-AND	cosmics-error-adjusted	455,425.04	12.86	0.965	0.057
	unique electron	455,428.01	12.76	0.946	0.057
cathode-OR	cosmics-error-adjusted	455,433.17	12.33	0.998	0.057
	unique electron	455,433.55	12.25	0.931	0.057

Table 4: Comparison of fitted rates to the cosmics-error-adjusted and “unique electron” lifetime histograms.

electron lifetime fits to the cosmics-corrected lifetime fits, we can see that the unique electron histogram’s χ^2 values are somewhat better—perhaps due to the lower background level? Regardless, the difference in rates between the two types of lifetime histograms is reasonably close to the statistically allowed variation (the cathode-OR fits are nearly identical), and the unique electron fit thus serves as a nice consistency check on the standard results. I see no reason to use it beyond this point.

6.10 Final pre-corrected muon disappearance rate

For my pre-corrected muon disappearance rate in hydrogen, I chose to fit lifetime histograms that had been created under the following conditions:

- The muon table came from muSC/muPC1/TPC coincidences, using the muSC+muPC1XY-OR table for 25 μs pileup protection.
- The electron tables came from ePC1/ePC2/eSC coincidences; I used tables for both the ePC cathode-AND and cathode-OR treatments, since I have not yet found a way to choose one over the other.
- I used my “standard” fiducial TPC cut, as described in Section 6.6.
- I enforced a 120 mm impact cut.

I fit these lifetime histograms under the following conditions:

- I rebinned by 32 (to obtain bin widths of 40 ns, the coarse CAEN clock period), with the rebin boundary set at 105 ns.
- I fit over the range [110,24900] ns.
- In my fits I assumed that cosmics accounted for half of the observed background, and adjusted the bin errors accordingly.

I performed such fits for both the Prod-50 data (clean fill production data with the TPC operating at 5 kV) and the CalibNat data (calibration fill that was doped with natural hydrogen to obtain a concentration $c_d \approx 120$ ppm), so that I can later perform the deuterium zero-extrapolation correction. My uncorrected fit results are given in Table 5.

Fill	ePC table	λ (s^{-1})	σ_λ (s^{-1})	χ^2/d
Prod-50	cathode-AND	455,428.06	12.86	0.964
	cathode-OR	455,436.29	12.33	0.998
CalibNat	cathode-AND	456,193.79	64.47	1.099
	cathode-OR	456,180.37	61.80	1.111

Table 5: *Fitted rates before corrections. The fits’ residuals are all nicely flat.*

6.11 A note on statistics

According to my May 2005 analysis presentation at UIUC, cuts due to DAQ and detector misbehavior (e.g. block duplication, CAEN error, CAEN rollover error, COMP error, TDC400 error, muSC matching error, eSC CAEN/COMP matching error, and global sparks) amount to 2% of the total data. Far more significant cuts to the statistics arise from pileup protection (31% retention for a 25 μs interval), fiducial TPC cuts (59% retention), electron detector efficiencies (83% retention), and the electron detector’s limited solid angle of acceptance (75% retention). Ultimately, once we have also taken into account run selection, impact parameter cuts, etc., the remaining number of decay events is $N \approx 1.6 \times 10^9$.

7 Lifetime corrections

7.1 Muon scatters

Muon scatter events systematically pull the fitted rate up and away from its correct value. The reason is that although the muon might appear to have stopped in the hydrogen gas inside the TPC fiducial volume, in reality the muon sometimes scatters into a surrounding high- Z material where it is subjected to a far higher capture rate. This behavior is clearly illustrated in Figure 49, where the fitted rates for the scattered muons are clearly much higher than for the majority of fiducial stops, which is consistent with the effects expected from muon scatter into high- Z materials.

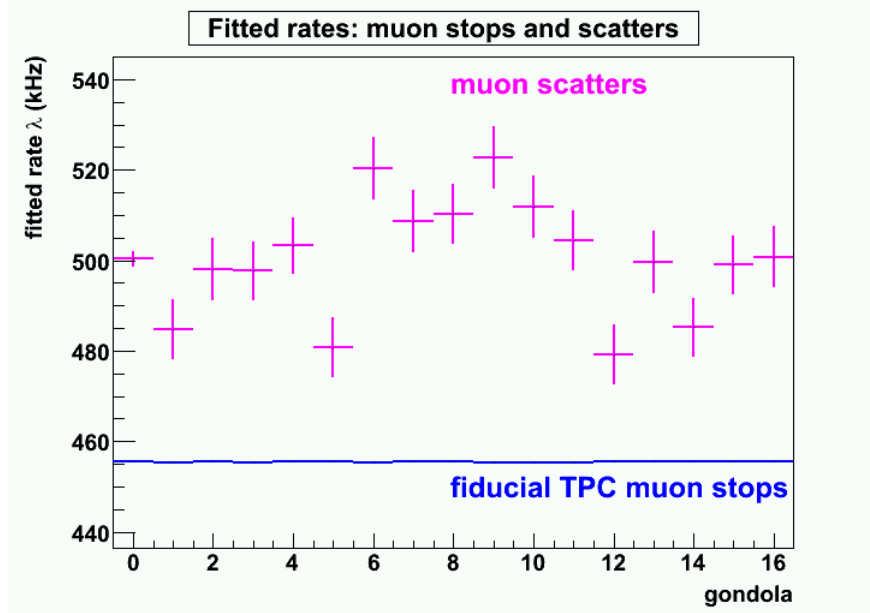


Figure 49: The per-gondola fitted rates for scattered muons and for all fiducial muon stops. Note that the fitted rates for the scattered muons is much higher, which is consistent with stops in high- Z materials.

When I remove muon scatter events from the lifetime histograms, the fitted rate shifts slightly downward as anticipated, usually by 3–4 Hz. However, I must also consider the fact that my software is most likely not catching all of the muon scatters; that is, some fraction of them go unidentified. In this section I will demonstrate the details of the muon scatter correction process using the Prod-50 cathode-OR data. I should mention that part of the reason that we ignored the Prod-48 data at TPC voltage 4817 V is because of the reduced track sensitivity makes the identification of scatters more difficult.

First, consider the muon scatter statistics from the Prod-50 clean fill and CalibNat data in Table 6, where the μ PC1XY-OR condition was used in pileup protection. For now let us focus on the clean fill numbers. As can be seen, TLS scatters comprise the majority of identified scatters, accounting for roughly 75% of scatter events, compared to 25% for the MWPC scatters. Inspection of the corresponding lifetime histogram fit results in Table 7 confirms that the fitted decay rate offsets from the two types of scatters are of similar proportion (i.e. TLS scatters account for about 2.2 Hz of the total 3 Hz rate shift), which is to be expected if the scattering effects are small and therefore linear in nature.

For the cathode-OR data, the rate difference due to scatters can be obtained by subtracting the study16

Fill	Quantity	Count	Fraction ($\times 10^{-4}$)
Prod-50	muon stops	2,378,668,731	—
	TLS scatters	240,717	1.012(2)
	MWPC scatters	82,571	0.347(1)
	total scatters (union)	288,826	1.214(2)
CalibNat	muon stops	95,241,158	-
	TLS scatters	3,237	0.339(6)
	MWPC scatters	2,542	0.266(5)
	total scatters (union)	5,365	0.563(8)

Table 6: *The muon scatter statistics for the Prod-50 clean fill and CalibNat (natural hydrogen) fill. I believe that the CalibNat TLS scatter fractions are lower because of the reduced TPC operating voltage during that fill (4.8 kV instead of the 5.0 kV in the clean fill).*

rate from the study13 rate in Table 7. We can place an error on the rate difference by using the formulas in Appendix D, which yields (for the Prod-50 data)

$$\Delta\lambda_{\text{scatter}} = -3.12 \pm 0.20 \text{ Hz} . \quad (5)$$

However, based upon my MC studies, I have good reason to believe that I’m only catching somewhere between 63% and 75% of all muon scatter events. Thus, the rate shift should be slightly larger than what is given in Equation 5. If we assume that I catch 69% of all scatters (the SRIM results suggest that the total scatter fraction is $\sim 1.762 \times 10^{-4}$), then the rate shift should be

$$\Delta\lambda_{\text{scatter}} = -4.52 \pm 0.20 \pm 0.5 \text{ Hz} . \quad (6)$$

Note that I have introduced an additional error term to account for the uncertainty in my scatter identification efficiency.

By using similar reasoning for the cathode-AND data, and by applying the same technique to the CalibNat fill, we can correct the results in Table 5; the results are given in Table 8. Notice that all of the scatter rate corrections in Table 8 are of roughly the same size—a reassuring feature.

Fill	ePC table	condition	study#	λ (s ⁻¹)	σ_λ (s ⁻¹)	χ^2/d
Prod-50	cathode-AND	scatters included	study2	455,428.06	12.86	0.964
		scatters only	study3	499,559.40	1,722.95	1.282
		all scatters removed	study4	455,425.04	12.86	0.965
	cathode-OR	scatters included	study12	455,436.29	12.33	0.998
		scatters only	study13	500,343.08	1,678.34	1.348
		TLS scatters removed	study14	455,433.56	12.34	0.998
		MWPC scatters removed	study15	455,435.34	12.33	0.998
		all scatters removed	study16	455,433.17	12.34	0.998
CalibNat	cathode-AND	scatters included	study2	456,193.79	64.47	1.099
		scatters only	study3	483,183.00	16,267.96	0.906
		all scatters removed	study4	456,192.34	64.47	1.101
	cathode-OR	scatters included	study12	456,180.37	61.80	1.111
		scatters only	study13	480,957.35	15,332.02	0.800
		TLS scatters removed	study14	456,179.40	61.80	1.112
		MWPC scatters removed	study15	456,179.62	61.80	1.111
		all scatters removed	study16	456,178.95	61.80	1.112

Table 7: Fitted rates for various clean fill and natural hydrogen fill lifetime histograms, under different muon scatter inclusion/exclusion conditions.

Fill	ePC table	Correction	Corrected rate	
		$\Delta\lambda_{\text{scatter}}$ (s ⁻¹)	λ (s ⁻¹)	σ_λ (s ⁻¹)
Prod-50	cathode-AND	-4.38 ± 0.54	455,423.68	12.87
	cathode-OR	-4.52 ± 0.54	455,431.77	12.34
CalibNat	cathode-AND	-4.53 ± 1.35	456,189.26	64.48
	cathode-OR	-4.44 ± 1.32	456,175.93	61.81

Table 8: The μ scatter corrections, and resulting rates.

Fill	μ stops	high- Z captures					
		EH			EVH		
		count	control	Yield (ppm)	count	control	Yield (ppm)
Prod-50	1,799,056,105	30,477	4146	14.64(9)	24,405	1,208	12.89(8)
CalibN2	132,859,288	103,700	315	778.2(2.4)	103,063	114	774.9(2.4)
CalibNat	108,492,995	5,823	216	51.7(7)	5,368	97	48.6(7)

Table 9: *High- Z impurity capture results gleaned from the TPC data by the Berkeley analysis software. The capture counts are from time-delayed events to the right of the muon stop, while the “control” events are from a mirror image search to the left, in the time preceding the muon stop. EH capture events required an EH pixel; EVH capture events are a subset of the EH events where at least one EVH threshold pixel was present. Both of the capture yields were calculated according to the formula $Yield = (count - control) / \mu stops$.*

7.2 High- Z impurities

Our strategy for dealing with high- Z impurities is to reduce their concentration levels as much as possible during the run, and then in the offline analysis to use calibration data to correct for any effects from the residual clean-fill contamination. The high- Z correction is far from trivial, however, and is arguably our most problematic correction. To date we have a plethora of conflicting data points and analyses, and thus far we have not succeeded in reconciling all of the available information into a completely consistent, definitive picture of impurity behavior—our understanding is still incomplete. Fortunately, we have good reason to believe that the change to the decay rate scales with the TPC capture yield, and in nearly the same way for all likely contaminating elements. As long as we can establish the relationship between the TPC capture yield and changes to the decay rate, it is not essential to ascertain the concentration levels with absolute certainty. Given our current level of ignorance, a little art and some plausibility arguments are inevitable, but we should nevertheless be able to reach a relatively robust value for the correction.

There are two basic formulas that govern our determination of the Run8 high- Z impurity corrections:

$$\Delta\lambda^{\text{exp}} = Y \sum_Z \frac{w_Z}{\epsilon_Z} \left(\frac{\Delta\lambda}{Y} \right)_Z^{\text{th}} \quad (7)$$

$$c_Z = \frac{Y_Z}{\epsilon_Z} \left(\frac{c}{Y} \right)_Z^{\text{th}}. \quad (8)$$

Equation 7 describes how the shift to the decay rate, $\Delta\lambda$, scales with the observed high- Z impurity capture yield Y . This linear relationship has been demonstrated using Monte Carlo to remain valid up to the highest concentrations we have worked with in MuCap (around 10 ppm). The summation in Equation 7 is performed over all of the high- Z species that are present in the hydrogen gas. Here w_Z is a weighting factor ($w_Z = Y_Z/Y$, so $\sum_Z w_Z = 1$), and ϵ_Z is the detection efficiency ($\epsilon_Z \leq 1$). The quantity $\left(\frac{\Delta\lambda}{Y} \right)_Z^{\text{th}}$ is the rate vs. yield scaling factor which we have calculated based upon values in the literature; Peter Kammel and myself have recently made an effort to finalize their values, although some further work remains. Equation 8 describes the simple linear relationship between the individual high- Z concentrations and yields.

The Run8 impurity yields are provided in Table 9, and the corresponding time distributions are plotted in Figure 50. An example of a fit to a capture time distribution is given in Figure 51. There is a wide variety of information from Run8, Run9, and Run10 studies, and unfortunately much of it is conflicting. However,

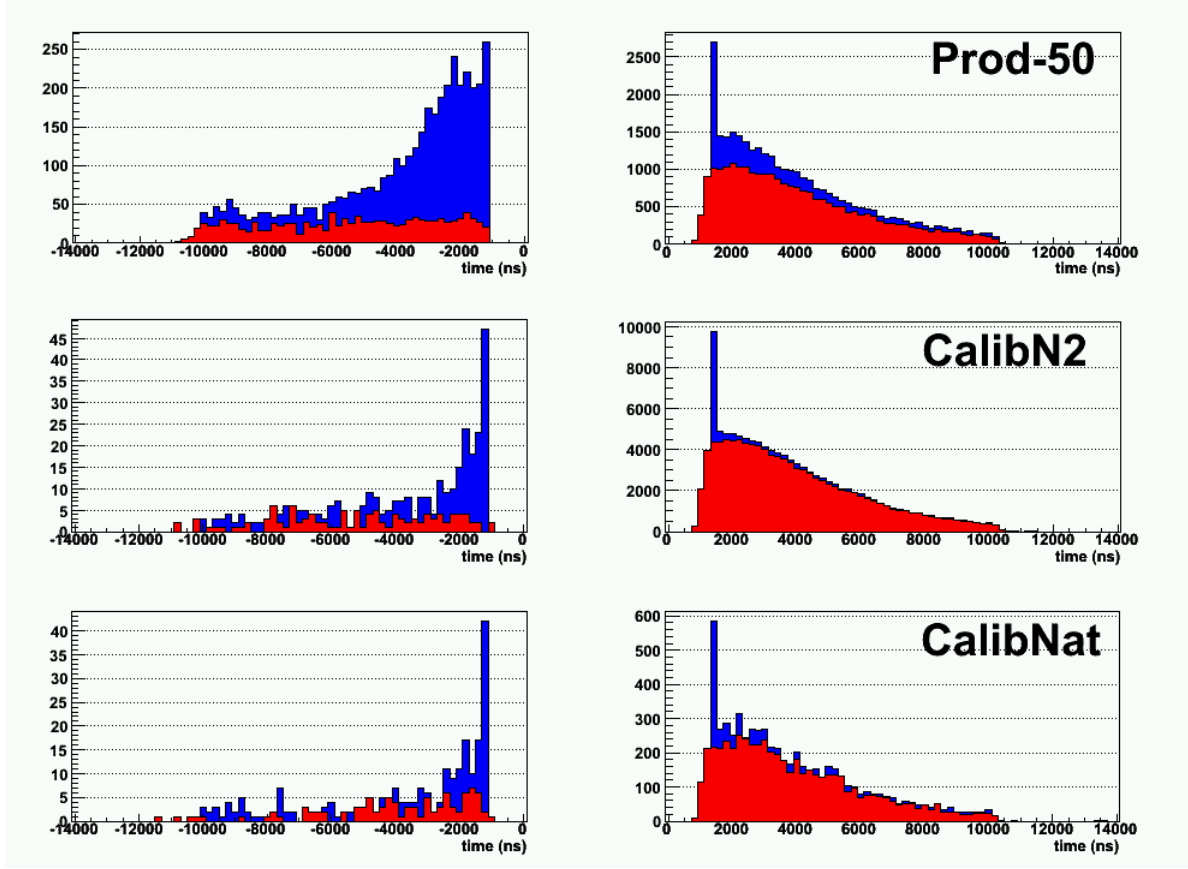


Figure 50: *High- Z impurity capture time distributions gleaned from the TPC data by the Berkeley analysis software. EH captures are shown in blue, and EVH captures are shown in red. The left column contains the “control” distributions, while the right column contains the rightward capture distributions. Note that the EVH distributions exhibit lower background and noise than the EH distributions—especially at times close to the muon stop time—which is to be expected. Although these time distributions all exhibit the correct basic shape, fits (see Figure 51) have not yielded results consistent with other knowledge about the impurity concentrations.*

we currently believe that nitrogen and oxygen (in the form of water) comprised nearly all of the high- Z impurities in our hydrogen target. In this case, Equation 7 becomes

$$\Delta\lambda = Y \left[\frac{w_N}{\epsilon_N} \left(\frac{\Delta\lambda}{Y} \right)_N^{\text{th}} + \frac{w_O}{\epsilon_O} \left(\frac{\Delta\lambda}{Y} \right)_O^{\text{th}} \right]. \quad (9)$$

We must now determine values for all of the parameters in the equation above.

First, comparison of the high- Z capture yields and the (somewhat questionable) gas chromatography measurements of samples taken during the Run8 clean fill tells us that we can reasonably assume that $w_N = 0.1$ and $w_O = 0.9$, which is consistent with more recent humidity studies from Runs 9 and 10 that suggest that water is the dominant impurity in outgasing. The same values are most likely applicable to the CalibNat fill.

Let us next turn our attention towards determining the detection efficiencies for nitrogen and oxygen. This is a relatively simple task in the case of nitrogen, because we performed a nitrogen-doped calibration

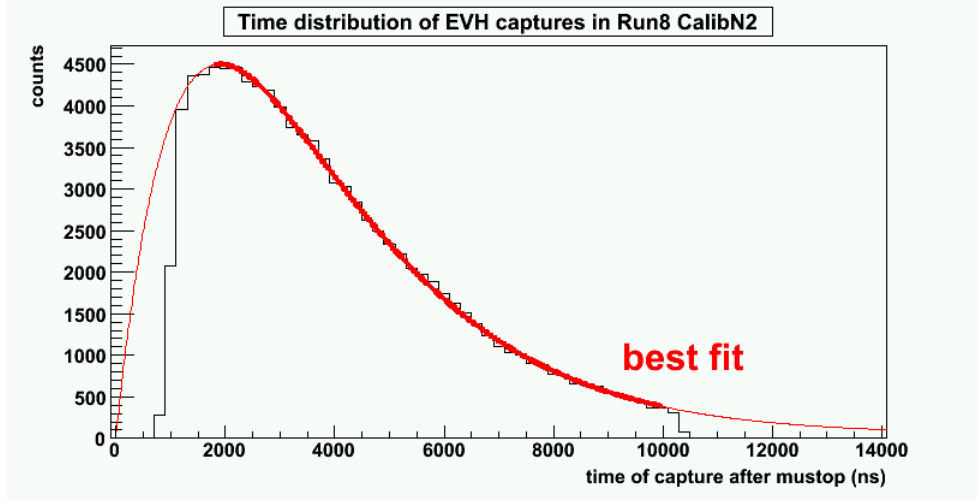


Figure 51: Example of a fit to a high- Z impurity capture time distribution—in this case, the CalibN2 EVH distribution. For this fill (and for most of the others), the time distribution has the correct shape and decent fits can be obtained, but the resulting numbers are mysteriously inconsistent with other knowledge. This is especially frustrating with the Run8 CalibN2 fill, which had such a high level of nitrogen that we expected the time distributions to give a straightforward confirmation of the impurity composition.

fill during Run8 (CalibN2). For this fill, nitrogen was by far the dominant impurity, so we can ignore any contributions from oxygen and write

$$\Delta\lambda \approx \frac{Y}{\epsilon_N} \left(\frac{\Delta\lambda}{Y} \right)_N^{\text{th}}. \quad (10)$$

Let us rearrange this formula to solve for ϵ_N^{EVH} :

$$\epsilon_N^{\text{EVH}} = \frac{\left(\frac{\Delta\lambda}{Y} \right)_N^{\text{th}}}{\left(\frac{\Delta\lambda}{Y_N^{\text{EVH}}} \right)}.$$

If we now plug in the numbers

$$\left(\frac{\Delta\lambda}{Y} \right)_N^{\text{th}} = 1.820 \text{ ppm of } \lambda_0/\text{ppm} = 0.828 \text{ s}^{-1}/\text{ppm}$$

$$\begin{aligned} \Delta\lambda &\approx \lambda_{\text{CalibN2}} - \lambda_{\text{Prod-50}} \\ &= 456,425.24(55.41) \text{ s}^{-1} - 455,436.29(12.33) \text{ s}^{-1} \\ &= 988.95 \pm 56.77 \text{ s}^{-1} \end{aligned}$$

$$\begin{aligned} Y_N^{\text{EVH}} &\approx Y_{\text{CalibN2}}^{\text{EVH}} - Y_{\text{Prod-50}}^{\text{EVH}} \\ &= 774.9(2.4) \text{ ppm} - 12.89(8) \text{ ppm} \\ &= 762.0(2.4) \text{ ppm} \end{aligned}$$

then we get a detection efficiency of

$$\epsilon_N^{\text{EVH}} \sim 64\% \pm 3\%(?) . \quad (11)$$

which is in good agreement with previous determinations of the TPC EVH nitrogen capture detection efficiency from Run7. (This result holds for both the cathode-AND and the cathode-OR data.) It should be noted that the maximum possible detection efficiency is $\sim 85\%$, based upon the limited time range over which the TPC impurity search is conducted. We can in turn plug the result from Equation 11 into Equation 8 to determine the nitrogen concentration:

$$\begin{aligned} c_N &= \frac{Y^{\text{EVH}}}{\epsilon_N^{\text{EVH}}} \left(\frac{c}{Y} \right)_N^{\text{th}} \\ c_N &= \frac{762 \text{ ppm}}{0.64} \left(\frac{1}{107.6 \text{ ppm/ppm @ } c = 10 \text{ ppm}} \right) \\ c_N &\approx 11.07 \pm \sim 1.0 \text{ ppm} . \end{aligned} \quad (12)$$

This concentration result agrees very nicely with Dr. Claude Petitjean's and Dr. Malte Mildebrandt's estimate at the time of the fill (using volumetric arguments)

$$c_N^{\text{estimate}} = 11.00 \pm 0.22 \text{ ppm} , \quad (13)$$

so I think we can place some confidence in our value for ϵ_N^{EVH} .

In contrast, the determination of the oxygen detection efficiency is far more dubious. Any uncertainty in an efficiency is especially dangerous, since ϵ appears in the denominator of the correction terms in Equation 9 and thus has the potential to blow up the value of the correction $\Delta\lambda$. An oxygen-doped calibration fill was attempted during Run8, but the oxygen quenched the TPC charge to such an extent that the data was useless. Nevertheless, I believe that we can use the EH and EVH capture data in concert to establish a lower bound on the oxygen efficiency ϵ_O . My reasoning is as follows: Nitrogen captures release 300 keV of recoil energy, while oxygen captures release 260 keV. Meanwhile, the TPC EH threshold is typically set around 65 keV, while the TPC EVH threshold setting is around 260 keV—i.e. in the vicinity of the oxygen capture energy. Thus we can expect the EH and EVH yields to be similar for nitrogen, but that there might be differences between the EH and EVH yields for oxygen, depending upon where the EVH threshold was set, exactly. This hypothesis is supported by inspection of the Prod-50 and CalibN2 time distributions in Figure 50. The fact that the EH and EVH yields are not too far apart in the Prod-50 clean fill data leads me to suspect that the oxygen detection efficiency must not be too different from the nitrogen detection efficiency. We can formulate this mathematically in the following manner:

$$\begin{aligned} Y^{\text{true}} &= Y^{\text{true}} \\ Y^{\text{EH}} \left[\frac{w_N^{\text{EH}}}{\epsilon_N^{\text{EH}}} + \frac{w_O^{\text{EH}}}{\epsilon_O^{\text{EH}}} \right] &= Y^{\text{EVH}} \left[\frac{w_N^{\text{EVH}}}{\epsilon_N^{\text{EVH}}} + \frac{w_O^{\text{EVH}}}{\epsilon_O^{\text{EVH}}} \right] . \end{aligned}$$

Now, the nitrogen efficiency should be the same for both EH and EVH,

$$\epsilon_N^{\text{EH}} = \epsilon_N^{\text{EVH}} \equiv \epsilon_N = 0.64 ,$$

and we can reasonably assume that the oxygen EH efficiency shares the same upper limit,

$$\epsilon_O^{\text{EH}} \approx \epsilon_N = 0.64 .$$

Furthermore, the element weights are also basically the same for both situations,

$$w_N^{\text{EH}} \approx w_N^{\text{EVH}} \equiv w_N \approx 0.1 \quad (14)$$

$$w_O^{\text{EH}} \approx w_O^{\text{EVH}} \equiv w_O \approx 0.9 . \quad (15)$$

We can thus solve for the oxygen EVH detection efficiency,

$$\epsilon_O^{\text{EVH}} = \frac{Y^{\text{EVH}} w_O}{(Y^{\text{EH}} - Y^{\text{EVH}}) \left(\frac{w_N}{\epsilon_N} \right) + Y^{\text{EH}} \left(\frac{w_O}{\epsilon_N} \right)} = \frac{Y^{\text{EVH}} w_O \epsilon_N}{(Y^{\text{EH}} - Y^{\text{EVH}} w_N)} \quad (16)$$

$$= 0.547 . \quad (17)$$

So the EVH oxygen detection efficiency appears to be roughly 55%, compared to 64% for nitrogen. This result seems perfectly reasonable.

Last, we need to determine the appropriate $(\Delta\lambda/Y)^{\text{th}}$ values to use in the corrections. These are given in the table below, and are based upon recent work by Peter Kammel and myself.

Fill	$(\Delta\lambda/Y)^{\text{th}}$ (s^{-1}/ppm)	
	Nitrogen	Oxygen
Prod-50	0.828 ($\pm 0.015?$)	0.805 ($\pm 0.015?$)
CalibNat	0.855 ($\pm 0.015?$)	0.832 ($\pm 0.015?$)

Table 10: The λ/Yield ratios for nitrogen and oxygen in the Prod-50 and CalibNat fills. The values are different because the two fills had different levels of high- Z and deuterium impurities.

Note that the Prod-50 and CalibNat fills require slightly different $(\Delta\lambda/Y)^{\text{th}}$ values because the high- Z and deuterium concentrations were different for the two fills, and $(\Delta\lambda/Y)^{\text{th}}$ is in principle a function of both.

We are now in a position to perform the high- Z impurity correction, by plugging all of the relevant numbers into Equation 9, and applying the results to the rate in Table 8 to get:

Fill	ePC table	Correction	Corrected rate	
		$\Delta\lambda_{\text{high-}Z}$ (s^{-1})	λ (s^{-1})	σ_λ (s^{-1})
Prod-50	cathode-AND	-18.69 ± 1.60	455,404.99	12.97
	cathode-OR	-18.69 ± 1.60	455,413.08	12.44
CalibNat	cathode-AND	-72.66 ± 6.22	456,116.60	64.78
	cathode-OR	-72.66 ± 6.22	456,103.27	62.12

Table 11: The high- Z corrections, and the resulting rates after the μ scatter and high- Z corrections. I obtained the error on the high- Z correction via Monte Carlo simulations with a ROOT macro, using errors $\sigma_{w_Z} = 0.05$, $\sigma_{\epsilon_Z} = 0.03$, and $\sigma_{(\Delta\lambda/Y)} = 0.015$.

7.3 Deuterium

The presence of deuterium in our hydrogen gas target increases the muon disappearance rate, away from its (desired) value in protium. This occurs because μd atoms rapidly diffuse away from the muon stopping point (due to the Ramsauer-Townsend minimum in the $\mu d + H_2$ scattering cross-section), which has two effects: (1) if any cuts are made on the muon/electron impact parameter, there is a chance that the μd atom will have diffused outside of the cut radius at the time of the muon's decay, thereby artificially raising the observed rate; and (2) the μd atom can diffuse into surrounding high- Z wall materials, where it is subjected to a much higher capture rate than in hydrogen gas. The MuCap strategy for dealing with deuterium is to suppress the deuterium level as much as possible during the run, and then to use zero-extrapolation techniques to correct for systematic effects from any remaining contamination. Using Monte Carlo simulations, Brendan Kiburg has demonstrated that deuterium-related effects on the fitted rate remain linear all the way up to $c_d = 140$ ppm, so the zero-extrapolation procedure is valid for the deuterium concentrations under consideration in MuCap.

To perform a zero-extrapolation correction, we need to have recorded data at two different deuterium concentrations. The correction procedure then follows a rather simple formula,

$$\lambda_1 = \frac{c\lambda_2 - \lambda_3}{c - 1}, \quad (18)$$

which is just the y-intercept of the line passing through the two data points, as illustrated in Figure 52. The point $(1, \lambda_2)$ here corresponds to a “clean” (i.e. low c_d) production measurement with high statistics,

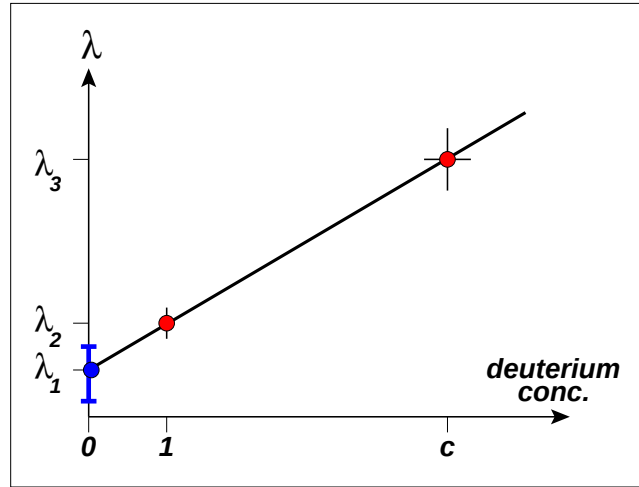


Figure 52: Illustration of the zero-extrapolation correction procedure for deuterium. The point $(1, \lambda_2)$ corresponds to the high-statistics clean fill, while the point (c, λ_3) corresponds to a deuterium-doped calibration fill. Note that the error bar is larger for the deuterium-doped fill because it is generally of shorter duration and therefore contains fewer statistics.

while point (c, λ_3) is from a deuterium-doped calibration measurement of shorter duration and therefore with lower statistics; $c = c_3/c_2$ is the ratio of the the deuterium concentrations in the two measurements.

Estimating the error on λ_1 is slightly more complicated. According to the general error propagation law,

$$\begin{aligned}\sigma_{\lambda_1}^2 &= E_{\lambda_2}^2 + E_{\lambda_3}^2 + E_c^2 + E_{\lambda_3 c}^2 \\ &= \sigma_{\lambda_2}^2 \left(\frac{c}{c-1} \right)^2 + \sigma_{\lambda_3}^2 \frac{1}{(c-1)^2} + \sigma_c^2 \frac{(\lambda_3 - \lambda_2)^2}{(c-1)^4} + 2\sigma_{\lambda_3 c}^2 \frac{(\lambda_2 - \lambda_3)}{(c-1)^3}.\end{aligned}\quad (19)$$

The errors σ_{λ_1} and σ_{λ_2} are represented by the vertical error bars in Figure 52, which primarily come from statistics but which also receive contributions from the preceding muon scatter and high- Z impurity corrections. Notice that each of the λ errors is weighted in Equation 19 according to its participation in Equation 18. The quantity σ_c is the horizontal error bar on point (c, λ_3) , and corresponds to the error in our estimation of the relative deuterium concentrations of the two data sets. Finally, $\sigma_{\lambda_3 c}$ is the covariant error that accounts for any correlations between λ_3 and c . In principle, we use the same data to obtain both λ_3 and c , so this term is probably nonzero. However, it's probably small, and I don't know how to go about calculating it, so I will ignore it. Besides, it has a negative sign, which means that excluding it will give an conservative upper bound on the error.

With the basic deuterium correction formalism in place, let us turn our attention towards how to apply it to the 2004 data. During Run8 we performed two deuterium-doped calibration measurements: one with $c_d \sim 20$ ppm (CalibD2) and one with $c_d \sim 120$ ppm (CalibNat, where “Nat” refers to the fact that the gas was doped with natural hydrogen, $c_d \sim 140$ ppm). Both of the calibration fills suffered from elevated high- Z impurity levels, and from a reduced TPC operating voltage of 4.8 kV. Although the CalibNat statistics are smaller than the CalibD2 statistics ($N_{\text{CalibNat}} \sim 6.2 \times 10^7$ vs. $N_{\text{CalibD2}} \sim 21.3 \times 10^7$), the CalibNat fill is more attractive because its higher deuterium concentration gives it a longer lever arm in the zero-extrapolation operation, thus yielding a smaller error on the corrected result by ~ 1 Hz.

The next step, then, is to determine the deuterium concentration ratio c between the Run8 CalibNat fill and the Prod-50 clean fill. There are two means of doing this: (1) via external measurements of hydrogen samples, and (2) from the data itself, by looking at the decay rate vs. impact parameter distributions for the two fills. Over the past two years, Dr. Claude Petitjean has coordinated a number of deuterium concentration measurements of both liquid and gaseous hydrogen samples, at a variety of locations and using different methods—e.g. the mass spectrograph at the PSI Lab of Atmospheric Chemistry, courtesy of Dr. Mathias Saurer; the Tandem isotope spectrograph at ETH, courtesy of Dr. Max Doebeli; the Ioffe Institute; etc. The results are too extensive to recount here in detail; Claude is best qualified to comment on the present state of our knowledge of deuterium concentrations, and can hopefully provide the collaboration with an up-to-date assessment of the situation at our October 2006 collaboration meeting. However, my understanding is that the many measurements point towards the following Run8 concentrations:

$$c_d (\text{Prod-50}) = 1.44 \pm 0.13 \text{ ppm} \quad (20)$$

$$c_d (\text{CalibNat}) = 117.6 \pm ? \text{ ppm} \quad (21)$$

I am not aware of any error bars on the CalibNat deuterium measurement, and I don't believe that any measurements have been reported for the Run8 CalibD2 fill.

The other method for determining the concentration ratio relies upon the fact that the fitted decay rate vs. impact cut distribution scales with deuterium concentration, as illustrated in Figure 53. The shape of these distributions is due to the fact that, for smaller impact cuts, the μd atom is more likely to have diffused outside the cut radius, thereby mimicking capture and increasing the effective disappearance rate. Unfortunately, the extraction of the deuterium ratio from the distributions in Figure 53 is far from straightforward, because each of them also contains a small, similarly-structured component which arises from a combination

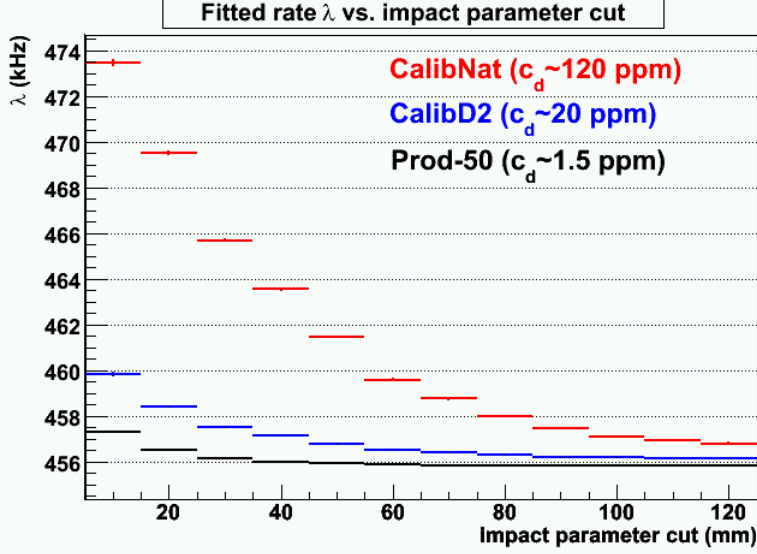


Figure 53: The decay rate vs. impact parameter distributions for different deuterium concentrations. Note that each of the distributions has roughly the same shape, but that they scale with deuterium concentration (and also shift upwards). Note that the structure visible in the Prod-50 distribution, where the deuterium concentration was very low, is actually due to the μp -diffusion-related effect that was described earlier and illustrated in Figure 30. It is also present in the CalibD2 and CalibNat distributions, but is obscured by the deuterium-related structure.

of μp diffusion plus electron scattering from the pressure vessel. However, in March–April 2006, shortly after our collaboration meeting at UC Berkeley, Steve Clayton performed a complex study of the subject, and, using “annulus” impact parameter cuts rather than the conventional “disk” impact cuts, he was able to extract the following deuterium concentration ratio from his data analysis [10]

$$c^{-1} = \frac{c_d (\text{Prod-50})}{c_d (\text{CalibNat})} = 0.0123 \pm 0.0010 . \quad (22)$$

If we use the external CalibNat measurement of 117.6 ppm as a fixed reference point, Steve’s extracted ratio gives

$$\begin{aligned} c_d (\text{Prod-50}) &= c^{-1} \times c_d (\text{CalibNat}) \\ &= 1.44 \pm 0.12 \text{ ppm} , \end{aligned} \quad (23)$$

in excellent agreement with Claude’s reported result in Equation 20 from external measurements. Thus the two methods for determining the Run8 deuterium concentration ratio appear to yield completely consistent results.

Using Steve’s deuterium ratio in Equation 22, let us now calculate the deuterium correction for the cathode-AND and cathode-OR data. First, let us invert Equation 22 to get

$$c = \frac{c_d (\text{CalibNat})}{c_d (\text{Prod-50})} = 81.3 \pm 7.3 . \quad (24)$$

Plugging this into Formulas 18 and 19, along with the λ values obtained following the high- Z correction (Table 11), we get:

ePC table	Correction $\Delta\lambda_{\text{deuterium}} (\text{s}^{-1})$	Corrected rate $\lambda (\text{s}^{-1}) \quad \sigma_\lambda (\text{s}^{-1})$	
cathode-AND	-8.88 ± 1.13	455,396.13	13.02
cathode-OR	-8.61 ± 1.11	455,404.48	12.49

Table 12: The deuterium corrections, and the resulting rates after the μ scatter, high- Z , and deuterium corrections. I obtained the error on the deuterium correction via Monte Carlo simulations with a ROOT macro.

7.4 Muon diffusion + pressure vessel scattering + impact cut

In the course of Steve’s studies to determine the deuterium concentration ratios among the Run8 fills [10], he had to simulate the expected rate offset due to the coupling between μp diffusion and electron scattering from the pressure vessel (see Section 5.5 and accompanying Figure 30). According to his simulations, the rate increase due to a 120 mm impact cut is

$$\Delta\lambda(b \leq 120 \text{ mm}) = 2.8 \pm ? \text{ Hz} . \quad (25)$$

At present I do not know what error to assign to this correction—perhaps 0.5 Hz is safe? Let us assume so, and subtract this rate offset from the deuterium-corrected values in Table 12 to get:

ePC table	Correction $\Delta\lambda_{\text{impact cut}} (\text{s}^{-1})$	Corrected rate $\lambda (\text{s}^{-1}) \quad \sigma_\lambda (\text{s}^{-1})$	
cathode-AND	-2.8 ± 0.5	455,393.33	13.03
cathode-OR	-2.8 ± 0.5	455,401.68	12.50

Table 13: The μp diffusion + scattering + impact cut correction, and the resulting rates after the μ scatter, high- Z , deuterium, and μp diffusion corrections.

7.5 Detector inefficiencies

In principle, inefficiencies in the muon and electron detectors can affect the observed rate. I performed extensive Monte Carlo studies on inefficiencies in the muon detectors, and confirmed the possibility for both time-independent and time-dependent inefficiencies to affect our results. However, even when using conservative estimates, the predicted effect from muon detector inefficiencies was relatively small (~ 4 Hz), and my Monte Carlo did not simulate tracking (i.e. impact parameter cuts) into account, which would presumably improve the situation. Indeed, recent studies performed by Steve using the Run8 data have indicated that, under the conditions of our analysis (i.e. $\mu\text{SC}+\mu\text{PC1}$ pileup protection, 120 mm impact cut), the entrance muon detector inefficiencies have no discernable effect on the fitted rate.

Regarding the electron detectors, Steve and I have both observed a lack of sensitivity in the fitted rate to changes in the ePC artificial deadtime width, from a range of 250–2000 ns. It seems unlikely, then, that the larger-than-expected difference between the cathode-AND and cathode-OR fitted rates in my analysis can be attributed to inefficiencies in the electron detectors. Nonetheless, until we find the culprit, I will leave that possibility open.

In light of the preceding information, I have elected not to perform any corrections for detector inefficiencies.

8 Conclusions and remaining work

8.1 Final blinded muon disappearance rates

My preliminary, blinded results for the effective muon disappearance rate in hydrogen are:

$\lambda_{\text{Run8}}(\text{cathode-AND}) = 455,393.33 \pm 13.03 \text{ s}^{-1}$	(26)
---	------

$\lambda_{\text{Run8}}(\text{cathode-OR}) = 455,401.68 \pm 12.50 \text{ s}^{-1}$	(27)
--	------

At present, the reasons for the difference between these two results remains unknown, so it's difficult to say how they should be reconciled. My gut instincts lead me to suspect that the cathode-OR rate is more reliable because it is more consistent with the eSC-only lifetime fit result, and because cathode-OR should be a more “forgiving” detector condition—but that is only speculation.

8.2 Post-unblinding steps

Up to this point in our data analysis, we have behaved as though the DAQ electronics frequency were exactly 100 MHz. In reality, of course, the DAQ frequency was slightly detuned from this value to blind our analyses and prevent us from inadvertently converging towards the expected result. Dr. Malte Hildebrandt was given responsibility for blinding the clock; having been asked to detune the clock frequency within 1% of 100 MHz, he selected a frequency somewhere in the range 100.0–100.1 MHz. Once this secret frequency setting has been revealed—hopefully at the October 2006 collaboration meeting—the formula for unblinding the fitted decay rate is

$$\lambda_{\text{Run8}}^{\text{unblinded}} = \lambda_{\text{Run8}}^{\text{blinded}} \left(\frac{f_{\text{DAQ}}}{100 \text{ MHz}} \right), \quad (28)$$

which holds true regardless of the direction of the clock detuning.

After the unblinding, the following steps must be performed to obtain our final result for the singlet capture rate Λ_S :

1. We need to incorporate the uncertainty in the stability of f_{DAQ} , which is generated by an Agilent E4400 synthesizer. According to its specifications, the clock signal is stable to $< \pm 1$ ppm from temperature-dependent effects when operated in the range 0–55 °C, and to $< \pm 1$ ppm/year from aging. In the latter respect, Francoise’s check in February 2005 (roughly four months after Run8) against the FAST Rubidium clock suggested that the Agilent signal was accurate to within 20 ppb, much better than the quoted specs. Nevertheless, if we add the two more conservative uncertainties in quadrature, we find that ± 1.4 ppm (i.e. ~ 140 Hz) should be a reasonable upper bound on the error due to clock stability. This value must then be applied to the error propagation formula for Equation 28.
2. It might be necessary to re-run Fred Gray’s Monte Carlo simulation of beating effects between the PSI cyclotron RF and our DAQ electronics clock using the unblinded frequency, to get a better estimate of the corresponding error.

3. Incorporate into λ_{Run8} any corrections and errors due to $pp\mu$ molecular formation. If the 1996 proposal is still to be trusted, the magnitude of $pp\mu$ effects should be less than 5 Hz.
4. When performing the subtraction $\Lambda_S = \lambda_{\text{Run8}} - \lambda_0$, we must take into account the following μp bound-state correction to $\lambda_0 = 455,160(8) \text{ s}^{-1}$:

$$\begin{aligned}\Delta\lambda_{\text{bound}} &= (Q - 1)\lambda_0 \\ &= -\frac{1}{2}(Z\alpha)^2\lambda_0 \\ &\sim -12 \pm ? \text{ s}^{-1},\end{aligned}$$

where Q is the so-called ‘‘Huff factor.’’ The complete expression for the capture rate is then

$$\begin{aligned}\Lambda_S &= \lambda_{\text{Run8}} - \lambda'_0 \\ &= \lambda_{\text{Run8}} - (\lambda_0 + \Delta\lambda_{\text{bound}}).\end{aligned}$$

If we ignore contributions from $pp\mu$ effects, we can estimate Λ_S from our preliminary result according to the formula

$$\begin{aligned}\Lambda_S &= \lambda_{\text{Run8}}^{\text{unblinded}} - \lambda'_0 \\ &= \left[\lambda_{\text{Run8}}^{\text{blinded}} \left(\frac{f_{\text{DAQ}}}{100 \text{ MHz}} \right) \right] - (\lambda_0 + \Delta\lambda_{\text{bound}}).\end{aligned}\tag{29}$$

5. Tally up all systematic errors.

Concerning the last item, in Table 14 I present preliminary error estimates for the μ^- disappearance rate in hydrogen.

We should soon have a μ^+ measurement from Fred to compare against λ_0 and serve as a consistency check on the MuCap apparatus. Also, as a collaboration we will have to decide how to reconcile Steve’s and my results into a single number. Hopefully our results will not be dramatically different.

8.3 Remaining serious concerns

At this point in time, I have three big concerns about my results:

1. My biggest worry is the mysterious discrepancy between the cathode-AND and cathode-OR treatments, as the present $\Delta\lambda \approx 8 \text{ s}^{-1}$ difference is significantly larger than the expected statistical deviation $\sigma = \sqrt{(12.86)^2 - (12.33)^2} = 3.7 \text{ s}^{-1}$. In contrast, Steve has previously reported a roughly 4 s^{-1} difference between his cathode-AND and cathode-OR results. I have therefore tried to track down any differences in our electron detector treatments, in the hopes of discovering what action is responsible for the different rates. Thus far I have found the following differences between our analyses:
 - (a) Steve has a much lower threshold for ePC cluster spark cuts (I think 10 wires in comparison to my 100 wires).
 - (b) Steve has a spark artificial deadtime of $\pm 25 \mu\text{s}$, compared to my $50 \mu\text{s}$.
 - (c) Steve dynamically masks hot ePC wires: if the number of wire hits within a certain time interval exceeds a pre-set threshold, Steve masks the offending wire for the remainder of the run.

Stage	Source	Correction (ppm)	Error (ppm)
Pre-unblinding	Statistics	—	27.5
	Muon stop definition	—	?
	Muon scatter	9.9	1.2
	High- Z impurities	41.0	3.5
	Deuterium	19.5	2.5
	μp diffusion + scattering + impact cut	6.1	1.1
	Muon detector inefficiencies	—	—
	Electron detector inefficiencies	—	?
Unblinding	Rescaling of error	—	< 1.1
Post-unblinding	DAQ clock stability (Agilent E4400)	—	1.4
	DAQ clock & beam structure beating	15	10
	Molecular formation ($\lambda_{pp\mu}$)	25	8
	Molecular transition (λ_{op})	10	7
	Total		~ 31 ?

Table 14: Preliminary tabulation of corrections and errors, statistical and systematic, for the total μ^- disappearance rate in hydrogen, $\lambda_{\text{Run8}}^{\text{unblinded}}$. Note that I have not included the errors on λ_0 and $\Delta\lambda_{\text{bound}}$, which will come into play when calculating Λ_S according to Equation 29. The numbers in the table are provided as ppm of $(\lambda_0 + \Lambda_S) \equiv (455,160 \text{ s}^{-1} + 688 \text{ s}^{-1}) = 455,848 \text{ s}^{-1}$. Thus, to convert from ppm to Hertz one must multiply by the factor $(455,848 \text{ s}^{-1} / (1 \times 10^6)) \approx 1/2$.

- (d) Steve incorporates eSC z -information into his electron detection algorithm.
- (e) Steve makes unique ePC1/ePC2 + eSC pairings, instead of using an “approval” technique like the Berkeley analysis.

Of these, the spark cut threshold setting seems to me to be the likeliest candidate for the different sizes of the separations between our respective cathode-AND and cathode-OR analysis results, since it seems most likely to introduce a time-dependent effect.

2. The high- Z impurity correction feels like a house of cards. It is the largest in magnitude of all of the corrections, it involves many assumptions and cherry-picking of information, and it covers different gas fills with varying TPC voltages and threshold settings. It would be reassuring if we could make the correction more robust, and get a better handle on its error bars.
3. Steve recently pointed out the existence of large fluctuations in the fitted decay rate for certain run groupings, which I also observe in my analysis (Figure 54). This raises the concern that there might be some (time-dependent?) phenomenon that we are not accounting for.

8.4 Possible improvements and further studies

Aside from the outstanding problems listed in the previous subsection, there are a few other things which could be done to improve my results or serve as consistency checks. However, I should preface the following

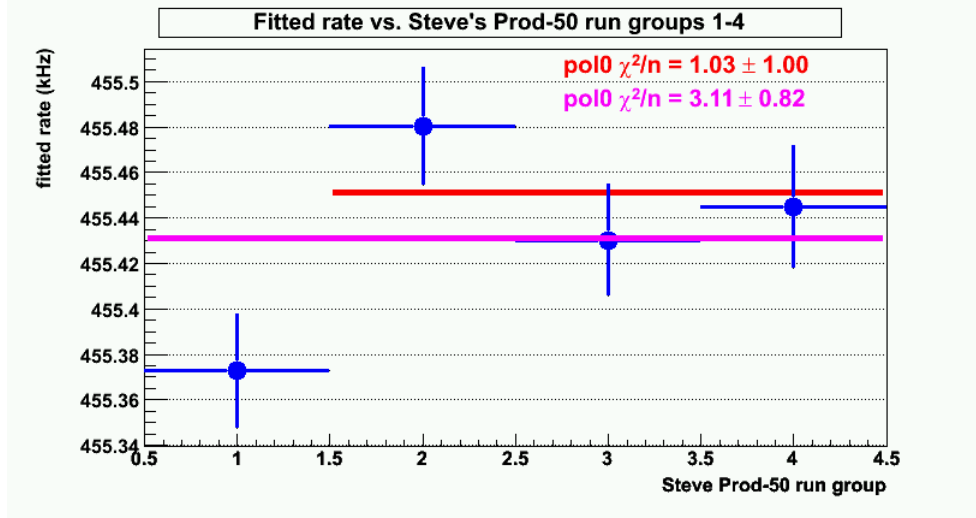


Figure 54: Illustration of the variation in the fitted rate vs. Steve's four Prod-50 run groupings (group1 = runs 11650–12374, group2 = runs 12376–12944, group3 = runs 12945–13364, group4 = runs 13365–14140). Groups 2–4 are statistically consistent, but the inclusion of group1 raises the χ^2 value dramatically, where the likelihood is below 5%.

list by saying that I do not expect any of its items to produce a significant departure from the existing result—in fact, some of them are likely negligible, and have been included only for the sake of completeness.

- Better establish the sensitivity of the final disappearance rate to various parameters involved in the corrections (especially the high- Z correction) and thus get a better handle on the final error bar.
- Finalize the high- Z correction numbers; we need to compile my and Peter Kammel's recent work on the subject into a definitive document.
- Look further at λ vs. time/run number.
- Perform the deuterium correction using the CalibD2 data set as well, and see if the resulting rate is consistent with the CalibNat correction.
- Include the CalibNat-46 and CalibNat-47 data sets, instead of just using “CalibNat” = CalibNat-48? When all of the CalibNat data files are included, the CalibNat and CalibD2 statistics are comparable.
- Look at λ vs. spatial variation? This is difficult because statistical considerations tend to limit the conclusions that can be drawn. Previous studies that divided up the TPC volume did not reveal any anomalous behavior, but I have seen unexpected structure in the rate vs. z_{rmeDet} plots.
- Move the standard fiducial cut's EL wall closer to the edges of the sensitive volume? I'm not sure what my original motivation for its placement was, and this step would increase the statistics.
- Double-check that my analysis was not afflicted with MQL “select” statement problems.
- Run the improved muon scatter identification algorithm which was recently implemented.

- Introduce timing offsets to align the absolute gondola times relative to the muSC, so as to avoid pushing the fit start time out to 110 ns?
- Look at the per-gondola rate uniformity after the deuterium correction.

References

- [1] S. Ritt et al., "MIDAS Analyzer."
http://midas.triumf.ca/doc/html/analyzer_struct.html
- [2] MuCap master CVS repository [mucap].
<http://kaon.physics.berkeley.edu/cgi-bin/viewcvs.cgi/#dirlist>
- [3] Tom Banks, "MuCap Berkeley Analysis Software, User Manual."
[mucap]/mu/doc
- [4] Steven Clayton, "MuCap Illinois Analysis Software, User Manual."
[mucap]/mu/src/uiuc/doc
- [5] Fred Gray, "MQL: The Muon Query Language."
[mucap]/mu/mql/doc
- [6] "The GNU m4 macro processor."
<http://www.gnu.org/software/m4/>
- [7] "ROOT: An object-oriented data analysis framework."
<http://root.cern.ch/>
- [8] D.R. Price, "CVS – Concurrent Versions System."
<http://www.nongnu.org/cvs/>
- [9] Tom Banks, "MuCap Run8 Muon Detector Inefficiencies," MuCap Note #39.
<http://ten.npl.uiuc.edu:8085/MuCap+Notes/39>
- [10] Steven Clayton, " μp Diffusion and Impact Parameter Cuts in MuCap."
http://www.npl.uiuc.edu/exp/mucapture/coll/mucap/smclayto/mup_bcut_calcs.pdf

A Shared Berkeley and Illinois analysis settings (common.h)

Below are the contents of the `common.h` header file which resides in the `mu/src/shared/` directory. Both the Berkeley and Illinois analyses use these settings either directly or indirectly.

```
// DAQ Constants
static const int kMaxEvents          = 31000;
static const int kMaxCrates          = 8;
static const int kMaxTDC400s        = 9;
static const int kMaxCaens           = 3;
static const int kMaxComps           = 3;
static const int kMaxCompFpgas       = 26;
static const int kMaxFadcs           = 4;
static const int kMaxCamacAdcs       = 10;
static const int kNumCaenChannels    = 128;
static const int kNumCompFpgaChannels = 64;
static const int kNumFadcChannels    = 4;

// CAEN trailing edge and rollover check parameters
static const int kCaenTrailingEdgeLimit = 100;
extern double rolloverPeriod;           // 400000 ns
extern double rolloverWidth;           // 5 ns
static const int kNumRollovers         = 3;
static const int kNumRolloverPairs     = 3;
extern double rolloverTimeOffset[kNumRollovers]; // 0, 15.2637, 11.924

// MuSC and MuSCA Parameters
static const int kMuSCADInterval       = 29; // ns
static const int kMuSCAD2Interval      = 120; // ns
static const int kMuSCAADInterval      = 90; // ns
static const int kMuSCMismatchThreshold = 3;
static const int kMuSCSignalstCoincHalfWidth = 7;
extern double muSCCopy2TimeOffset;
extern double muSCATimeOffset;
extern double muSCGateBTimeOffset;
extern double muSCGateBBarTimeOffset;
extern double muSCRoutedTimeOffset[4];

// MuPC Constants
static const int kMuPC1NumXWires = 24;
static const int kMuPC1NumYWires = 24;
```

```

// TPC Constants
static const int kTPCNumThresholds      = 3;
static const int kTPCNumAnodes          = 75;
static const int kTPCNumAnodeSections   = 5;
static const int kTPCNumStrips          = 35;
static const int kTPCNumStripSections   = 3;
static const int kTPCMaxNumSectionWires = 16;

// ePC Constants
static const int kePC1NumInnerStrips    = 192;
static const int kePC1NumAnodes         = 512;
static const int kePC1NumOuterStrips    = 192;
static const int kePC2NumInnerStrips    = 320;
static const int kePC2NumAnodes         = 1024;
static const int kePC2NumOuterStrips    = 320;

// eSC CAEN/COMP consistency cuts
static const int keSCCaenOnlyCutThreshold = 10;
static const int keSCCompOnlyCutThreshold = 25;

// Time intervals associated with skimming
static const double kSkimPP              = 25000;
static const double kSkimInterval        = 55000;
static const double kGeneralSkimInterval = 200000;

// Spark thresholds used in the "skimming" process
static const int kePC1AnodeSparkSizeLowCutoff    = 35;
static const int kePC1AnodeSparkSizeHighCutoff   = 490;
static const int kePC1IstripSparkSizeLowCutoff    = 35;
static const int kePC1IstripSparkSizeHighCutoff   = 180;
static const int kePC10stripSparkSizeLowCutoff    = 35;
static const int kePC10stripSparkSizeHighCutoff   = 180;
static const int kePC2AnodeSparkSizeLowCutoff     = 35;
static const int kePC2AnodeSparkSizeHighCutoff    = 990;
static const int kePC2IstripSparkSizeLowCutoff     = 35;
static const int kePC2IstripSparkSizeHighCutoff    = 300;
static const int kePC20stripSparkSizeLowCutoff     = 35;
static const int kePC20stripSparkSizeHighCutoff    = 300;
static const int keSCSparkSizeLowCutoff            = 10;
static const int keSCSparkSizeHighCutoff           = 55;
static const int kMuPC1XSparkSizeLowCutoff         = 10;
static const int kMuPC1XSparkSizeHighCutoff        = 24;
static const int kMuPC1YSparkSizeLowCutoff         = 10;
static const int kMuPC1YSparkSizeHighCutoff        = 24;

```

```
// Miscellaneous
static const double kPi    = M_PI;
static const double k2Pi   = (2.0 * M_PI);
static const bool  ON      = 1;
static const bool  OFF     = 0;
```

B Berkeley-specific settings (ucb_common.h)

Below are the contents of the `ucb_common.h` header file, which resides in the `mu/src/ucb/` directory. These settings are specific to the Berkeley side of the analysis.

```
// MuSC Constants
static const int kMuSCBookendWidth = 40500; // ns
static const int k25usPP           = 25000; // ns
static const int k35usPP           = 35000; // ns
static const int k40usPP           = 40000; // ns

// MuPC1 Constants
static const int kMuPC1XClusterMaxGap = 2;
static const int kMuPC1XClusterInterval = 260; // ns
static const int kMuPC1XSparkSizeCutoff = 21;
static const int kMuPC1YClusterMaxGap = 2;
static const int kMuPC1YClusterInterval = 260; // ns
static const int kMuPC1YSparkSizeCutoff = 21;

// TPC Constants
static const int kTPCFirstAmpAnode = 5;
static const int kTPCLastAmpAnode = 73;
static const int kTPCAnodeFlag = 1;
static const int kTPCStripFlag = 2;
static const double kTPCXDimension = 140.0; // mm
static const double kTPCYDimension = 120.0; // mm
static const double kTPCZDimension = 300.0; // mm
static const double kTPCXOffset = 0.0; // mm
static const double kTPCYOffset = -2.0; // mm
static const double kTPCZOffset = 8.5; // mm
static const double kTPCMWPCOffset = 375.0; // ns
static const double kTPCMaxDriftTime = 22420.0; // ns
static const double kTPCSVDriftInterval = (kTPCMaxDriftTime -
                                           kTPCMWPCOffset); // ns

// TPC Segment Search Definitions
static const int kSegMaxNumClusters = 5;
static const int kSegMinClustSep = 2;
static const int kSegMaxNumEndpts = 5;
static const int kSegUpFlag = 1;
static const int kSegDownFlag = -1;
```

```

// ePC1 Constants
static const double kePC1ActiveLength          = 580; // mm
static const int kePC1InnerStripClusterMaxGap  = 2;
static const int kePC1InnerStripClusterMaxSize = 11;
static const int kePC1InnerStripADInterval1    = 245; // ns
static const int kePC1InnerStripClusterTInterval1 = 240; // ns
static const int kePC1InnerStripADInterval2    = 1000; // ns
static const int kePC1InnerStripClusterTInterval2 = 300; // ns
static const int kePC1InnerStripSparkSizeCutoff = 90;
static const double kePC1InnerStripDiameter    = 376; // mm
static const double kePC1InnerStripCircum      =
    (M_PI * kePC1InnerStripDiameter); // mm
static const double kePC1InnerStripPhiOffset   = -0.048; // rad
static const double kePC1InnerStripScrew       = 46.194; // degrees
static const double kePC1InnerStripTanScrew    = 1.0426;
static const int kePC1AnodeClusterMaxGap       = 1;
static const int kePC1AnodeClusterMaxSize      = 10;
static const int kePC1AnodeADInterval1         = 215; // ns
static const int kePC1AnodeClusterTInterval1   = 210; // ns
static const int kePC1AnodeADInterval2         = 900; // ns
static const int kePC1AnodeClusterTInterval2   = 300; // ns
static const int kePC1AnodeSparkSizeCutoff     = 110;
static const double kePC1AnodeDiameter         = 384; // mm
static const double kePC1AnodeAbsPhiOffset      = -1.411; // rad
static const int kePC1OuterStripClusterMaxGap   = 2;
static const int kePC1OuterStripClusterMaxSize  = 11;
static const int kePC1OuterStripADInterval1     = 215; // ns
static const int kePC1OuterStripClusterTInterval1 = 210; // ns
static const int kePC1OuterStripADInterval2     = 1000; // ns
static const int kePC1OuterStripClusterTInterval2 = 300; // ns
static const int kePC1OuterStripSparkSizeCutoff = 90;
static const double kePC1OuterStripDiameter     = 392; // mm
static const double kePC1OuterStripCircum       =
    (M_PI * kePC1OuterStripDiameter); // mm
static const double kePC1OuterStripPhiOffset    = -0.171; // rad
static const double kePC1OuterStripScrew        = 43.811; // degrees
static const double kePC1OuterStripTanScrew     = 0.9593;
static const double kePC1HardcutLowZLimit       = -300; // mm
static const double kePC1HardcutHighZLimit      = 300; // mm

```

```

// ePC2 Constants
static const double kePC2ActiveLength          = 800; // mm
static const int kePC2InnerStripClusterMaxGap  = 1;
static const int kePC2InnerStripClusterMaxSize = 10;
static const int kePC2InnerStripADInterval1    = 245; // ns
static const int kePC2InnerStripClusterTInterval1 = 240; // ns
static const int kePC2InnerStripADInterval2    = 1000; // ns
static const int kePC2InnerStripClusterTInterval2 = 300; // ns
static const int kePC2InnerStripSparkSizeCutoff = 110;
static const double kePC2InnerStripDiameter    = 632; // mm
static const double kePC2InnerStripCircum      =
(M_PI * kePC2InnerStripDiameter); // mm
static const double kePC2InnerStripPhiOffset   = 0.026; // rad
static const double kePC2InnerStripScrew       = 45.738; // degrees
static const double kePC2InnerStripTanScrew    = 1.026;
static const int kePC2AnodeClusterMaxGap       = 1;
static const int kePC2AnodeClusterMaxSize      = 10;
static const int kePC2AnodeADInterval1        = 245; // ns
static const int kePC2AnodeClusterTInterval1  = 240; // ns
static const int kePC2AnodeADInterval2        = 1000; // ns
static const int kePC2AnodeClusterTInterval2  = 300; // ns
static const int kePC2AnodeSparkSizeCutoff     = 110;
static const double kePC2AnodeDiameter         = 640; // mm
static const double kePC2AnodeAbsPhiOffset     = -1.501; // rad
static const int kePC2OuterStripClusterMaxGap  = 1;
static const int kePC2OuterStripClusterMaxSize = 10;
static const int kePC2OuterStripADInterval1    = 245; // ns
static const int kePC2OuterStripClusterTInterval1 = 240; // ns
static const int kePC2OuterStripADInterval2    = 900; // ns
static const int kePC2OuterStripClusterTInterval2 = 300; // ns
static const int kePC2OuterStripSparkSizeCutoff = 110;
static const double kePC2OuterStripDiameter    = 648; // mm
static const double kePC2OuterStripCircum      =
(M_PI * kePC2OuterStripDiameter); // mm
static const double kePC2OuterStripPhiOffset   = -0.003; // rad
static const double kePC2OuterStripScrew       = 44.308; // degrees
static const double kePC2OuterStripTanScrew    = 0.976;
static const double kePC2HardcutLowZLimit      = -410; // mm
static const double kePC2HardcutHighZLimit     = 410; // mm

```

```

// eSC Constants
static const int keSCNumGondolas      = 16;
static const int keSCClusterInterval  = 65; // ns
static const int keSCSparkSizeCutoff  = 55;
static const double keSCDiameter      = 772; // mm
static const double keSCActiveLength  = 900; // mm
static const double keSCAbsPhiOffset  = (M_PI / 2.0); // rad
extern double keSCInnerTOffset[keSCNumGondolas+1];
extern double keSCInnerTSigma[keSCNumGondolas+1];
extern double keSCOuterTOffset[keSCNumGondolas+1];
extern double keSCOuterTSigma[keSCNumGondolas+1];
extern double keSCGondolaTOffset[keSCNumGondolas+1];
extern double keSCGondolaTSigma[keSCNumGondolas+1];
extern double keSCCaenCompTOffset;
extern double keSCCaenCompTSigma;

// Miscellaneous
static const int kSparkCutDeadtime    = 50000; // ns
static const double kEPS               = 0.001;
static const double kePC1eSCPhiSigma  = 0.20; // rad
static const double kePC2eSCPhiSigma  = 0.13; // rad
static const double PV_wall_radius    = 140; // mm

```

C List of Berkeley analysis modules

Stage 1: Skimming (From file mu/work.skim/MODULES)

MUnCompressRawData
MUnDuplicator
MCAenCompProcessRaw
MRolloverCheckMQL
MRolloverCheckC
MTDC400ProcessRaw
MMuSCAnalysisMQL
MMuSCAnalysisC
MeSCCaenCompCheckMQL
MeSCCaenCompCheckC
MSkimmerMQL
MSkimmerC

Stage 2: Ntuple production (From file mu/work.ana-skim/MODULES)

MUnCompressRawData
MFadcProcessRaw
MCamacAdcProcessRaw
MMuSCAnalysisMQL
MMuSCAnalysisC
MMuPC1AnalysisC
MMuPC1AnalysisMQL
MTPCSegmentSniffer
MTPCTrackAnalysisC
MTPCTrackAnalysisMQL
MePC1AnalysisC
MePC1AnalysisMQL
MePC1AnalysisMQL_copy2
MePC2AnalysisC
MePC2AnalysisMQL
MePC2AnalysisMQL_copy2
MeSCAnalysisC
MeSCAnalysisMQL
MGlobalSparkCut
MRates
MeDetCoincidenceCalcs
MCoincidenceCalcs
MTPCImpurityCaptureSearch
MFadcAnalysisC
MCamacAdcC
MCamacAdcMQL
MThresholdCalc

Stage 3: Ntuple analysis (From file mu/work.ana-ntuple/MODULES)

MNtupleAnalysisMQL
MNtupleAnalysisC

D The muon scatter rate shift error calculation

I have found that when scatter events are removed from the lifetime histograms, the fitted decay rate decreases from λ' to λ by

$$\Delta\lambda = \lambda - \lambda' \sim -3 \text{ Hz} . \quad (30)$$

Although this downward shift is to be expected, it's not obvious what the corresponding error $\sigma_{\Delta\lambda}$ should be, since the shift itself is within the statistical error $\sigma_\lambda \sim 12 \text{ Hz}$. The following calculations address this problem.

Since scatter events are relatively rare ($\sim 1 \times 10^{-4}$), we can assume that their effects on the decay rate scale linearly. If we let N be the number of normal TPC muon stops and N_s be the number of scatters, then we can describe things in the following manner:

$$\begin{aligned} N\lambda + N_s\lambda_s &= (N + N_s)\lambda' \\ N\lambda + N_s(\lambda + \Delta\lambda_s) &= (N + N_s)\lambda' \\ (N + N_s)\lambda + N_s\Delta\lambda_s &= (N + N_s)\lambda' \\ \lambda + f_s\Delta\lambda_s &= \lambda' \\ \Delta\lambda &= -f_s\Delta\lambda_s . \end{aligned} \quad (31)$$

The error on $\Delta\lambda$ is therefore

$$\sigma_{\Delta\lambda} = \sqrt{f_s^2\sigma_{\Delta\lambda_s}^2 + (\Delta\lambda_s)^2\sigma_{f_s}^2} . \quad (32)$$

Since $\Delta\lambda_s = (\lambda_s - \lambda)$, the error $\sigma_{\Delta\lambda_s}$ is dominated by the scatters, owing to their far smaller statistics and correspondingly larger error:

$$\sigma_{\Delta\lambda_s} \approx \sigma_{\lambda_s} . \quad (33)$$

As a demonstration, let us calculate the error for the Prod-50, cathode-OR muon scatters. The relevant numbers are

$$\begin{aligned} f_s &= 1.214 \times 10^{-4} \\ \sigma_{f_s} &= 0.002 \times 10^{-4} \\ \Delta\lambda_s &= 44,900 \text{ Hz} \\ \sigma_{\Delta\lambda_s} &= 1,678 \text{ Hz} \end{aligned}$$

Plugging them in yields

$$\begin{aligned} \sigma_{\Delta\lambda_s} &= \sqrt{(1.214 \times 10^{-4})^2(1678 \text{ Hz})^2 + (44,900 \text{ Hz})^2(0.002 \times 10^{-4})^2} \\ &\approx 0.20 \text{ Hz} , \end{aligned}$$

so the rate shift due to the removal of muon scatter events is

$$\Delta\lambda_{\text{scatter}} = -3.12 \pm 0.20 \text{ Hz} .$$

E Addenda

This section will be used to list any discoveries, issues, etc. that arose after the document was first completed.

E.1 Update on Run8 deuterium concentrations (September 20, 2006)

Claude has informed me of two recent (April 2006) external measurements of the CalibNat deuterium concentration, again performed by Dr. Mathias Saurer of PSI and Dr. Max Doebeli of ETH, who independently obtained:

Investigator	Fill sample	c_d (ppm)
Doebeli	CalibNat	126.9 ± 1.9
Saurer	CalibNat	$126.7 \pm ?$

Claude considers these measurements to be more reliable than Saurer's previous CalibNat result from January 26, 2005, which (after slight correction) gave $c_d = 117.6$ ppm. The existing Prod-50 c_d measurement results remain unchanged. The present situation is thus summarized in the following table:

Fill	c_d (ppm)	Method of determination
Prod-50	1.44 ± 0.13	External measurements
CalibD2	17.75 ± 0.25	Claude calculation
CalibNat	126.9 ± 1.9	External measurements

Unfortunately, the external c_d measurements and Steve's impact-parameter-based calculations are no longer in great agreement. Recall from Section 7.3 that Steve originally obtained a CalibNat/Prod-50 deuterium concentration ratio of $c = 81.3 \pm 7.3$ ppm, which was almost exactly the same as the value from earlier external measurements, $c = (117.6/1.44) \approx 81.7$ ppm. However, the latest Saurer result $c = 88.1 \pm 8.1$ ppm is now $\sim 1\sigma$ away from Steve's latest result, $c = 80.6 \pm 7.1$ ppm. Fortunately, the rate λ is not too strongly affected by these changes. For example, in the cathode-OR case, the two c ratios (80.6 vs. 88.1) give zero extrapolation results that are different by only 0.75 s^{-1} ($455,404.41 \text{ s}^{-1}$ vs. $455,405.16 \text{ s}^{-1}$).

E.2 Detector geometry

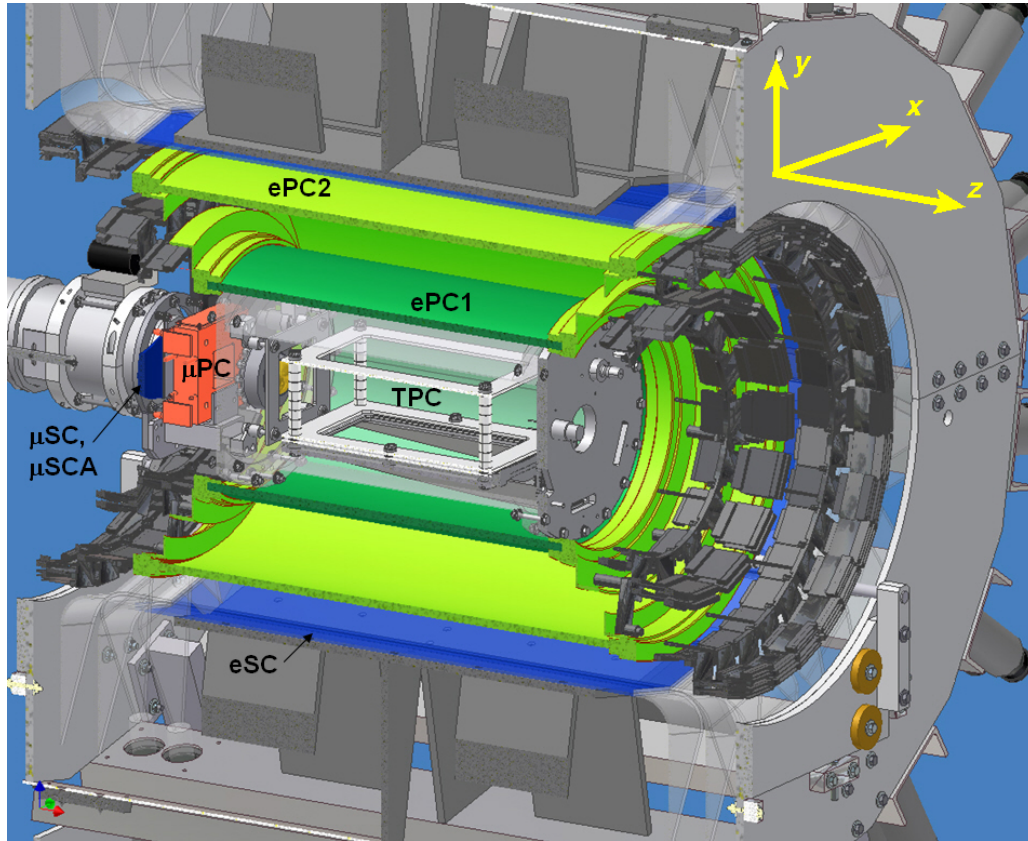


Figure 55: The MuCap apparatus in cross section, labeled with our conventional coordinate system.

E.3 Unsolved mysteries

Below is a list of unsolved mysteries from the Run8 analysis, sorted roughly in decreasing order of importance:

- Why do I observe such large differences between my eSC-only, cathode-OR, and cathode-AND fitted rates?
- Why are there so many inconsistencies in the high- Z results? For example, time fits such as the one shown in Figure 51 give concentration values that are wildly inconsistent with other information. Furthermore, many of the capture yields, gas chromatography measurements, and calculated estimates are in disagreement.
- Why does the 120 mm impact cut produce such a large effect on the decay rate? According to my estimates, that particular impact cut should raise the rate by roughly 4 Hz (2.8 Hz from μp diffusion + scattering, and 1.2 Hz from μd diffusion + scattering). In reality, the 120 mm cut increases the rate by 10.6 Hz! Is this to be expected? Does it reflect an improvement in the strength of the decay signal, or does it reflect the incorporation of some unaccounted-for effect?
- What is responsible for the small but distinct nonuniformity in the high- Z capture yield vs. y -position? The yield decreases linearly with increasing y .
- What creates the “tail” in the accidental background on the right-hand side (i.e. in the time interval 22.7–25 μ s) of the μ SC/TPC drifttime plot?

E.4 Comparison of Berkeley and Illinois Run8 μ^- results

Below I present a side-by-side comparison of the sequence of steps leading to the final μ^- disappearance rates quoted in my and Steve Clayton's analysis reports. Although our final results are fairly close, the paths we follow are quite different, so the apparently good agreement may be accidental.

Condition	Steve			Tom		
	$\Delta\lambda$ (s $^{-1}$)	λ (s $^{-1}$)	σ_λ (s $^{-1}$)	$\Delta\lambda$ (s $^{-1}$)	λ (s $^{-1}$)	σ_λ (s $^{-1}$)
eSC-only		455,425.1	12.6		455,433.9	12.9
cathode-OR	−5.3	455,419.8	12.2	−8.2	455,425.7	12.4
cathode-AND	−4.2	455,415.6	12.6	−8.6	455,417.1	13.0
cathode-OR		455,419.8	12.2		455,425.7	12.4
cathode-OR, 120 mm impact cut	+14.2	455,434.0	12.1	+10.6	455,436.3	12.3
... remove scatters	−1.1	455,432.9	12.1	−4.5	455,431.8	12.3
... high- Z correction	−13.9	455,419.0	12.1	−18.7	455,413.1	12.4
... deuterium correction	−12.1	455,406.9	12.2	−8.6	455,404.5	12.5
... μp diffusion correction	−2.7	455,404.2	12.2	−2.8	455,401.7	12.5

Table 15: Comparison of rates from my and Steve Clayton's analysis reports. All of the rates are from lifetime histograms which used good fiducial TPC muon stops, pileup protected by ± 25 μ s using both the μ SC and μ PC1XY-OR tables. Steve's fit range was 100–24000 ns, while mine was from 110–24900 ns, but this should make little difference.

Next is a side-by-side comparison of the fitted rates and EVH capture yields that Steve and I obtained for the Prod-50, CalibN2, and CalibNat fills. These are the numbers which are used to calculate the high- Z and deuterium impurity corrections listed in Table 15.

Fill	Steve			Tom		
	Y^{EVH} (ppm)	λ (s $^{-1}$)	σ_λ (s $^{-1}$)	Y^{EVH} (ppm)	λ (s $^{-1}$)	σ_λ (s $^{-1}$)
Prod-50	10.7 ± 0.1	455,434.0	12.1	12.9 ± 0.1	455,436.3	12.3
CalibN2	728.9 ± 2.8	456,370.4	53.0	774.9 ± 2.4	456,425.2	55.4
CalibNat	67.5 ± 0.6	456,471.7	35.5	48.6 ± 0.7	456,175.9	61.8

Table 16: Comparison of rates and EVH capture yields for different Run8 gas fillings, as found in my and Steve Clayton's analysis reports. The rates were obtained from fits to cathode-OR, 120-mm-impact-cut lifetime histograms.

Tables 15 and 16 raise several questions:

- Why do Steve and I obtain such different rates for the eSC-only, cathode-OR, and cathode-AND lifetimes?
- Why are there such large disparities in our fitted rates for the CalibN2 and CalibNat calibration fills?
- Why are my capture yields larger than Steve's for the Prod-50 and CalibN2 fills, but smaller for the CalibNat fill? Is it because Steve included CalibNat-46,47 in his CalibNat data?