SUITABILITY OF THE TEACHSPIN MUON LIFETIME DEVICE FOR TESTING QTD

HOWARD A. LANDMAN

CONTENTS

1. Introduction

D. Apsel[1, 2, 3] and I[4], by entirely different methods, both reached theories that predict the lifetime of an unstable particle can be altered by changing its energy. I call this effect Quantum Time Dilation (QTD). In particular, both of us predict that muon lifetime will be altered by an electrostatic potential to the extent of about 1% per 1.05 MV. This is clearly a testable assertion, but is sufficiently outside the mainstream of current physics research that it could be difficult to obtain funding from the usual sources to test it. In the hope of keeping the cost of testing as low as possible, I explore whether an inexpensive commercial student-grade muon lifetime apparatus can, or can be modified to, provide a sufficiently accurate measure of muon lifetime to conclusively falsify or confirm these theories.

2. The Teachspin device

The Teachspin muon lifetime device[5, 6] is self-contained except for needing an electrical outlet and an external (Windows or Linux) computer to store and display data and run the user interface software. It has three main components.

2.1. **The scintillator cylinder.** The primary detector comprises a solid transparent cylinder of scintillator-doped polyvinyltoluene, a single photo-multiplier tube (PMT), a high-voltage power supply for the PMT, and an amplifier that converts the PMT current output to a voltage signal. These are all contained in a light-tight aluminum shell which keeps out external photons that could cause experimental noise or even damage the PMT, and also keeps the high PMT voltage safely away from users. The shell also includes an LED circuit for creating artificial light pulses to test the PMT independently of particles, and some control knobs for the LED and the PMT. I did not use the LED feature.

Charged cosmic ray particles, mostly muons, strike the cylinder and can generate photons in the scintillator. Since cosmic rays have very high momenta (up to about 4 GeV/c) and the cylinder is relatively small with low stopping power (the manual says 120 MeV/c; our simulations using SRIM/TRIM software[7] gave about 62-70

MeV/c), most (about 97%) of the particles pass through the scintillator without stopping.

2.2. **The discriminator and counter.** Most of the active electronics is contained in a separate chassis. It includes a discriminator circuit with adjustable threshold, that can be tuned to reject pulses below the threshold. This produces an amplified "digital" pulse if the input "analog" pulse exceeds the threshold. The digital pulse has fixed height, but its width varies with the width of the analog pulse.

The digital pulse is then fed into an FPGA, which starts a lifetime count if one isn't already running. Since the FPGA runs on a 50 MHz (20 nS) clock, it is incapable of distinguishing times any shorter than that. The count runs for 1000 clock cycles (20 μS); if a second pulse is seen in that time, it is counted as a decay event, otherwise it is counted as a transit event.

Once a second, the number of events in the last second is written out, along with any decay events. Each decay event, and each event summary, is communicated via USB to the computer.

2.3. **Computer and software.** Software running on the host computer writes one line of text to the data file for each decay event, and one for each event summary. These are the only kinds of data in the files.

For summary records, the number of events seen is recorded, followed by a timestamp. Consecutive timestamps differ by one.

For decay events, the time between pulses in nS is written, and is always a multiple of 20 since it is computed by multiplying the raw count from the FPGA by the nominal clock period. It is also followed by a timestamp (number of seconds since some arbitrary starting date). Typically there are many seconds between decay events, although it is possible to have more than one in the same second.

For the most part, in my analysis I only use the decay event records. I did however perform some "sanity checks" on the summary records, with interesting results described below.

2.4. **Alternatives.** Similar experiments, using different equipment, are routinely performed at CalTech[8], UC Berkeley[9], U Mich[10], U Minnesota[11], and elsewhere. Most of these use scintillators far too large for our purposes. However, the lab notes describe equipment configurations for discrimination, timing and counting that could be useful.

There are other inexpensive detector options, including several do-it-yourself ones[12], but I don't analyze them in detail here. The FermiLab detector[13] seems the most promising of these, as it can do muon lifetime experiments and be powered by batteries.

3. Statistical preliminaries: exponential and geometric distributions

Before getting into the many sources of experimental noise and error, it will be worthwhile to analyze the situation where no such problems exist. For this purpose, we assume that we can measure a decay to infinite precision and that all events seen are real decays. We may also assume that the desired effect exists and can be precisely controlled, so that we may alter the muon lifetime at will. Then, we are left with the purely statistical problem of estimating the muon lifetime from a finite sample of decay events.

Even in the absence of an kind of experimental error, the inherently random nature of exponential decay introduces a source of variance which cannot be removed. Given the importance and simplicity of the exponential distribution, the question of how much data must be sampled has been studied extensively. Computing this will give us a hard lower bound for the actual experiment, as truncation, binning, noise and error can only increase this number.

Not everything in this section is actually used in the data analysis that follows later. Topics that aren't are included in the hope that they may prove useful in related experiments.

3.1. **Maximum Likelihood Estimate and its confidence intervals.** The maximum likelihood estimate $\hat{\tau}$ for the lifetime is just the average of all the measured decay times,

$$\hat{\tau} = \frac{1}{n} \sum_{k=1}^{n} t_k$$

with an exact $100(1 - \alpha)\%$ confidence interval given by[14]

$$\frac{2n\hat{\tau}}{\chi^2_{1-\alpha/2,2n}} < \tau < \frac{2n\hat{\tau}}{\chi^2_{\alpha/2,2n}}$$

Using an online χ^2 calculator[1] we get Table 1.

This table can be read in two ways. First, it tells us how many data points we need to know the true lifetime to within a certain error with a given level of confidence. For example, to know the lifetime to within 0.1% with 98% confidence, we need 8439886 events (first column, last row). This is a two-tailed test, as our estimate could be either too high or too low. For this purpose, we read the χ^2 but ignore the voltage.

We can also use it to determine the number of events needed (assuming the theory is true) to produce a result with any given degree of unlikeliness in the null hypothesis (that there is no effect). This varies with the voltage of the VDGG,

---

[1]Using the "Calculate X^2 from probability Q and d" function at `http://www.fourmilab.ch/rpkp/experiments/analysis/chiCalc.html` and iterating on d until it converges to the desired value for χ^2. This algorithm only appears to be numerically stable to about 7 decimal places, so any digits beyond that may not be meaningful.

TABLE 1. Events n needed to get confidence $(1 - \alpha)$ at given χ^2

	$\alpha = .02$	$\alpha = .01$	$\alpha = .005$	$\alpha = .002$	$\alpha = .001$
5.25 MV ($\frac{\chi^2_{\alpha/2,2n}}{2n} = 1.050$)	3459	4446	5457	6820	7866
2.10 MV ($\frac{\chi^2_{\alpha/2,2n}}{2n} = 1.020$)	21303	27353	33549	41903	48316
1.05 MV ($\frac{\chi^2_{\alpha/2,2n}}{2n} = 1.010$)	84786	108825	133447	166644	192125
525 kV ($\frac{\chi^2_{\alpha/2,2n}}{2n} = 1.005$)	338287	434124	532287	664630	766205
210 kV ($\frac{\chi^2_{\alpha/2,2n}}{2n} = 1.002$)	2111068	2708846	3321111	4146496	4779886
105 kV ($\frac{\chi^2_{\alpha/2,2n}}{2n} = 1.001$)	8439886	10829342	13276595	16575156	19105704

since higher voltages are predicted to produce stronger effects which are easier to distinguish from random noise. Thus, we can read it as the number of events required, at a given voltage, to produce data making the null hypothesis as unlikely as desired. (If we pick a single direction, we have a one-tailed test and so the statistical significance level is actually $\alpha/2$.)

Given that higher voltages should require many fewer events, it seems desirable to get as large a voltage as practicable, so as to reduce the data requirements. However, larger VDGGs are more expensive. Very roughly, the sphere diameter needs to be at least 1 mm per kV (= 1 m per MV), and the mass (and hence material cost) of the VDGG goes up as at least the cube of this.

3.2. Sequential Probability Ratio Test.
When one is only trying to distinguish two hypotheses H_0 and H_1, and is collecting data slowly over time, a log-likelihood approach leads directly to the Sequential Probability Ratio Test (SPRT)[2] of Signal Detection Theory. To the ith piece of evidence e_i (in this case, a measured decay time) we assign a weight

$$w_i = \log \frac{P(e_i|H_1)}{P(e_i|H_0)}$$

and our decision variable y, the log-likelihood after n samples, is given by the simple sum

$$y_n = \log \frac{P(e_1, e_2, ..., e_n|H_1)}{P(e_1, e_2, ..., e_n|H_0)} = \sum_{i=1}^{n} \log \frac{P(e_i|H_1)}{P(e_i|H_0)} = \sum_{i=1}^{n} w_i$$

The termination criterion is that we stop and say H_1 is true if $y_n \geq \log \frac{1-\alpha}{\alpha}$, and we stop and say H_0 is true if $y_n \leq \log \frac{\beta}{1-\beta}$, and otherwise we continue to collect data. Here α is the acceptable probability that we falsely conclude H_1 is true, and β the acceptable probability that we falsely conclude H_0 is true.

---

[2]see Gold & Shadlen[15] pages 538-542 for a nice brief exposition, particularly the sidebar on page 541.

For the special case of two exponential distribution hypotheses differing only in their lifetime parameter, ($\frac{1}{\tau_0}e^{-t/\tau_0}$ versus $\frac{1}{\tau_1}e^{-t/\tau_1}$), we have that

$$P(e_i|H_0) = \frac{1}{\tau_0}e^{-t_i/\tau_0}$$

$$P(e_i|H_1) = \frac{1}{\tau_1}e^{-t_i/\tau_1}$$

$$w_i = w(t_i) = \log\frac{P(e_i|H_1)}{P(e_i|H_0)} = \log(\frac{1}{\tau_1}e^{-t_i/\tau_1}) - \log(\frac{1}{\tau_0}e^{-t_i/\tau_0})$$

$$= \frac{-t_i}{\tau_1} - \log(\tau_1) - \frac{-t_i}{\tau_0} + \log(\tau_0)$$

$$= (\log(\tau_0) - \log(\tau_1)) + t_i(\frac{1}{\tau_0} - \frac{1}{\tau_1})$$

The expected weight if H_1 is true would then be

$$\langle w|H_1\rangle = \int_0^\infty P(t)w(t)dt$$

$$= \int_0^\infty \frac{1}{\tau_1}e^{-t_i/\tau_1}\left((\log(\tau_0) - \log(\tau_1)) + t_i(\frac{1}{\tau_0} - \frac{1}{\tau_1})\right)$$

$$= (\log(\tau_0) - \log(\tau_1))\int_0^\infty \frac{1}{\tau_1}e^{-t/\tau_1}dt + (\frac{1}{\tau_0} - \frac{1}{\tau_1})\int_0^\infty \frac{1}{\tau_1}te^{-t/\tau_1}dt$$

But the first integral is just 1 and the second integral just gives the expected value of t over the entire distribution, which we know is τ_1, so

$$\langle w|H_1\rangle = (\log(\tau_0) - \log(\tau_1)) + (\frac{\tau_1}{\tau_0} - 1)$$

If we are able to change the muon lifetime by a fraction δ, so that $\tau_1 = (1+\delta)\tau_0$, then

$$\langle w|H_1\rangle = (\log(\tau_0) - \log((1+\delta)\tau_0)) + (\frac{(1+\delta)\tau_0}{\tau_0} - 1) = \delta - \log(1+\delta)$$

For $\delta = 0.01$, which is what we would predict for a voltage of 1.05MV, we get

$$\langle w|H_{1.01\tau_0}\rangle = 0.01 - 0.009950331 = 0.000049669$$

This can be used to directly calculate the expected number of events needed to achieve any given likelihood ratio. For example, if we want H_1 to be 1000 times more likely than H_0, we need about

$$\frac{\log(1000)}{\langle w|H_1\rangle} = \frac{6.907755}{0.000049669} = 139075 \pm 373$$

decay events, where $373 \approx \sqrt{139075}$ is an estimate of the standard deviation of this number. This is roughly comparable to the estimate 166644 given in the $\chi^2 = 1.01, \alpha = .002$ entry of Table 1.

3.3. **Fisher Information analysis.** We can also analyze this in terms of Fisher Information. FI is mostly useful through the Cramer-Rao inequality

$$(\Delta \tau)^2 \geq \frac{1}{I(\tau)}$$

in which the inverse of the Fisher Information gives a lower bound on the variance of our estimate of the lifetime τ.

The FI for our purposes measures the amount of information that a single decay time t carries about the unknown lifetime parameter τ upon which the probability of t depends. It is defined as the expectation of the square of the partial derivative with respect to the parameter of the log-likelihood function.

$$I(\tau) \equiv E\left[\left(\frac{\partial}{\partial \tau} \log f(t; \tau)\right)^2 \bigg| \tau\right] = E\left[\left(\frac{\partial}{\partial \tau} \log\left(\frac{1}{\tau} e^{-t/\tau}\right)\right)^2 \bigg| \tau\right]$$

$$= E\left[\left(\frac{\partial}{\partial \tau}\left(\frac{-t}{\tau} - \log(\tau)\right)\right)^2 \bigg| \tau\right] = E\left[\left(\frac{t}{\tau^2} - \frac{1}{\tau}\right)^2 \bigg| \tau\right] = \frac{1}{\tau^4} E\left[(t - \tau)^2 \big| \tau\right]$$

But this last expectation is just the variance of t, which is τ^2, so the FI for a single decay event is

$$I(\tau) = \frac{1}{\tau^2}$$

and for n decay events is just

$$I(\tau) = \frac{n}{\tau^2}$$

so that

$$(\Delta \tau)^2 \geq \frac{1}{I(\tau)} = \frac{\tau^2}{n}$$

or

$$\frac{\Delta \tau}{\tau} \geq \frac{1}{\sqrt{n}}$$

If we assume approximately Gaussian error in τ for n large, we can use this to estimate the minimum number of events required to know *tau* with a standard deviation $\sigma \equiv \Delta \tau < \alpha \tau$ as

$$n \geq \frac{1}{\alpha^2}$$

From this we can see that $\alpha = .001$ gives $n \geq 1000000$. This is not exactly comparable to the numbers calculated above, since for H_1 and H_0 differing by 1% it would separate the two hypotheses by 10σ.

If we apply this to the latest MuLan muon lifetime number 2.197019 ± 0.000021 μS[19], we get $n \geq 10945334436$, implying that it should have taken a *minimum* of about 11 billion muon decay events to achieve this accuracy, even in the complete absence of any experimental error.

3.4. **Truncation of distribution.** It is traditional to truncate the exponential distribution on the right; the Teachspin software does this by default at no more than 20 μS (about 9 lifetimes). This is done both because we cannot wait forever for any single muon to decay, and because after some time any "decay" event seen is much more likely to be noise than an actual decay. However, it is rare that this truncation is treated exactly.

In this subsection we continue to ignore the noise issue, but focus entirely on the problem of estimating the true distribution mean from a truncated sampling procedure. This problem has been solved analytically by Al-Athari[17]; most of what follows is adapted from his paper.

Let the length of the sampling interval be b. If our complete distribution is

$$\frac{1}{\tau}e^{-t/\tau}$$

with mean τ, the truncated distribution has mean

$$\tau' = \tau - \frac{b}{e^{b/\tau} - 1}$$

where the second term essentially subtracts off the mean of the truncated portion, weighted by the probability that a sample from the full distribution would be truncated.

Especially for large sample size n, we would expect the sample mean $\bar{t}$ to be less than $b/2$, but there is some probability that this will not be true. If it isn't, then no maximum likelihood estimator exists. However, the chance of this occurring rapidly approaches 0 as $n \to \infty$.

The likelihood function is

$$\frac{e^{-n\bar{t}/\tau}}{\tau^n(1 - e^{-b/\tau})^n}$$

Maximizing this (or more usually its log) gives the MLE $\hat{\tau}$ for τ. (Al-Athari actually finds the MLE for $\theta = 1/\tau$, but since he uses the log-likelihood, the inversion becomes just a sign change and the MLE is the same either way.) If $\bar{t} < b/2$ we are guaranteed that the MLE exists, i.e. that the derivative of this is zero at some finite value. (If $\bar{t} \geq b/2$, the MLE doesn't exist, but we can use $b/2$ as our estimate. This is Al-Athari's Modified MLE (MMLE), and is equivalent to assuming a uniform distribution, which is the limit of the exponential distribution as the lifetime goes infinite.)

Thus (skipping a few steps) we want to solve

$$n(\hat{\tau}' - \bar{t}) = 0$$

i.e. find the $\hat{\tau}'$ that correctly predicts the sample mean. (In the full exponential case this is simply $\hat{\tau} = \bar{t}$.) Since we know n is not zero, we get $\hat{\tau}' = \bar{t}$.

Al-Athari stops here, but that doesn't give us a closed-form solution for $\hat{\tau}$ as a function of b and $\bar{t}$. Substituting back in, we get

$$\hat{\tau}' = \hat{\tau} - \frac{b}{e^{b/\hat{\tau}} - 1} = \bar{t}$$

which can be solved numerically for $\hat{\tau}$ given $\bar{t}$ and b.

3.5. **Binning and the geometric distribution.** The Teachspin timing FPGA operates on a 50 MHz (20 nS) clock ± 100 ppm. Ignoring clock error for the moment, what is the effect of putting decays into finite-width bins?

3.5.1. *Geometric distribution.* Binning converts the exponential distribution into a geometric distribution. If we number the bins starting at 0, the probability that a single muon will decay in bin k is

$$P(K = k) = (1 - p)^k p$$

where p is the probability that a muon will decay in 1 bin time if it has not decayed already. For a given lifetime τ and bin width b, we have

$$p = p(\tau, b) = \int_{t=0}^{b} \frac{1}{\tau} e^{-t/\tau} dt = -e^{-t/\tau} \Big|_0^b = 1 - e^{-b/\tau}$$

For $\tau = 2031.79$ nS (our experimental result below), and $b = 20$ nS, this gives $p = 0.0097574$, so for each bin there is about a 1% chance that a muon will decay, and a 99% chance it will survive. The expected value is just

$$\langle K \rangle = \frac{1}{p}$$

and the MLE for p just the inverse of the sample mean

$$\hat{p} = \frac{1}{\bar{K}}$$

Truncation of the geometric distribution has similar effects to truncation of the exponential distribution. The probability that a muon which decays in one of the first B bins will decay in bin k is

$$P(K = k) = \frac{(1 - p)^k p}{1 - (1 - p)^B}$$

which renormalizes the truncated distribution to have total probability 1.

3.5.2. *Bin offset.* The decays within a bin could of course have happened at any time during the bin window. For very small bins, we expect that the average decay time within a bin will be slightly less than the bin start time plus half the bin width, since the exponential distribution is monotonic decreasing. We can quantify this precisely using the truncated exponential analysis given above. Let $\tau'(\tau, b)$ be the expected value of the decay time in a bin of width b. Then

$$\tau'(\tau, b) = \tau - \frac{b}{e^{b/\tau} - 1}$$

as before, except that we now expect b/τ to be fairly small. For $\tau = 2031.79$ nS and $b = 20$ nS, this gives $\tau' = 9.9836$ nS, which is (as expected) slightly less than 10 nS. Thus, for an unbiased estimate with that τ and b, we should treat all the decays in the kth bin (which spans $20k$ to $20(k+1)$ nS) as if they decayed at $20k$ + 9.9836 nS.

Failing to do this and instead using 10 nS would give a positive bias to our estimated τ of about 0.0164 nS, or about +8 ppm. This is small relative to the clock uncertainty of ± 100 ppm.

3.6. **Exponential and geometric decay are memoryless.** We note here in closing that a property unique to the exponential and geometric distributions is that they are *memoryless*; the probability of an event occurring does not depend on how long we have already waited for it to occur. Another way of saying this is that they have a constant failure rate. Because muon decay (like nuclear decay) follows an exponential decay pattern, we do not need to be concerned with the history of the muon before it arrives in our apparatus. Indeed, the entire experiment would not be possible without this property, since we do not measure the *entire* lifetime of the muon from its creation in a pion decay while still traveling at nearly lightspeed, but only its lifetime after coming to rest in our detector.

4. Sources of Experimental Error

Here are all the sources of error I can think of in the Teachspin experiment, along with analyses of their effects and any ways of eliminating or compensating for them.

4.1. **Offset between stopping and decaying pulses.** The light in a stopping pulse is generated BEFORE the muon stops, while it is still moving. The light in a decay pulse is generated AFTER the muon decays, mostly by the emitted electron. Since the pulse detector works on the leading edge of both, that means it starts counting the stopping pulse too soon. If the muon is still moving, there may be some SR time dilation that would lengthen its lifetime.

NET EFFECT: Should make measured lifetimes longer than actual ones. Rough order of magnitude would be the width of the stopping pulse. For my setup, the analog pulses were about 25-35 nS wide at the base, and about 4-20 nS wide at

trigger level, based on visual inspection of about a hundred pulses on a digital storage scope. Thus we estimate a systematic bias of about $(4 + 20)/2 = 12$ nS and a σ of no more than 8 nS for this ($\sigma^2 = 64$ nS^2).

SOLUTION: Discarding initial bins (see below) makes this problem go away for the stopping pulse. For the decay pulse, we get a triggering offset of about 9 nS (from start of pulse to time of trigger) ± 8 nS. We also have to add another 5.5 ± 1.5 nS offset from the analog pulse to the digital pulse. This gives us a total triggering correction and error of $\delta_{trig} = -14.5 \pm 9.5$ nS.

4.2. Error due to variability in pulse size/shape.

Since the discriminator circuit is tuned to a specific voltage level, and the leading edge has finite slope, and the size of the pulse may vary, we would expect a larger pulse to trigger slightly sooner than a smaller pulse.

NET EFFECT: Random errors in the time that the pulse is detected. Min to max range would be somewhat less than half the typical pulse width (soonest would be the point where the largest pulse triggers, latest would be near the middle (top) of a small pulse just barely able to trigger). Based on the observations above, we estimate this to contribute $\sigma_p = 4$ nS ($\sigma_p^2 = 16$ nS^2) for each pulse.

SOLUTION: Discarding initial bins removes this problem for the stopping pulse, but not for the decay pulse.

4.3. Metastability of discriminator.

Any time a circuit has to take an analog input and make a digital decision on it, there will be signals very close to the decision threshold that can cause the circuit to hang in a metastable intermediate state. (This problem occurs most famously with flip-flops receiving data asynchronously to their clocks, but it can be caused by thresholds in voltage as well as in time.) There is no theoretical upper bound to how long this hanging can occur; however, the probability of continued hanging decays exponentially with time (it is an unstable equilibrium like a pencil balancing on its point). The exact behavior depends on circuit construction; a good circuit can make this problem very unlikely, at the cost of taking more time to make the decision. I don't know yet what the details of the Teachspin discriminator are.

NET EFFECT: Until analyzed more closely, we should assume that there may be occasional pulses which are effectively delayed by an entire clock cycle, or even (much more rarely) two. Since this can happen on either pulse, it could lengthen or shrink the measured decay time. Large and small pulses may behave differently; if there is a difference in pulse sizes between stopping and decay pulses, we could get a net average offset, but if not the mean of this should be zero. Although the magnitude of such an error would be on the order of ± 10 nS, metastability events are very rare (1 in millions or billions), so the estimated σ_{ms} here would be 0.1 nS or less.

4.4. **Digitization error of pulse time.** I assume that the pulses are digitized to synchronize with a clock pulse that has 20 nS (the bin size) cycle time since the file data is reported in multiples of 20 nS.

NET EFFECT: This adds a random error which is roughly uniformly distributed over the 20 nS cycle time, giving a systematic offset of approximately half the cycle time or 10 nS, and adding a variance of approximately $\sigma_{dig}^2 = \frac{1}{12}(20 \ nS)^2$ [18] to a single event time. If both stopping and decay pulses contribute this, the offsets cancel but the variances add. For n events, this variance scales linearly, so the standard deviation of the average scales as $1/\sqrt{n}$.

SOLUTION: Discarding initial bins removes this problem for the stopping pulse, but not for the decay pulse. For either pulse, to zeroth order we can treat all decays in a bin as occurring at the center of the bin (+10 nS). As noted earlier, a more exact correction can be computed if we know the decay time. Since we use this number to compute the estimated decay time, we would appear to be using circular reasoning; however, since the effect of the decay time on the offset is small, and the effect of the offset on the decay time is small, in practice we can "go around the circle" a few times and get excellent convergence.

IMPLEMENTATION: Using the bin offset formula above plus the uniform approximation for variance, we take $\delta_{dig} = (\tau - \frac{b}{e^{b/\tau}-1}) \pm \frac{10}{\sqrt{3n}}$ nS.

4.5. **Inability to detect very fast decays.** Very fast decays may be difficult or impossible to detect. Since the scintillator takes nanoseconds to emit all its light, two pulses starting less than a few nanoseconds apart may appear to be one wide pulse. There may also be issues throughout the analog circuitry, from the PMT to the discriminator; each may require some time after emitting a pulse to recover and be ready for another one. The recording computer, USB interface, and software may also impose limits. Examination of a large output file with over 100K decay events shows zero decays in the 0 nS and 20 nS bins, and a reduced number of decays in the 40 nS bin. This implies that all decays shorter than about 40-50 nS are being lost.

NET EFFECT: Decays faster than some minimum time are lost. Fitting to this data will give wrong answers.

SOLUTION: Fortunately, exponential decay is memoryless. We can simply discard the first few bins to solve this problem. Doing so also fixes many of the above problems. It also has the effect of synchronizing all the muon decay start times with the beginning of the first undiscarded bin.

IMPLEMENTATION: My analysis software currently discards the first 3 bins (all decays marked as 40 nS or less). This loses about 0.1% of the data, which is acceptable.

4.6. **Clock jitter and drift.** The clock driving the bin sampling is not perfect, but will have some jitter. It may also drift slightly over time due to temperature, circuit aging, power supply variation, or other causes. For any reasonably good

crystal oscillator, this will probably be much smaller than some other sources of error listed here and so negligible, but it is not zero. Teachspin claims its clock is accurate to ± 100 ppm. This could be measured by a device with a more accurate clock, but for this report we simply assume that the clock period may both have a fixed offset of ± 2 pS (100 ppm of 20 nS) and a uniformly-distributed random jitter of the same magnitude, contributing $\frac{1}{12}(4 \text{ pS})^2$ variance to individual decay times. This jitter is so much smaller than the digitization error above that we ignore it.

NET EFFECT: We are left with a $\delta_{clock} = \pm 10^{-4}\tau$ from our uncertainty about the actual clock period.

4.7. False decay noise due to separate transit events close together. This is probably the largest and most serious noise problem in the experiment.

NET EFFECT: If we assume that the transits are random, then the first transit has a uniform probability over time and the second one has the same distribution, so the probability of a false decay event from two transits is proportional to the decay window size times the square of the transit probability.

SOLUTIONS: Given the power of 2, the easiest way to reduce this noise would be to reduce transit events. Cutting them down by a factor of 10 would reduce the false decay from this cause by a factor of 100. One approach of course is to use a muon beam with momentum tuned to mostly stop in the detector, i.e., to increase the number of true decay events relative to the false ones. Staying with cosmogenic muons only, detector paddles above and below the main scintillator could give signals that would allow us to detect and filter out many transit events.

It is also possible to estimate the mean number of false decays per bin and to "subtract that out" of the data in various ways. However, this still leaves residual errors from (a) the error in estimating the mean, and (b) the variation about the mean. An alternate statistical approach would be to leave in all the decay events (true and false), but estimate the probability that each one is true (which will depend on the bin) and weight them accordingly. This is particularly simple in the Bayesian approach, where the log (relative) probability of the (true) event is simply added to the log (relative) likelihood of the hypothesis: we can just multiply that log event probability by the probability that the event is true and add that instead.

IMPLEMENTATION: We fit a model which has noise level as a parameter. Both errors in fitting the average noise level, and variance from the noise, should contribute error to our lifetime estimate, but quantifying these errors is not trivial. It can be estimated from Monte Carlo simulation, but we haven't done that.

4.8. Correlated noise. In the previous subsection we assumed that the background noise is independent, i.e. that the probability of a transit event is not correlated with other transit events. There could be cases where this is not true. For example, a highly energetic cosmic ray hitting part of the building structure

could generate a shower of secondary particles and thus cause a cluster of transit events.

Since these particles would mostly still be traveling at close to lightspeed, and light travels about a foot per nanosecond, the time skew between two particles (in nS) would be roughly the length difference of their paths to the detector (in feet). We expect that most of these clusters of events would spread only a few nS, and thus are being filtered out both by the inability of the device to measure short delays and by our discarding of the first three 20 nS bins. Thus, only skews of greater than 60 nS (implying either a delta-path-length of over 60 feet, or one particle moving much slower than lightspeed) would be able to cause false decay events. These seem sufficiently unlikely that we can ignore them.

4.9. **Transit Time Spread in PMT.** "The delay between the absorption of a photon at the photocathode and the output pulse from the anode of a PMT varies from photon to photon. The effect is called 'transit time spread' or TTS. There are three major TTS components in conventional PMTs and MCP PMTs - the emission at the photocathode, the multiplication process in the dynode system or microchannel plate, and the timing jitter of the subsequent electronics."[20]

NET EFFECT: Even in fairly slow PMTs. the TTS is rarely greater than 100 pS. This is small relative to some of the other sources of error we have considered, but we include a $\sigma_{tts} = \pm 0.1$ nS for completeness.

4.10. **Afterpulsing.** "Most detectors have an increased probability to produce a dark count pulse in a time interval of some 100 nS to some μS after the detection of a photon."[20] If we were doing single-photon detection, this would be interpreted as a (false) decay event if it occurred within the decay window. In this sense, afterpulsing is similar to background noise from transit events. However, the background noise is uncorrelated, while afterpulsing is highly correlated. Also, since we are looking for a large burst of photons rather than single photons, a single-photon-sized event would not normally trigger the pulse detector (our detector threshold (about 14 mV) was set well above the dark noise level of the PMT (about 1 mV)).

NET EFFECT: We expect afterpulsing to at worst slightly raise the dark noise level, and thus not be a serious source of false decay events by itself. However, it might slightly alter the pulse height threshold for decays at the afterpulse time. This could have the effect of slightly increasing the number of decay events seen around the afterpulse time, and hence slightly biasing the decay lifetime collection.

Since we have so far no details about the afterpulse behavior of the specific PMT used, it is difficult to estimate the magnitude of any such error.

4.11. **Binomial sampling noise of μ^+/μ^- mixture.** All of the above would apply even if all the muons were a single type. However, cosmogenic muons are are a mixture of muons (μ^-) and antimuons (μ^+) with a slight excess of antimuons.

Values for the ratio of μ^+/μ^- (the "charge ratio" r) in the literature vary from 1.08 to 1.3.

The CPT Theorem plus the assumption of EM-gauge invariance (as in the Standard Model) would require that these have identical masses and lifetimes in vacuum, but in matter their lifetimes differ because μ^- have an additional decay mode of being captured by an atomic nucleus. The best current lifetime numbers are $\tau_+ = 2197.019 \pm 0.021$ nS[19] for μ^+ in vacuum and $\tau_- = 2043 \pm 3$ nS for μ^- in carbon.

The lifetime measured in the Teachspin device is a weighted average of these two numbers, with the weights given by the actual numbers of μ^+ and μ^- detected. However, we don't know these actual numbers, and they will vary from data set to data set. So the expected mix of muons and antimuons will have a binomial distribution

$$f(n_+; n, p) = \binom{n}{n_+} p^{n_+} (1-p)^{n_-}$$

where n is the total number of muons, $p = r/(r+1)$ is the *a priori* probability that a detected muon will be a μ^+, n_+ is the actual number of μ^+ in our sample, $n_- = n - n_+$ is the number of μ^-. This has mean

$$\langle n_+ \rangle = np$$

and variance

$$\sigma^2(n_+) = np(1-p)$$

The variance is highest when $p = 1/2$, or $r = 1$. Thus we will have a higher variance if $r = 1.08$ than if $r = 1.3$. For $r = 1.08$ we get $p = 0.519$, $\langle n_+ \rangle = 0.519n$, and $\sigma^2(n_+) = 0.2496n$. For $r = 1.3$ we get $p = 0.565$, $\langle n_+ \rangle = 0.565n$, and $\sigma^2(n_+) = 0.2457n$.

NET EFFECT: Fortunately, the uncertainty in r doesn't affect our τ calculations as long as we assume that r is constant over time. We don't know the exact ratio of μ^+ to μ^-, since estimates in the literature vary and may depend on altitude, latitude, and other factors. This adds about $\pm \frac{1}{2}(0.565 - 0.519)(\tau_+ - \tau_-) = \pm(0.046)(154 \text{ nS}) = \pm 7.08$ nS uncertainty to what our effective (mixed) τ should actually be in theory, but it doesn't affect how we measure the effective τ.

Still, we have the binomial sampling variance of up to $0.2496n(\tau_+ - \tau_-)^2 = 5919.51n$ nS^2. (Note that this is very close to the theoretical maximum. Even at $p = 1/2$, we would see exactly 0.25 instead of 0.2496.) This is the variance contribution for the sum of all decay times; normalizing this down to a single average decay time gives us

$$\sigma^2_{binom} = \frac{5919.51}{n} \text{nS}^2$$

$$\sigma_{binom} = \frac{76.94}{\sqrt{n}} \text{nS}$$

For our larger data sets, with n from approximately 2330 to 20161, this gives $\sigma_{binom} = 1.59$ nS to $\sigma_{binom} = 0.54$ nS.

5. Analysis of actual Teachspin data

In this section we analyze actual Teachspin data to evaluate the real-world performance of the methods developed earlier.

5.1. **The data sets.** 25 pre-existing data sets were available from students in the spring 2012 Advanced Physics Lab course at CSU. These are shown in Table 2 as A through Y. Some of these were very short and recorded zero decay events. Equipment settings are not recorded in the data files and were impossible to reconstruct. I also collected 3 large data sets of my own between Sept 7th and Oct 15th, shown as Z_1 through Z_3. For these, the threshold dial was set to 3.5, resulting in a trigger voltage of about 14 mV at the pulse detector.

5.2. **Missing timestamps.** Some internal inconsistencies were noted in the files, particularly missing timestamps. One line is supposed to be written every second recording the number of transit events seen in that second; the last field in the line is a timestamp in seconds. The timestamps should be consecutive, and the number of them (shown in the table as "Seconds") should equal the last time stamp minus the first timestamp plus one (shown as "Length"). The difference "Missing" (= Length - Seconds) gives the number of missing timestamps.

Most of the time, only a small fraction of the data (less than 2%) is missing. This kind of effect might happen if the counter clock were slower than the PC clock, and the timestamps were generated by the PC. In that case we would expect the fraction of missing seconds to be consistent across all data sets, because it would be roughly equal to $1 - \frac{PCClockPeriod}{CounterClockPeriod}$. However, this is not what we see in the given data sets. The fraction missing, and their locations in the files, seem more random than that.

This could perhaps be explained by occasional software glitches dropping a data packet, possibly caused by the PC software sometimes not getting enough CPU time and a failure to implement a reliable communications protocol with adequate hardware buffering and handshaking. However, in data set C more than 90% of the data is missing, which cannot easily be explained in this way. Most of the missing data in C comes from a single gap in timestamps between 1328055282 and 1328210272, accounting for 154991 of the 156553. However, there are still many individual single missing timestamps. Thus it seems likely that there are at least 2 or 3 different causes of data dropout problems. Debugging these problems would be time-consuming and is beyond the scope of the present report.

This discrepancy complicates the data analysis, especially as regards the average number of transit events per second, which is one important way of estimating the background noise. It is probably fairest to divide Transits by Seconds to estimate this, since any transit events in the lost seconds would not have been counted.

TABLE 2. Muon lifetime data sets

	Date & Time	Seconds	Length	Missing	Decays	Transits
A	2012/01/31 14:32	61	61	0	0	428
B	2012/01/31 14:34	12	39	27	0	59
C	2012/01/31 14:46	10062	166615	156553	85	45678
D	2012/02/02 13:05	431255	431916	661	9009	3174760
E	2012/02/14 14:01	3238	3271	33	3196	24259
F	2012/02/16 13:08	3537	3543	6	89	26983
G	2012/02/16 14:10	11	11	0	0	46
H	2012/02/23 13:44	1032730	1034759	2029	21757	7718681
I	2012/03/06 13:36	331	332	1	7	2074
J	2012/03/06 13:42	592	593	1	5	1805
K	2012/03/06 13:52	348	348	0	1	2391
L	2012/03/06 13:58	1384	1386	2	24	9448
M	2012/03/06 14:21	646320	647669	1349	13902	3655672
N	2012/03/22 14:08	127	128	1	4	913
O	2012/03/22 14:11	7	7	0	0	42
P	2012/03/22 14:15	4358	4367	9	20034	213503
Q	2012/04/05 15:35	473249	474128	879	0	810
R	2012/04/19 13:22	9191	9570	379	89	44903
S	2012/04/24 14:41	167083	167379	296	2586	1155656
T	2012/04/26 13:21	4765	4772	7	81	22537
U	2012/04/26 15:00	5419	5427	8	86	28342
V	2012/05/01 15:32	1469	1471	2	21	7473
W	2012/05/03 13:14	5	5	0	0	18
X	2012/05/03 13:17	9583	9598	15	178	68048
Y	2012/05/03 15:57	430651	431341	690	7128	2981465
Z_1	2012/09/07 12:05	485774	486625	851	6726	2939733
Z_2	2012/09/14 11:32	736361	737832	1471	10454	4463818
Z_3	2012/09/28 12:04	1466550	1468753	2203	21502	8939629

5.3. **Gradient descent: Design.** In a Machine Learning approach, we could simply treat the data as being modeled by 3 parameters: the decay lifetime τ, the initial number of decaying muons N, and the background noise level ν. A minimum-variance fit to the data would give us estimates of all 3 parameters.

A more detailed procedure is as follows: We estimate the initial decay lifetime τ, the initial population N, and the initial background noise ν. The hypothesis for each bin k is then

$$H(N, \tau, \nu, k) = N(1-p)^k p + \nu$$

where $p = p(\tau) = 1 - e^{-b/\tau}$ as above, and ν is the expected number of false decay events in a single bin. We want to find the N, τ, ν that minimizes the error or

variance

$$V = \sum_{k=0}^{B-1} (H(N,\tau,\nu,k) - Y(k))^2$$

where $Y(k)$ is the actual data for bin k and B is the number of bins.

Since this is nonlinear in p and τ, direct solution of the equations using linear algebra is not possible. A gradient descent approach might work well. To apply it though, we need the following partial derivatives (arguments to Y and H suppressed for brevity):

$$\frac{\partial V}{\partial N} = \sum_{k=0}^{B-1} 2\,(H-Y)\,\frac{\partial H}{\partial N} = 2p \sum_{k=0}^{B-1} (H-Y)(1-p)^k$$

using the chain rule.

$$\frac{\partial V}{\partial \tau} = \sum_{k=0}^{B-1} 2\,(H-Y)\,\frac{\partial H}{\partial p}\frac{\partial p}{\partial \tau}$$

$$= 2N \sum_{k=0}^{B-1} (H-Y)((1-p)^k - k(1-p)^{k-1}p)e^{-b/\tau}\frac{b}{\tau^2}$$

$$= 2N \frac{b}{\tau^2} \sum_{k=0}^{B-1} (H-Y)((1-p)^{k+1} - k(1-p)^k p)$$

since $e^{-b/\tau} = (1-p)$, and

$$\frac{\partial V}{\partial \nu} = \sum_{k=0}^{B-1} 2\,(H-Y)\,\frac{\partial H}{\partial \nu} = 2 \sum_{k=0}^{B-1} (H-Y)$$

$$= 2B(\frac{1}{B} \sum_{k=0}^{B-1} H - \frac{1}{B} \sum_{k=0}^{B-1} Y) = 2B(\bar{H} - \bar{Y})$$

This last equation is particularly simple: it tells us that the minimum variance value of ν, given some fixed N and τ, is the value that makes the average H equal the average Y. So it might also be possible to use a hybrid approach, where we use gradient descent at each step for N and τ, and then solve exactly for the best ν.

5.4. **Gradient descent: Implementation and results.** We implemented the above approach in GNU Octave, a Matlab clone[21, 22]. Convergence on the N and τ terms was much slower than on the ν term. By itself, ν converged in 5 to 10 iterations as long as the descent scaling factor α was 0.005 to 0.009. At $\alpha = 0.01$, it showed signs of overshooting and oscillation; much higher than that and it failed to converge at all. On the other hand, at $\alpha = 0.009$, N and τ took a few million iterations to converge; a higher α could have sped this up.

We ran this process for 10 million iterations with $\alpha = 0.009$ on each of the data sets in Table 2 that had more than 1000 decays, resulting in Table 3. The last row (labeled Z_{all}) is the union of my 3 data sets. Each run took about 82 minutes on a

Decay time data and best exponential+noise fit
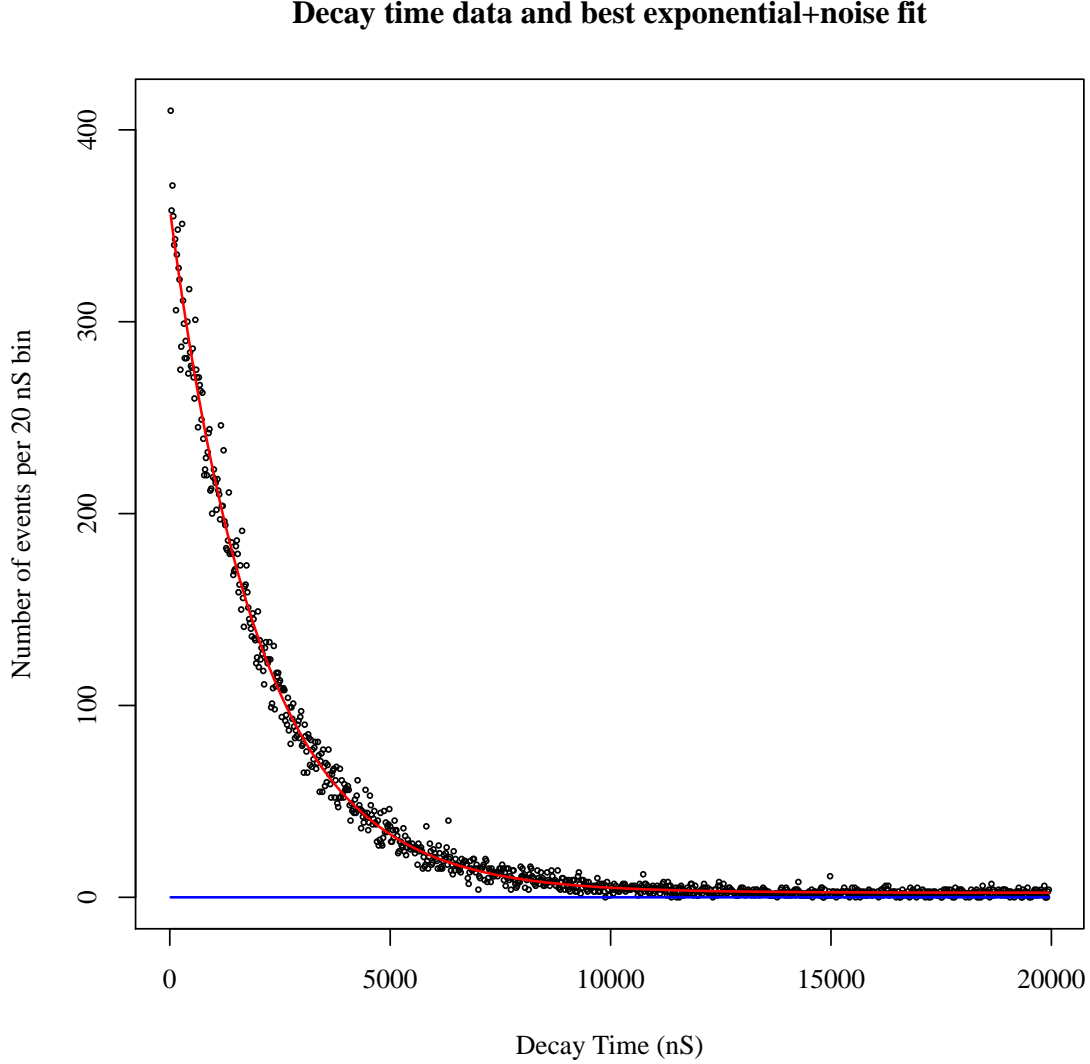
MacBook Pro 5.1 (2.53 GHz Intel Core 2 Duo) with 4 GB memory. Only 1 of the 2 processors was utilized, since Octave does not multiprocess. The first 3 bins of each data set were discarded (for reasons discussed above), leaving 997 bins each still 20 nS wide, with all the remaining decay times effectively reduced by 60 nS to match. This preprocessing was done by a Perl script, which wrote out a text file that could be read into Octave as an array. The array contained only the number

of decay events for each bin, and not any individual events; this collective data is what the gradient descent attempted to fit.

TABLE 3. Muon lifetime gradient descent results

	Total Decays	Raw N	Raw $B\nu$	Raw τ (nS)	Normalized N/D	Normalized $N/B\nu$
D	9009	8340.89	600.49	1991.95	0.9258	13.89
E	3196	3905.13	-758.08	1753.17	1.2219	-5.15
H	21757	20161.15	1441.73	1985.46	0.9267	13.98
M	13902	12984.06	817.30	1903.08	0.9340	15.89
P	20034	25717.74	-5686.85	1941.21	1.2837	-4.52
S	2586	2330.06	232.04	1981.91	0.9010	10.04
Y	7128	6502.15	561.19	2021.04	0.9122	11.59
Z_1	6726	6332.51	347.85	2040.65	0.9415	18.20
Z_2	10454	9664.95	731.46	1977.26	0.9245	13.21
Z_3	21502	20049.80	1315.43	2056.30	0.9325	15.24
Z_{all}	38682	36044.13	2397.83	2031.79	0.9318	15.03

One problem with this approach to solving for τ is that it may give us the best fit for τ, but gives us no information about the variance of this estimate.

5.5. **Issues with data sets E and P.** The results for data sets E and P in Table 3 seem suspect, since they have negative ν but a real noise level cannot be less than zero, and because we expect N to be somewhat less than Decays. We investigated this and found that for both data sets E and P, the decay times were not exponentially distributed but rather clustered in a few bins. For example, in data set P almost all the counts were in bins 42-47 (spanning 840-960 nS), as shown in Table 4. The most likely possibility is that the LED pulser was turned on,

TABLE 4. Event spike in data set P

Bin	nS	Events
39	780-800	0
40	800-820	2
41	820-840	5
42	840-860	161
43	860-880	3839
44	880-900	10698
45	900-920	4683
46	920-940	259
47	940-960	9
48	960-980	1

giving fairly regular pulse pairs with little variance in duration. I also considered that the discriminator threshold might have been set too low, resulting in PMT noise being recorded as decay events; but this would tend to cause a fairly broad distribution of noise events, and the observed spike seems far too narrow for that. In either case, the recorded events would not correspond to actual muon decays. Further evidence supporting this is that data set E had 3196 decay events in 3238 seconds (0.987 decays/S), and data set P had 20034 decay events in 4358 seconds (4.6 decays/S), while most data sets had a much lower rate (e.g. data set Z_3 had 21502 decay events in 1466550 seconds, or 0.0147 decays/S). At any rate, we exclude E and P from further analysis.

5.6. **Results with uncertainties.** To the raw τ from Table 3, we need to apply all our corrections and errors:

$$\sigma_{trig} = \pm 1.45 \pm \frac{9.5}{\sqrt{n}} \text{ nS}$$

where the 1.45 comes since our estimate of $\delta_{trig} = -14.5$ for the triggering offset was based on visual inspection of about 100 pulses so was probably only accurate to about 10%. The triggering offset itself cancels out because it affects not only the decay pulse, but also whether the stopping pulse makes it into the first non-discarded bin.

$$\sigma_p = \pm 4 \text{ nS}$$
$$\sigma_{ms} = \pm 0.1 \text{ nS}$$
$$\delta_{clock} = \pm 10^{-4} \tau$$
$$\sigma_{tts} = \pm 0.1 \text{ nS}$$

We don't need to apply the bin offset of about 10 nS since this is already accounted for in the fitting process, so instead of $\delta_{dig} = (\tau - \frac{b}{e^{b/\tau}-1}) \pm \frac{10}{\sqrt{3n}}$ nS we need only the variance part

$$\sigma_{dig} = \pm \frac{10}{\sqrt{3n}} \text{ nS}$$
$$\sigma_{binom} = \frac{76.94}{\sqrt{n}} \text{nS}$$

Table 5 show all the errors except σ_p, σ_{ms}, and σ_{tts}, which are not data-dependent and are the same for all data sets.

Since we believe that all these sources of error are independent, the correct way to combine all the sigmas is to add the variances (σ^2) and take the square root of that. This will give us one standard deviation for the whole data set. However, δ_{clock} should not be combined in this way since it represents an uncertainty about the time measurements.

For our combined data set Z_{all}, we get

$$\sigma^2 = \sigma_{samp}^2 + \sigma_{trig}^2 + \sigma_p^2 + \sigma_{ms}^2 + \sigma_{tts}^2 + \sigma_{dig}^2 + \sigma_{binom}^2 = 132.99$$

TABLE 5. Corrected lifetime results, with errors

	τ	σ_{trig}	σ_{samp}	σ_{dig}	δ_{clock}	σ_{binom}
D	1991.95	±1.55	±21.81	±0.06	±0.20	±0.84
H	1985.46	±1.52	±13.98	±0.04	±0.20	±0.54
M	1903.08	±1.53	±16.70	±0.05	±0.19	±0.68
S	1981.91	±1.65	±41.06	±0.12	±0.20	±1.59
Y	2021.04	±1.57	±25.06	±0.07	±0.20	±0.95
Z1	2040.65	±1.57	±25.64	±0.07	±0.20	±0.97
Z2	1977.26	±1.55	±20.11	±0.06	±0.20	±0.78
Z3	2056.30	±1.52	±14.52	±0.04	±0.21	±0.54
Z_{all}	2031.79	±1.50	±10.70	±0.03	±0.20	±0.41

$$\sigma = \sqrt{132.99} = 11.53 \text{ nS}$$

We can see from this that the sampling error dominates. A 95% confidence interval for this number, assuming the error is normally distributed, would be $\pm 1.96\sigma$. That is, based on the Z data sets only, we should be 95% confident that the actual τ from the mixture of μ^+ and μ^- is between

$$2031.79 - 1.96\sigma - \delta_{clock} = 2008.98 < \tau < 2054.60 = 2031.79 + 1.96\sigma + \delta_{clock}$$

However, 4 out of 5 student data sets fall below this confidence interval, which is disturbing. There may well be sources of error or bias for which we have not accounted, or not accounted properly.

Given $\tau_+ = 2197$ nS and $\tau_- = 2043$ nS from the literature, and μ^+/μ^- ratios from 1.08 to 1.3, we would expect the "real" value to be in the range 2123 nS to 2130 nS. This is also well outside my confidence interval.

5.7. **Gradient descent: Speedup.** The long runtimes with the above implementation seemed unreasonable. Based on the observation that ν converged much more quickly than the other parameters, but became unstable if α was increased, I tried increasing α while decreasing the $\partial V/\partial \nu$ adjustment by the same ratio. This had the net effect of keeping the magnitude of corrections to ν the same while increasing the magnitude of corrections to the other parameters.

I saw a substantial improvement in rate of convergence with this change, even with α only a few times larger. With $\alpha = 0.18$, 200 times higher, I got excellent convergence in 15000 iterations (6.8 seconds run time). The solutions were identical to the previous ones to within a few parts per million. Source code for this version is included later in this report.

It is possible that the convergence problem was caused by the difference in size between the various parameters. Andrew Ng (Stanford), in his online Machine Learning course on `coursera.org` (which I took this last spring), recommends scaling different features to be of comparable magnitude before applying gradient descent, but I (perhaps foolishly) did not follow his advice. I have not yet had

time to explore this issue further. The *ad hoc* adjustment I made might well have had pretty much the same effect anyway.

6. Summary and discussion

I ran the standard Teachspin muon lifetime experiment, and collected 3 data sets with a total of 38682 recorded decay events. Since the data were already grouped into 20 nS bins by the hardware, I fit this to a model consisting of a geometric decay (but with parameters N and τ corresponding to the underlying exponential) and a uniform background noise with intensity ν. The fit estimated that about 2398 of the "decays" were due to background, leaving 36284 real decays. It also gave $N = 36044$, which seems slightly low, since if we started with 36044 muons we would expect a couple of them to not decay within the 20 μS window. The effective decay lifetime τ was estimated to be $2031.79 \pm 11.53(\sigma) \pm 0.20(clock)$, with a 95% confidence interval of $2008.98 < \tau < 2054.60$. The largest contributor to σ was the sampling error; this implies that we should be able to get tighter results by collecting even more data, up to roughly 2M events, at which point other sources of error like triggering uncertainty would begin to dominate.

This all seems fine, but I also analyzed the student data sets from the Spring 2012 course, and 4 out of 5 of the large ones that appeared to have valid data gave τ below this confidence interval. So it would appear that we need to invoke some other source of error to explain those results. A seasonal variation in μ^+/μ^- ratio might be one such source. Some bias from different equipment settings might be another. Also disturbing is that the "real" value of τ (2123-2130 nS) is well above the confidence interval.

In terms of the suitability of the Teachspin device for my own proposed QTD experiment, the scintillator module seems adequate. It is already configured to run off a DC power supply, so adapting it for use with batteries would not be hard. It is rather small for use with cosmogenic muons, giving a fairly low event rate, but should be fine with a muon beam from an accelerator. It cannot distinguish μ^+ from μ^-, but again this is OK with a pure beam of one type. (For cosmic rays we would prefer a higher-Z material like CsI or NaI, where nuclear capture of μ^- proceeds rapidly, leaving mostly μ^+ to decay normally.) It might be desirable to add paddles before and after the main detector in order to filter out transit events more easily.

The remainder of the device is problematic. The 20 nS clock speed could be improved. Even at that speed, decays shorter than about 40-50 nS are not detected, forcing us to discard all decays shorter than 60 nS, which might have given useful information about e.g. pion contamination. No information about pulse shape or size is retained, making sophisticated analysis of the data impossible. Some data is lost; it appears the communication protocol between the FPGA and host computer is not reliable. It would be better to use faster, more sophisticated, and more reliable equipment.

Appendix A. Source code

We include here the source to all programs written for this project.

A.1. **teachspin.perl : data array generator source code in Perl.** This reads a Teachspin data file and converts it to a count file containing a list of numbers giving the number of decay events seen for each 20 nS bin.

```perl
#!/usr/bin/perl
# Parse a data file from a Teachspin experiment
# and convert it into counts of the number of
# decay events at each bin time.
# Also perform some sanity checks on the data.

# Split filename for later use.
($root,$suffix) = split(/\./,$ARGV[0]);
if ($suffix ne "data")
{
        print STDERR "WARNING: bad filename: $ARGV[0]\n";
}

$events = 0;            # total events seen
$seconds = 0;           # total number of 1-second summaries
$min_decay = 99999;     # ~100 uS, longer than decay window
$max_decay = 0;
$decay_window = 20000;  # nS, = 20 uS
$bin_size = 20;         # nS
$tau_nominal = 2200;    # nS
$start_time = -1;
$end_time = -1;

while (<>)
{
        if (/^4(\d\d\d\d) (\d+)/)
        {
                # 1-second summary record
                $n = $1;        # number of events
                $t = $2;        # timestamp
                # Keep track of earliest and latest timestamps
                # All timestamps appear to be generated in order,
                # but we code as if that might not be true.
                if (($start_time < 0) || ($t < $start_time))
                {
                        $start_time = $t;
                }
                if ($t > $end_time)
                {
```

```perl
40                         $end_time = $t;
41                 }
42                 $events += $n;
43                 $seconds ++;
44         }
45         elsif (/(\d+) \d+/)
46         {
47                 # Decay event record.
48                 # $1 is the decay time in nS.
49                 # It should always be a multiple of 20 nS.
50                 $bin = int(0.5 + $1 / $bin_size);
51                 if ($1 != $bin * $bin_size)
52                 {
53                         print "Rounding error on $1\n";
54                 }
55                 $decay[$bin] ++;
56                 $decays ++;
57                 if ($1 < $min_decay)
58                 {
59                         $min_decay = $1;
60                 }
61                 if ($1 > $max_decay)
62                 {
63                         $max_decay = $1;
64                 }
65         }
66         else
67         {
68                 print "Couldn't parse: $_";
69         }
70 }
71
72 # Check seconds count
73 $time = $end_time - $start_time + 1;
74 if ($seconds != $time)
75 {
76         $diff = $time - $seconds;
77         print "Timing mismatch: $seconds seconds vs ";
78         print "$time computed time ($diff missing)\n" ;
79 }
80
81 # Calculate estimated noise distribution
82 # based on number of transit events.
83 $arrival_rate = $events / $seconds;
84 print "$events events in $seconds seconds\n";
```

```perl
85   print "Average␣arrival␣rate␣$arrival_rate␣events/second\n" ;
86   $decay_rate = $decays / $seconds;
87   print "$decays␣decays\n";
88   print "Average␣decay␣rate␣$decay_rate␣decays/second\n" ;
89
90   # Let's look at that issue:
91   print "Shortest␣decay␣was␣$min_decay␣nS\n";
92   # On most large datasets this gives 40 nS,
93   # so we ARE losing short decay events.
94   #print "Longest decay was $max_decay nS\n";
95   # It looks like the device is rounding to the nearest 20 nS
96   # and reports zero events for 0 nS and 20 nS bins.
97   # Also, the 40 nS bin usually has fewer events than the
98   # 60 nS bin, so we are probably losing some 40 nS events.
99
100  $n_bins = $decay_window / $bin_size;
101  print "$n_bins␣bins␣of␣$bin_size␣nS␣each␣";
102  print "in␣a␣window␣of␣$decay_window␣nS\n";
103
104  # Fix the beginning problems by cutting off
105  # the initial empty bins and the first nonempty bin
106  # (which may only be partially full).
107  # For consistency delete first 3 bins from all datasets.
108  for ($i = 1; $i <= 3; $i++)
109  {
110          $n = 0 + shift(@decay);
111          print "Discarded␣initial␣bin␣with␣$n␣decays\n";
112          $decays -= $n;
113          $n_bins --;
114  }
115  print "$decays␣decays␣in␣$n_bins␣bins␣remaining\n";
116
117  # Construct output data file name.
118  $COUNT_FILE = "$root.count";
119  # Write output data.
120  open(COUNT_FILE,">$COUNT_FILE");
121  for ($i = 0; $i < 997; $i++)
122  {
123          print COUNT_FILE (0 + $decay[$i]), "\n";
124  }
125  close(COUNT_FILE);
126  print "Wrote␣997␣lines␣to␣$COUNT_FILE\n";
```

A.2. **gd2.m : gradient descent fitter source code in Octave.** This takes the count file data and fits a decay curve plus background noise to it.

```octave
1   % Do gradient descent optimization to fit muon lifetime,
2   % and a uniform background noise
3   % to normalized data from a Teachspin experiment.
4   % AUTHOR: Howard A. Landman, Oct 2012
5   %
6   % This was written and tested in Octave, but uses only basic
7   % features, and should be compatible with Matlab as well.
8
9   % Load Teachspin data as preprocessed by teachspin.perl.
10  fprintf('loading data.txt ...');
11  Y = load('data.txt');    % Should contain 997 bin counts,
12  bins = length(Y);        % but we don't hard-code that anywhere.
13  decays = sum(Y);         % total (true and false) decay events
14  fprintf('done, %d decays in %d bins\n',decays,bins);
15
16  % Set up for gradient descent.
17  b = 20;          % nS, bin size
18  alpha = 0.18;    % gradient descent rate-of-change factor
19                   % lower is slower but higher may not converge
20  num_iters = 15000;
21
22  % initial guesses for tau, nu, and n, all floating point
23  tau = 2000.0; % nS
24  END_Y = Y((bins-99):bins); % estimate noise from last 100 bins
25  s = sum(END_Y);
26  nu = 0.0 + s / 100.0;    % expected number of decays in each bin
27                           % nu*bins = est. number of false decays
28  n_real = 0.0 + decays - nu*bins;       % est. number of real decays
29  % Since the fraction of muons that should decay in our window
30  % is 1 - exp(-bins*b/tau), we can estimate the total muons.
31  n = n_real / ( 1 - exp(-bins*b/tau));
32
33  K = 0:(bins-1); % row vector 0 to 996
34  K = K';         % column vector 0 to 996
35
36  % Hypothesis -
37  % what we predict for each bin, based on tau nu and n.
38  % Note that this is a vector (one number per bin).
39  p = 1 - exp(-b/tau);    % prob. that a muon decays in 1 bin time
40  H = zeros(size(K));
41  H = nu + n*p*((1-p) .^ K);
42
43  % error
44  D = H - Y;
45  E = D .^ 2;
```

```
46  v = sum(E);
47  fprintf('Initial␣settings\n');
48  fprintf('n=%f,␣tau=%f,␣nu=%f,␣v=%f\n',n,tau,nu,v);
49
50  % Begin main loop.
51  for iter = 1:num_iters
52  %fprintf('iter=%d\n',iter);
53
54  % We could solve for nu exactly with this code,
55  % since its gradient has a particularly simple form,
56  % but choose instead to treat it the same as tau and n.
57  %fprintf('Solve for best nu\n');
58  %delta_nu = sum(D) / bins;
59  %nu = nu + delta_nu ;
60  % Update hypothesis and recalculate error
61  %H = nu + n*p*((1-p) .^ K);
62  %D = H - Y;
63  %E = D .^ 2;
64  %v = sum(E);
65  %fprintf('nu: iter=%d, n=%f, tau=%f, nu=%f, v=%f\n',
66  %        iter,n,tau,nu,v);
67
68  % Calculate one step of gradient descent for nu, n and tau.
69  % See the discussion in the main paper for derivations.
70
71  % dV/dnu
72  dvdnu = 2*sum(D); % This points UPHILL but we want to go DOWNHILL
73  delta_nu = -alpha*dvdnu; % hence the minus sign here.
74  delta_nu = delta_nu/200.0;
75  % NOTE: nu converged much more quickly than the other parameters
76  % but became unstable when alpha > 0.0009.
77  % Keeping alpha this low led to long run times (~82 minutes).
78  % The divide by 200 here allows us to use alpha = 0.18,
79  % which gives much faster convergence on tau and n,
80  % while keeping the convergence of nu stable.
81  % Run times are now typically under 10 seconds.
82
83  % Cross-check dV/dnu calculation (for debugging).
84  %Hplus = (nu + 0.001) + n*p*((1-p) .^ K);
85  %Hminus = (nu - 0.001) + n*p*((1-p) .^ K);
86  %Dplus = Hplus - Y;
87  %Dminus = Hminus - Y;
88  %Eplus = Dplus .^ 2;
89  %Eminus = Dminus .^ 2;
90  %Vplus = sum(Eplus);
```

```
91    %Vminus = sum(Eminus);
92    %dvdnu_est = (Vplus - Vminus) / 0.002
93    %fprintf('dvdnu=%f, dvdnu_est=%f\n',dvdnu,dvdnu_est);
94
95    % dV/dn
96    DN = D .* ((1-p) .^ K);
97    dvdn = 2*p*sum(DN);
98    delta_n = -alpha*dvdn;
99
100   % Cross-check dV/dn calculation (for debugging).
101   %Hplus = nu + (n + 0.001)*p*((1-p) .^ K);
102   %Hminus = nu + (n - 0.001)*p*((1-p) .^ K);
103   %Dplus = Hplus - Y;
104   %Dminus = Hminus - Y;
105   %Eplus = Dplus .^ 2;
106   %Eminus = Dminus .^ 2;
107   %Vplus = sum(Eplus);
108   %Vminus = sum(Eminus);
109   %dvdn_est = (Vplus - Vminus) / 0.002
110   %fprintf('dvdn=%f, dvdn_est=%f\n',dvdn,dvdn_est);
111
112   % dV/dtau
113   PK = (1-p) .^ K;
114   DTAU = D .* ((PK .* (1-p)) - p*(K .* PK));
115   dvdtau = (-2*n*b / (tau*tau)) * sum(DTAU);
116   delta_tau = -alpha*dvdtau;
117
118   % Cross-check dV/dtau calculation (for debugging).
119   %pplus = 1 - exp(-b/(tau+0.001));
120   %pminus = 1 - exp(-b/(tau-0.001));
121   %Hplus = nu + n*pplus*((1-pplus) .^ K);
122   %Hminus = nu + n*pminus*((1-pminus) .^ K);
123   %Dplus = Hplus - Y;
124   %Dminus = Hminus - Y;
125   %Eplus = Dplus .^ 2;
126   %Eminus = Dminus .^ 2;
127   %Vplus = sum(Eplus);
128   %Vminus = sum(Eminus);
129   %dvdtau_est = (Vplus - Vminus) / 0.002;
130   %fprintf('dvdtau=%f, dvdtau_est=%f\n',dvdtau,dvdtau_est);
131
132   % Update simultaneously.
133   nu = nu + delta_nu;
134   n = n + delta_n;
135   tau = tau + delta_tau;
```

```
136
137   v_old = v;
138   % Update hypothesis and recalculate error
139   p = 1 - exp(-b/tau);
140   H = nu + n*p*((1-p) .^ K);
141   D = H - Y;
142   E = D .^ 2;
143   v = sum(E);
144   v_diff = v - v_old;
145
146   % For iteration-by-iteration statistics, just move the
147   % following fprintf statement inside the loop here.
148   % With 10 million iterations, this is too verbose.
149
150   end
151
152   % Print final results
153   fprintf('iter=%d,␣n=%f,␣tau=%f␣(p=%f),␣nu=%f,␣v=%f,␣delta=%f\n',
154           iter,n,tau,p,nu,v,v_diff);
```

A.3. **h2tex.perl : hypothesis curve generator source code in Perl.** This takes the parameters fit by gd2.m and prints out the detailed error estimates for Table 5, and also writes a data file containing the bin-by-bin predictions of the hypothesis in the same format as the count file.

```
1   #!/usr/bin/perl
2   # Take the hypothesis parameters from a gradient descent fit
3   # of Teachspin data, and write out a detailed error analysis
4   # in TeX format for the paper.
5
6   sub do_line {
7   my($a,$n,$tau,$nu) = @_;
8   print "dataset=$a,␣n=$n,␣tau=$tau,␣nu=$nu␣(nu␣is␣currently␣ignored)\n";
9
10  $b = 20;          # bin size in nS
11
12  $i = 0;
13
14  # delta_trig
15  #$name[$i] = '$\delta_{trig}$';
16  $name[$i] = '$\sigma_{trig}$';
17  #$offset[$i] = -14.5;
18  $offset[$i] = 0.0;
19  #$trig_offset = -14.5;
20  $trig_offset = 0.0;
21  $sigma[$i] = 1.45 + 9.5/sqrt($n);
```

```perl
22  $i++;
23
24  # sigma_p
25  #$name[$i] = '$\sigma_{p}$';
26  #$offset[$i] = 0.0;
27  #$sigma[$i] = 4.0;
28  #$i++;
29
30  # sigma_ms
31  #$name[$i] = '$\sigma_{ms}$';
32  #$offset[$i] = 0.0;
33  #$sigma[$i] = 0.1;
34  #$i++;
35
36  # delta_dig
37  $name[$i] = '$\delta_{dig}$';
38  #$offset[$i] = $tau - $b/(exp($b/$tau) - 1);
39  $offset[$i] = 0.0;
40  $sigma[$i] = 0.5*$b/sqrt(3*$n);
41  $i++;
42
43  # delta_clock
44  $name[$i] = '$\delta_{ck}$';
45  $offset[$i] = 0.0;
46  $sigma[$i] = 0.0001*$tau;
47  $i++;
48
49  # sigma_binom
50  $name[$i] = '$\sigma_{binom}$';
51  $offset[$i] = 0.0;
52  $sigma[$i] = 76.94/sqrt($n);
53  $i++;
54
55  # print them all out
56  $i_max = $i;
57
58  # header line, first time only
59  if (!$header_printed)
60  {
61          print $a;
62          print ' & $\tau_{adj}$ ';
63          for ($i=0; $i<$i_max; $i++)
64          {
65                  print "& $name[$i] ";
66          }
```

```
67                print "\n";
68                $header_printed = 1;
69     }
70
71     # data line
72     print '␣&␣', $tau + $trig_offset, '␣&␣$';
73     for ($i=0; $i<$i_max; $i++)
74     {
75                if ($offset[$i] != 0.0)
76                {
77                        print $offset[$i];
78                }
79                if ($sigma[$i] != 0.0)
80                {
81                        print "\\pm$sigma[$i]";
82                }
83                print '$␣';
84     }
85     print "\n";
86
87     # Additionally, write out a detailed bin-by-bin model of the
88     # hypothesis in the same format as the summary data files.
89                $one_minus_p = exp(-$b/$tau);
90                $p = 1 - $one_minus_p;
91                $decay = $p * $n;          # number expected to decay in first bin
92                $AFILE = "$a.hyp";
93                open(AFILE,">$AFILE");
94                for ($i=0; $i<997; $i++)
95                {
96                        $hyp = $decay + $nu;
97                        print AFILE "$hyp\n";
98                        $decay *= $one_minus_p;
99                }
100               close AFILE;
101    }
102
103    # Process all data sets.
104    &do_line('D',    8340.885384, 1991.954669,  0.602296);
105    &do_line('E',    3905.128384, 1753.169409, -0.760365);
106    &do_line('H',   20161.148389, 1985.457413,  1.446067);
107    &do_line('M',   12984.064489, 1903.078861,  0.819760);
108    &do_line('P',   25717.742580, 1941.212151, -5.703965);
109    &do_line('S',    2330.060498, 1981.906108,  0.232737);
110    &do_line('Y',    6502.150953, 2021.043352,  0.562875);
111    &do_line('Z1',   6332.506662, 2040.646386,  0.348901);
```

```
112  &do_line('Z2',   9664.946710, 1977.263647, 0.733657);
113  &do_line('Z3', 20049.800011, 2056.299940, 1.319391);
114  &do_line('ZZ', 36044.133506, 2031.792283, 2.405053);
```

A.4. **myplot.R : plot the data and fitted curve in R.** This takes the count file data and the hypothesis file data and plots them on top of each other. File names would need to be changed for other data sets.

```R
1   ## Plot our data from teachspin
2   par(bg="white")
3
4   setwd("~/bin/Teachspin/data")
5   ## Read actual data.
6   ZZY <- t(read.delim("howard.count",header=FALSE))
7   ## Read hypothesis data.
8   ZZy <- t(read.delim("ZZ.hyp",header=FALSE))
9   x <- 1:length(ZZY)
10  ## time vector in nS
11  t <- 20*x
12
13  ## Create PDF file of plot, with font matching TeX font.
14  pdf("ZZ.pdf",family="Times")
15  ## Create plot and plot data in it.
16  plot(t,ZZY,main="Decay time data and best exponential+noise fit",
17       xlab="Decay Time (nS)",
18       ylab="Number of events per 20 nS bin",cex=0.4)
19  ## Plot hypothesis curve on top of data.
20  lines(t,ZZy,col="red",lwd=1.5)
21  ## Plot zero line for comparison with tail noise level.
22  zeros <- 0*x
23  lines(x,zeros,col="blue",lwd=1.5)
24  ## Close the file.
25  dev.off()
```

APPENDIX B. SRIM/TRIM SIMULATIONS

To help understand the behavior of muons entering the detector, I ran dozens of simulations using SRIM/TRIM software[7], which is designed for ion implantation studies. SRIM-2008 algorithms were used. TRIM does not include a muon model; it appears that the best way to model one is to treat it as a hydrogen ion (i.e. a proton) and adjust the mass downward to 0.1135 AMU. I modeled the Teachspin scintillator module as 6.5 mm of aluminum, followed by 152 mm of plastic, followed by another 6.5 mm of aluminum. For some simulations, this was preceded by a thin layer of metal (Al or Fe) representing a VDGG shell and 1 or 2 layers of air, so as to model scattering and energy loss before arriving at the detector. This would be

reasonable for a narrow centered beam entering the cylinder horizontally; I made no attempt to model the curvature of the cylinder, or the PMT/circuitry portion of the scintillator module (which would affect muons entering vertically). Since polyvinyltoluene was not available in the materials library, I used polystyrene as a guide for how to model the plastic; it has similar density and elemental composition and bond structure.

For a monochromatic beam of muons, the depth of penetration was surprisingly well-controlled. For momenta between about 25 MeV/c and 62 MeV/c, nearly all muons were stopped in the plastic. However at 70 MeV/c and higher, nearly all muons exited the back of the detector. The following figure shows the results for 10K simulated muons at 51 MeV/c, including a metal sphere layer and 1 air gap. (it may be easier to read this figure by zooming in to it.) These numbers are at
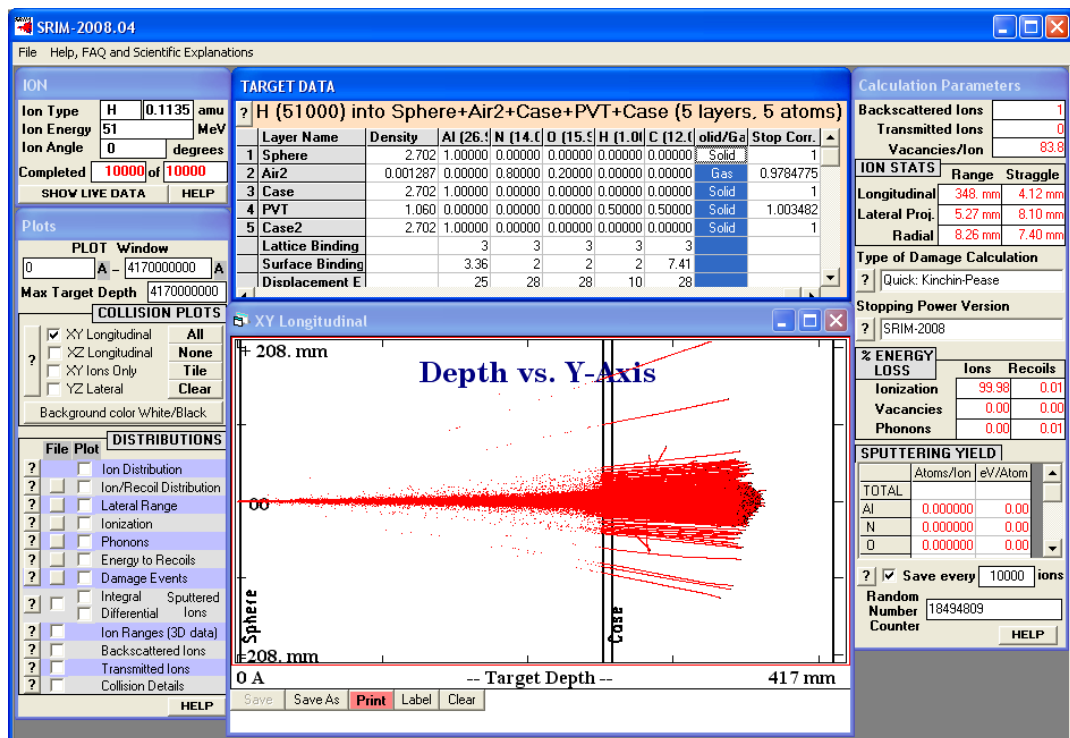
odds with the stopping power claimed in the Teachspin documentation, which is 120 MeV/c in one place and 180 MeV/c in another. These higher numbers may be for vertical muon entry, which I did not model.

References

[1] D. Apsel, "Gravitational, electromagnetic, and nuclear theory ", *International Journal of Theoretical Physics* v.17 #8 643-649 (Aug 1978) DOI: 10.1007/BF00673015

[2] D. Apsel, "Gravitation and electromagnetism", *General Relativity and Gravitation* v.10 #4 297-306 (Mar 1979) DOI: 10.1007/BF00759487

[3] D. Apsel, "Time dilations in bound muon decay", *General Relativity and Gravitation* v.13 #6 605-607 (Jun 1981) DOI: 10.1007/BF00757247

[4] H. Landman, "An Elementary Approach to Quantum Time Dilation", `http://www.riverrock.org/~howard/QuantumTime21.pdf` An earlier version of this was posted to the Arxiv server but immediately removed by site administrators.

[5] T. Coan, T. Liu, J. Ye, "A Compact Apparatus for Muon Lifetime Measurement and Time Dilation Demonstration in the Undergraduate Laboratory", *Am.J.Phys.* 74, 161-164 (2006) arXiv:physics/0502103v1

[6] `http://www.teachspin.com/instruments/muon_physics/`

[7] `http://www.srim.org`

[8] "Lifetime of the muon", Physics 77 experiment 15 handout, Caltech (Sep 2000) `http://www.pma.caltech.edu/~ph77/labs/exp15.pdf`

[9] "Muon Lifetime", Physics 111 wiki, U C Berkeley, `http://labs.physics.berkeley.edu/mediawiki/index.php/Muon_Lifetime`

[10] C. W. Akerlof, "Measurement of the Muon Lifetime", `https://instructor.physics.lsa.umich.edu/adv-labs/Muon_Lifetime/MuonLifetime.pdf`

[11] A. Hansen, "Measurement of Muon Lifetime and Mass Using Cosmic Ray Showers", Physics 4052 report, U. Minnesota (4 May 2001), `http://mxp.physics.umn.edu/s01/Projects/Muon/muonfinal1c.pdf`

[12] "Cosmic Ray Muon Detectors: Particle Physics Using Nature's Accelerator" `http://quarknet.fnal.gov/toolkits/new/crdetectors.html`

[13] "QuarkNetter Reports: The Fermilab Detector", `http://quarknet.fnal.gov/toolkits/new/fnaldet.html`

[14] `http://en.wikipedia.org/wiki/Exponential_distribution`

[15] J. I. Gold & M. N. Shadlen, "The Neural Basis of Decision Making", *Annu. Rev. Neurosci.* 30:53574 (2007) doi: 10.1146/annurev.neuro.29.051605.113038

[16] G. M. El-Sayyad, "Information and Sampling from the Exponential Distribution", *Technometrics* v.11 #1 41-45 (Feb 1969)

[17] F.M. Al-Athari, "Estimation of the Mean of Truncated Exponential Distribution", *Journal of Mathematics and Statistics* 4(4) 284-288 (2008)

[18] `http://en.wikipedia.org/wiki/Uniform_distribution_(continuous)`

[19] MuLan Collaboration (Chitwood et al.), "Improved Measurement of the Positive Muon Lifetime and Determination of the Fermi Constant", *Phys. Rev. Lett.* 99:032001, (2007), arXiv:0704.1981. Bibcode 2007PhRvL..99c2001C, doi:10.1103/PhysRevLett.99.032001

[20] W. Becker & A. Bergmann, "Detectors for High-Speed Photon Counting", (2003) `http://www.boselec.com/products/documents/FastDetectorsandSources.pdf` or `http://www.scribd.com/doc/15722917/85/Transit-Time-Spread`

[21] `http://www.gnu.org/software/octave/`

[22] `http://en.wikipedia.org/wiki/GNU_Octave`