Investigation of Combined Positive and Negative Muon Decay in a Scintillator

Appendices

Brian Lim and D. Ruben

Cornell University, Ithaca, NY 14853, USA

(received 7 November 2005)

Appendix A: Weighted Least Squares Fitting

Developed from Taylor's discourse[16], the following derives the determination of the slope of a linear fit to data where each dependent variable, y_i, have different errors, σ_i . Assuming no error in the independent variable, x_i, the probability of obtaining y_i is

$$Prob_{A,B}(y_i) \propto \frac{1}{\sigma_i} e^{-(y_i - A - Bx_i)^2/2\sigma_i^2}$$

The probability of obtaining the set of y_i's is then

$$\begin{aligned} Prob_{A,B}(y_1, \ldots, y_N) &= Prob_{A,B}(y_1) \cdots Prob_{A,B}(y_N) \\ &\propto \frac{1}{\sigma_1} e^{-(y_1 - A - Bx_1)^2/2\sigma_1^2} \cdots \frac{1}{\sigma_N} e^{-(y_N - A - Bx_N)^2/2\sigma_N^2} \\ &= \frac{1}{\sigma_1 \cdots \sigma_N} e^{-\chi^2/2} \end{aligned}$$

where $\chi^2 = \sum_{i=1}^{N} \frac{(y_i - A - Bx_i)^2}{\sigma_i^2}$. The best fit would have χ^2 minimized by setting its partial derivatives with respect to A and B equal to zero and solving,

$$\frac{\partial \chi^2}{\partial A} = -2 \sum_{i=1}^{N} \frac{1}{\sigma_i^2} (y_i - A - Bx_i) = 0$$
$$\frac{\partial \chi^2}{\partial B} = -2 \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} (y_i - A - Bx_i) = 0$$

where $w_i = \frac{1}{\sigma_i^2}$ to give

$$A = \frac{\sum wx^2 \sum wy - \sum wx \sum wxy}{\Delta} \qquad \text{(A.1)}$$

$$B = \frac{\sum w \sum wxy - \sum wx \sum wy}{\Delta} = \frac{\sum \left[S_w w_i x_i - S_{wx} w_i \right] y_i}{\Delta} \qquad \text{(A.2)}$$

where $\Delta = \sum w \sum wx^2 - \left(\sum wx \right)^2$, $S_w = \sum w$, $S_{wx} = \sum wx$.

Using the method of quadrature for error propagation, the error in B can be found.

$$
\begin{aligned}
\delta B &= \sqrt{\sum \left(\frac{\partial B}{\partial y_i} \delta y_i \right)^2} = \sqrt{\sum \left(\frac{\partial B}{\partial y_i} \right)^2 \frac{1}{w_i}} \\
&= \sqrt{\sum \left(\frac{S_w w_i x_i - S_{wx} w_i}{\Delta} \right)^2 \frac{1}{w_i}} \\
&= \sqrt{\frac{1}{\Delta^2} \sum \left(S_w^2 w_i x_i^2 + S_{wx}^2 w_i - 2 S_w S_{wx} w_i x_i \right)} \\
&= \sqrt{\frac{1}{\Delta^2} \left(S_w^2 S_{wxx} + S_{wx}^2 S_w - 2 S_w S_{wx}^2 \right)} \\
&= \sqrt{\frac{S_w}{\Delta^2} \left(S_w S_{wxx} - S_{wx}^2 \right)} = \sqrt{\frac{\sum w}{\Delta^2} \Delta} \\
&= \sqrt{\frac{\sum w}{\Delta}}
\end{aligned}
\tag{A.3}
$$

Appendix B: Data Files

For data collected after 7th October 2005, the PMTs were set at 1.005V, at the middle of the voltage plateau of the scintillator/PMT setup for most accurate counting of muon detection.

oct12.txt

```
   2     0     0     0     0    28  1678  1870  1743  1706  1579  1577
1489  1341  1356  1278  1205  1198  1083  1057   963   934   941   865
 845   823   777   744   665   614   635   598   591   585   548   512
 506   436   428   392   433   410   351   328   329   315   327   328
 293   287   240   243   222   226   234   224   221   209   175   187
 171   175   182   154   168   115   135   119   117   125   119   123
  93   100    88   108   103    77    89    95    85    80    71    71
  72    74    56    67    69    66    64    71    63    42    47    35
  57    52    51    37    51    54    39    30    39    48    42    36
  39    41    29    38    18    24    36    28    29    32    29    31
  36    41    43    33    36    35    26    31    34    35    31    23
  32    38    29    25    25    25    18    29    22    23    20    25
  24    19    29    25    32    17    23    25    28    23    20    26
  35    20    25    32    29    24    31    19    29    24    29    24
  24    27    18    31    23    21    23    30    27    25    27    31
  22    32    24    20    30    25    19    30    31    23    30    27
  24    18     8     0     0     0     0     0     1     0     1     0
   0     0     0     0     0     1     0     0     0     0     0     0
   0     0     0     0     0     0     0     0     0     0     0     0
   1     0     0     0     0     0     0     0     1     0     0     1
   0     0     0     0     0     0     0     0     0     3     0     0
   0     1     0     0
423399
```

oct14.txt

```
   2     0     0     0     0    11   607   670   589   594   594   592
 527   551   494   440   415   406   382   365   379   316   308   312
 271   287   264   244   257   268   213   205   200   202   168   180
 162   161   162   145   137   138   116   150   109   133   109   104
 107   114    87   101    89    92    79    83    58    75    68    63
  53    65    60    54    41    51    59    52    51    36    39    51
  43    35    40    36    37    19    39    32    21    21    27    30
  21    29    23    20    26    19    19    23    16    25    18    18
  22    16    19    15    19    13    18    13    12    18    19    12
   9    18    22    12    13    13    12    14     5     7    11    19
  13    17    20    12    15    16    11     4    11     8    11     7
  16    11     7    11     9    10    11     3    10    10     5     6
  12    15     5     7    12     5    14     7    14     9     6    10
  16    10    13    10     7     7     3    13     9     7     8     9
  11     6     5     8     9    10    12     8    13    10    11    13
   7     9     8     6     7     6     6    11     7     9    10     8
   4     5     3     0     0     0     0     0     0     0     0     0
   0     0     0     0     1     0     0     0     0     0     0     0
   0     0     0     1     0     0     0     0     0     0     0     0
   0     0     0     0     0     0     0     0     0     0     0     0
   0     0     0     0     0     0     0     0     0     1     0     0
   0     0     0     0
153504
```

oct26.txt

```
  11     0     0     0     0    89  4278  4842  4393  4298  4148  3942
3818  3768  3340  3353  3039  2891  2891  2694  2629  2491  2353  2249
2192  2033  1946  1873  1819  1704  1610  1575  1491  1328  1378  1312
1231  1157  1077  1089  1017   993   990   839   869   791   793   737
 723   658   649   640   621   620   583   546   539   529   483   440
 475   394   405   384   358   339   356   348   337   303   277   283
 270   259   248   264   227   203   210   219   195   212   190   148
 176   183   182   161   184   153   141   154   132   141   141   127
 125   118   120   137   109   102   113   113   130   105   107    88
 112    91    96    97    77   103    91    90    93    81   106    85
  85    74    71    77    76    76    87    67    82    68    87    79
  60    73    74    72    73    69    70    76    77    73    59    63
  64    68    67    82    70    66    70    64    63    64    65    64
  74    57    63    72    70    62    68    55    64    63    57    61
  47    65    53    68    59    55    70    74    63    62    76    74
  69    48    56    66    66    65    55    59    64    53    72    56
  66    43    12     0     0     1     0     0     0     0     0     0
   0     2     0     0     1     1     0     1     1     0     1     1
   0     0     1     0     0     0     1     0     2     0     0     0
   0     0     1     0     0     0     1     0     0     1     0     1
   2     0     1     0     1     1     1     1     0     0     1     0
   0     1     2     0
1036965
```

nov01.txt

```
   9     0     0     0     0    62  2260  2401  2217  2278  2110  2040
1881  1826  1718  1666  1639  1553  1455  1364  1293  1213  1252  1129
1090   991  1022   919   903   860   846   797   752   733   781   644
 662   639   567   489   521   446   516   502   464   432   427   393
 386   376   332   316   323   296   290   299   270   270   244   243
 233   202   221   200   199   185   176   165   151   168   154   156
 141   123   137   136   125   138   108    93    89    92    91   101
 102    93   101    70    80    71    82    73    66    61    58    74
  54    67    47    52    67    51    52    49    66    54    46    55
  39    57    48    44    49    41    43    40    54    47    44    40
  46    66    44    38    56    41    54    43    46    34    42    42
  31    42    37    36    35    32    29    32    31    40    33    31
  27    38    46    24    34    30    26    27    21    40    29    28
  30    36    35    35    41    37    31    23    34    32    24    38
  37    27    28    34    29    33    44    28    26    29    43    23
  30    30    30    25    21    19    40    39    28    39    36    30
  24    24    10     0     0     0     1     0     0     0     0     0
   0     0     0     0     1     0     0     0     0     0     1     0
   0     0     0     0     0     0     0     0     0     1     0     0
   0     1     0     0     1     0     0     0     1     0     0     0
   0     0     1     0     0     1     0     0     0     0     0     0
   0     0     0     0
 536912
```

The following data was taken with the PMT calibrated at 9.963V and was not used in determining the muon lifetime.

oct07.txt

```
  1     0     0     0     0    11   690   750   710   663   646   595
601   583   529   478   497   444   414   415   412   422   384   328
325   349   279   316   269   276   243   267   234   253   213   237
188   180   181   186   157   168   147   122   139   134   137   111
104   114   113   112   114   115    67    77    88    93    69    75
 75    70    60    61    56    51    67    42    49    40    49    50
 49    39    36    45    41    32    41    26    29    24    20    23
 30    29    28    29    28    17    27    27    25    19    19    22
 24    17     9    19    13    13    23    17    21    14    23    22
 15     8    16    15    18    15    13    21    17    16    12    10
 15     9    10    17    13    15    11    21    14    14    18     7
 15    15    12    11    13    12    14     8    15    11    10    14
 12     9    12    10     9     8    10     7    13    11    16    13
 12    10     8    15    11    12     7    18     4    13     4    14
 10    13    10    12     8    12    14    10     6     7    11     9
 15    10    12    11     8    13    13    13     5     8    15    10
 11     9     1     0     0     0     0     0     0     0     0     0
  0     0     0     0     0     0     0     0     0     0     0     0
  0     0     0     0     1     0     0     0     0     0     0     0
  0     0     0     0     0     0     0     0     0     0     0     0
  0     0     0     0     0     0     0     0     0     0     0     1
  1     0     0     0
221815
```

Appendix C: Matlab Code

lab02.m

```
% Lab N-17

dat0 = load('../data/oct12-01.txt')'; % note transpose
dat1 = load('../data/oct14-01.txt')'; % note transpose
dat2 = load('../data/oct26-01.txt')'; % note transpose
dat3 = load('../data/nov01-01.txt')'; % note transpose

S1n = [10037064*(423399/423600) 3614150 25081541 12881573]; % scaler 1 counts
Tn =  [423399                   153504  1036965  536912]; % durations

%------------------%-------------------------------------------------------
% Flux calculation %
%------------------%
stdT = 1./Tn; % std dev of Tn
wT = 1 ./ stdT.^2; % weights for durations

P = polyfit(S1n,Tn, 1);
F = 1/P(1) % flux
y = polyval(P,S1n);

DELTA = sum(wt)*((wT*Tn').^2) - (wT*Tn')^2;
dB = sqrt(sum(wT)/DELTA);
dF = F^2 * dB % error of the flux

%-----------------------------%-------------------------------------------
% collate the data into 1 set %
%-----------------------------%

dat = dat0 + dat1 + dat2 + dat3;
X = 1:255; % time bins
Y = zeros(size(X));
for i = 1:255
    Y(i) = dat(i);
end
weights = Y; % w = 1/s^2, s = 1/sqrt(Y)

T = sum(Tn); dT = 4; % because 4 samples
[days, hrs, min, sec] = parsetime(T);
S1 = sum(S1n);

%-------------------%-------------------------------------------------------
% Nonlinear fitting %
%-------------------%

% Uncomment to start the curve fitting tool
% - this would give estimates for Nbg, N0 and lambda0
% - and their standard deviations due to numerical approximations
% cftool(X,Y, weights);

% settings
fitrange = 8:191;

% results
```

```
N0 = 2.94e5;
Nbg = 123.3;
tau0 = 2.197e-6;

% function to display graphs from last fitting session with cftool
curvefitting(X,Y, weights);

%----------------%-------------------------------------------------------------
% Chi-square fit %
%----------------%

O = Y(fitrange); % observed
E = muondecay([Nbg, N0, tau0], X(fitrange)); % expected
c = sum((O - E).^2 ./ E) % test statistic
dof = length(fitrange) - 3 - 1; % degrees of freedom
% chi2inv(0.95, dof); chi-square at 5% level of significance
p = chi2cdf(c, dof) % smaller p => better fit

%---------------------------------------%---------------------------------
% Linear fit for single exponential model %
%---------------------------------------%

% remove background
bgRange = 160:190;
origY = Y; % save
origX = X;
Y = Y-Nbg;
Y(find(Y<0)) = 0;

% limit range (manual, need to adjust for different data)
range = 10:120;

indices = find(Y ~= 0); % use only non-zero counts
Y = Y(indices);
X = X(indices);

lnY = log(Y);
slnY = sqrt(Y)/Y; % err in lnY
% to narrow the range for straight region
tempRange = find(X==range(1)):find(X==range(end));
P = polyfit(X(tempRange), lnY(tempRange), 1);
lny = polyval(P, X);

Xn = X;
X = origX; % restore
Y = origY;

%--------------------------------%----------------------------------------
% To work out the physical error %
%--------------------------------%

% background determination
nB = length(bgRange); % number of bins
NB = sum(Y(bgRange)); % count in bins
dNB = sqrt(NB); % because Poisson distributed

w = NB / (F^2 * nB * T)
```

```
dw = 1/nB * sqrt((NB/(F^2*T^2)*dT)^2 + (dNB/(F^2*T))^2)

% accepted data
lambda_c = 3.76e4; dlambda_c = .04e4; % muon carbon capture rate
r = 1.268; dr = 0.008; % +/- muon ratio

% theory
weights = origY(range); % stdY = 1/sqrt(origY)
DELTA = sum(weights)*(weights*(origX(range)'.^2)) - (weights*origX(range)')^2;
B = P(1) % to map to the same notation
dB = sqrt(sum(weights) / DELTA);

lambda_eff = -B / w % in Hz
dlambda_eff = sqrt((dB/w)^2 + (B/w^2 * dw)^2)
tau_eff = 1/lambda_eff;
dtau_eff = dlambda_eff / lambda_eff^2;
lambda_0 = lambda_eff - lambda_c /(1+r)
dlambda_0 = sqrt(dlambda_eff^2 + (-dlambda_c/(1+r))^2 + (-lambda_c*dr/(1+r)^2)^2)
tau_0 = 1 / lambda_0 % lifetime
dtau_0 = dlambda_0 / lambda_0^2

%----------------------------------%--------------------------------------
% Linear graph of single exponential fit %
%----------------------------------%

% linear log graph
figure;
hold on;
grid on;
plot(Xn,lnY, 'xr');
drawErrorBars(Xn,lnY, slnY, 1, [1,.5,.5]);
plot(Xn,lny, 'b-');
titlestr = ['Decays after ' num2str(days) ' days ' num2str(hrs) 'h '...
           num2str(min) 'm ' num2str(sec) 's (\tau_0=' num2str(tau_0*1e6)...
           ' +/- ' num2str(dtau_0*1e6) '\mus)'];
title(titlestr);
xlabel('Time bins in 0.1\mu s');
ylabel('ln (Counts)');
```

---

curvefitting.m

```
function curvefitting(X,Y,weights)
%CURVEFITTING    Create plot of datasets and fits
%   CURVEFITTING(X,Y,WEIGHTS)
%   Creates a plot, similar to the plot in the main curve fitting
%   window, using the data that you provide as input.  You can
%   apply this function to the same data you used with cftool
%   or with different data.  You may want to edit the function to
%   customize the code and this help message.
%
%   Number of datasets:  1
%   Number of fits:  1

% Data from dataset "Data":
%    X = X:
```

```
%    Y = Y:
%    Weights = weights:
%
% This function was automatically generated on 04-Nov-2005 02:16:00

% Set up figure to receive datasets and fits
f_ = clf;
figure(f_);
legh_ = []; legt_ = {};   % handles and text for legend
xlim_ = [Inf -Inf];       % limits of x axis
ax_ = subplot(2,1,1);
ax2_ = subplot(2,1,2);
set(ax2_,'Box','on');
legrh_ = []; legrt_ = {};
set(ax_,'Box','on');
grid(ax_,'on');
grid(ax2_,'on');
axes(ax_); hold on;

% --- Plot data originally in dataset "Data"
X = X(:);
Y = Y(:);
weights = weights(:);
h_ = line(X,Y,'Parent',ax_,'Color',[0.333333 0 0.666667],...
     'LineStyle','none', 'LineWidth',1,...
     'Marker','.', 'MarkerSize',12);
xlim_(1) = min(xlim_(1),min(X));
xlim_(2) = max(xlim_(2),max(X));
legh_(end+1) = h_;
legt_{end+1} = 'Data';

% Nudge axis limits beyond data limits
if all(isfinite(xlim_))
   xlim_ = xlim_ + [-1 1] * 0.01 * diff(xlim_);
   set(ax_,'XLim',xlim_)
   set(ax2_,'XLim',xlim_)
end

% --- Create fit "Nonlinear Multiple Exponential Fit"

% Apply exclusion rule "valid"
if length(X)~=255
   error('Exclusion rule "valid" is incompatible with X');
end
ex_ = logical(zeros(length(X),1));
ex_([[(1:7) (192:255)]]) = 1;
fo_ = fitoptions('method','NonlinearLeastSquares','Weights',[24 0 0 0 0 190 8823...
      9783 8942 8876 8431 8151 7715 7486 6908 6737 6298 6048 5811 5480 5264 4954...
      4854 4555 4398 4134 4009 3780 3644 3446 3304 3175 3034 2848 2875 2648 2561...
      2393 2234 2115 2108 1987 1973 1819 1771 1671 1656 1562 1509 1435 1308 1300...
      1255 1234 1186 1152 1088 1083 970 933 932 836 868 792 766 690 726 684 656...
      632 589 613 547 517 513 544 492 437 446 439 390 405 379 350 371 379 362...
      318 359 309 306 321 277 269 264 254 258 253 237 241 246 220 222 205 247...
      225 214 191 199 207 195 191 157 181 182 172 181 167 190 175 180 198 178...
      160 183 168 178 145 173 145 171 151 139 164 147 144 142 136 128 140 140...
      146 117 125 127 140 147 138 148 118 133 123 126 136 120 128 155 123 136...
      149 147 130 133 110 136 126 118 132 119 125 104 141 120 119 149 140 129...
```

```
      126 157 141 128 119 118 117 124 115 120 139 130 124 148 121 118 90 33 0 0...
      1 1 0 1 0 1 0 0 2 0 0 3 2 0 1 1 0 2 1 0 0 1 1 0 0 1 0 2 1 0 0 1 1 1 0 1 0...
      1 0 2 1 0 2 2 0 2 0 1 2 1 1 0 4 1 0 0 2 2 ],'Robust','On','Lower',...
      [0 0 0 ],'Upper', [Inf 150 9e-006 ],'MaxFunEvals',327,'MaxIter',335);
st_ = [400000 100 2.2e-006 ];
set(fo_,'Startpoint',st_);
set(fo_,'Weight',weights);
set(fo_,'Exclude',ex_);
ft_ = fittype('Nbg + (N0/(1+1.268)*1.0089e-7)*(1/tau0*1.268*exp(-1/tau0*(...
      1.0089e-7)*n) + (1/tau0 + 37.6e3)*exp(-(1/tau0 + 37.6e3)*(1.0089e-7)*n))',...
      'dependent',{'N_i'},'independent',{'n'},...
      'coefficients',{'N0', 'Nbg', 'tau0'});

% Fit this model using new data
cf_ = fit(X,Y,ft_ ,fo_);

% Or use coefficients from the original fit:
if 0
   cv_ = {294871.8701479, 123.2575626172, 2.197140698272e-006};
   cf_ = cfit(ft_,cv_{:});
end

% Plot this fit
h_ = plot(cf_,'fit',0.95);
legend off;  % turn off legend from plot method call
set(h_(1),'Color',[1 0 0],...
     'LineStyle','-', 'LineWidth',2,...
     'Marker','none', 'MarkerSize',6);
legh_(end+1) = h_(1);
legt_{end+1} = 'Nonlinear Multiple Exponential Fit';
res_ = Y(~ex_) - cf_(X(~ex_));
[x_,i_] = sort(X(~ex_));
axes(ax2_); hold on;
h_ = line(x_,res_(i_),'Parent',ax2_,'Color',[1 0 0],...
     'LineStyle','none', 'LineWidth',1,...
     'Marker','.', 'MarkerSize',6);
axes(ax_); hold on;
legrh_(end+1) = h_;
legrt_{end+1} = 'Nonlinear Multiple Exponential Fit';

% error bars
for i = 1:length(X)
    drawPoissonErrorBars(X,Y, X(i), 1, 1, [.2,.2,1]);
end

hold off;
legend(ax_,legh_, legt_);
legend(ax2_,legrh_, legrt_);
```

---

muondecay.m

```
% Convenience funtion to calculate expected decay count

function Ni = muondecay(params, n)
    bgRange = 170:190;
```

```
    % fact data
    lambda_c = 37.6e3; dlambda_c = 0.4e3; % capture rate in Hz
    r = 1.268; dr = 0.008; % pos/neg muon ratio
    w = 1.0089e-007; % time bin width, from lab02.m

    Nbg = params(1);
    N0 = params(2);
    tau0 = params(3);

    t = n*w;
    Ni = Nbg + (N0/(1+r)*w)*(1/tau0*r*exp(-1/tau0*t) +...
         (1/tau0 + lambda_c)*exp(-(1/tau0 + lambda_c)*t));
end
```

---

drawErrorBars.m

```
% Utility function to draw error bars on data points

function m = drawErrorBars(X, Y, s, d, color)
    % s = std err of Y
    % d = width of bars

    for i = 1:length(X)
        xi = X(i);
        Xi = find(X == xi);
        if size(Xi,1) ~= 0
            Yi = Y(Xi);
            m = mean(Yi);
            lb = m - s;
            ub = m + s;

            hold on;
            line([xi-d, xi+d], [ub, ub], 'Color', color);
            line([xi, xi], [lb, ub], 'Color', color);
            line([xi-d, xi+d], [lb, lb], 'Color', color);

        else
            m = NaN;
            xi = xi;
        end
    end
end
```

Appendix D: Matlab Output

Relevant graphs are shown in the report in Section 4.

From cftool while conducting the nonlinear fitting with the following settings:
Dataset: Y vs. X with weights (=Y)
Robust: on
Algorithm: Trust-Region
Include points: 8 to 191

```
General model:
       f(n) = Nbg + (N0/(1+1.268)*1.0089e-7)*(1/tau0*1.268*exp(-
1/tau0*(1.0089e-7)*n) + (1/tau0 + 37.6e3)*exp(-(1/tau0 + 37.6e3)*(1.0089e-
7)*n))
Coefficients (with 95% confidence bounds):
       N0 =  2.949e+005  (2.943e+005, 2.955e+005)
       Nbg =       123.3  (114.7, 131.8)
       tau0 =  2.197e-006  (2.191e-006, 2.203e-006)
Goodness of fit:
  SSE: 8.92e+007
  R-square: 1
  Adjusted R-square: 1
  RMSE: 702
```

Printout of calculated values:

```
F =

   24.3396


dF =

  4.3904e-016


c =

  186.7549


p =

    0.6505


w =
```

```
  1.0089e-007

dw =

  1.5982e-009

B =

   -0.0479

lambda_eff =

  4.7480e+005

dlambda_eff =

  7.5862e+003

lambda_0 =

  4.5822e+005

dlambda_0 =

  7.5885e+003

tau_0 =

  2.1824e-006

dtau_0 =

  3.6142e-008
```

Appendix E: Determination of Operating Voltage of the PMT

The response of the PMT detection was investigated with respect to voltage, and the relationship plotted in Figure E.1. Starting at low voltages, the count increases somewhat exponentially until about 700, where it changes very minimally with respect to voltage. This is where the voltage is large enough for detection of all scintillator activations. At higher voltages, the circuit starts picking up noise and this increases in an exponential-like manner. The best operating point is at the point of inflexion (or plateau) where there is a balance with counting due to noise and being sensitive enough to excitations in the scintillator. At that point, the count changes least with respect to voltage changes too.

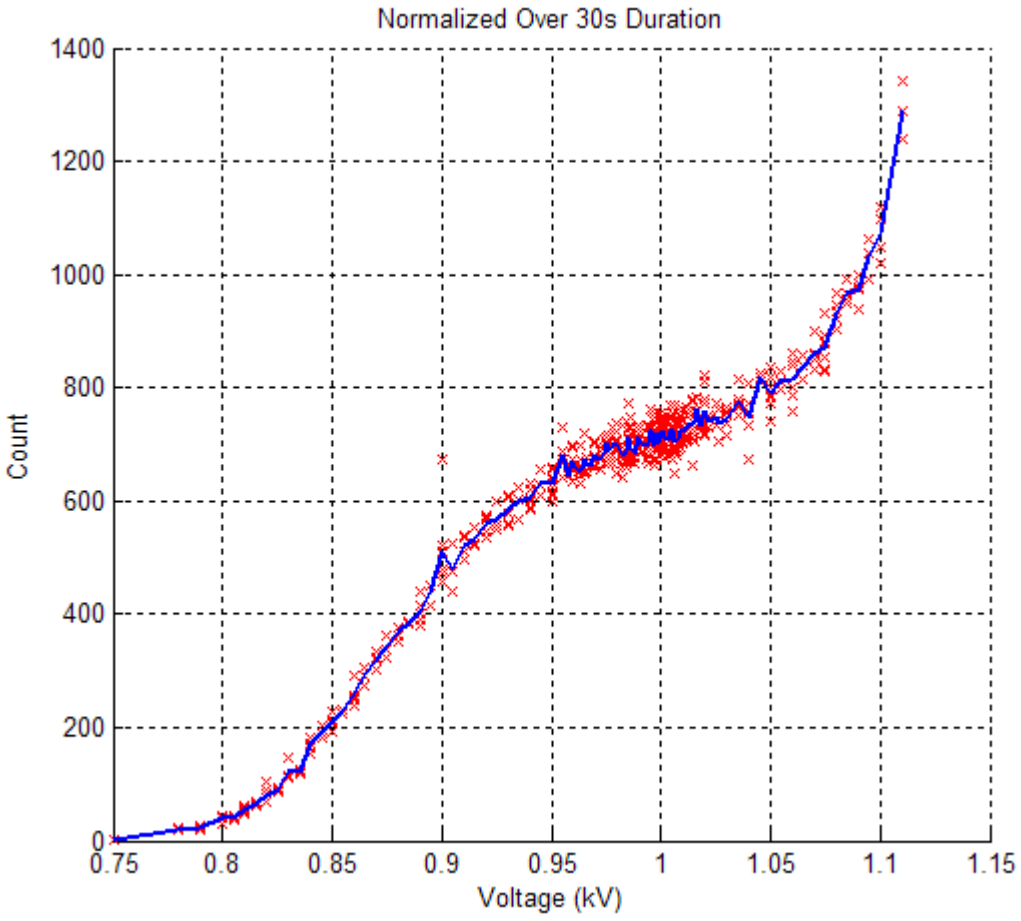

Figure E.1: Determination of operating voltage of the PMT

Error bars are not shown in Figure E.1 so as not to clutter the graph with too many lines and hide the

evident trend. Figure E.2 shows a zoomed in view of Figure E.1, allowing one to find that the best operating point is at 1.005V.

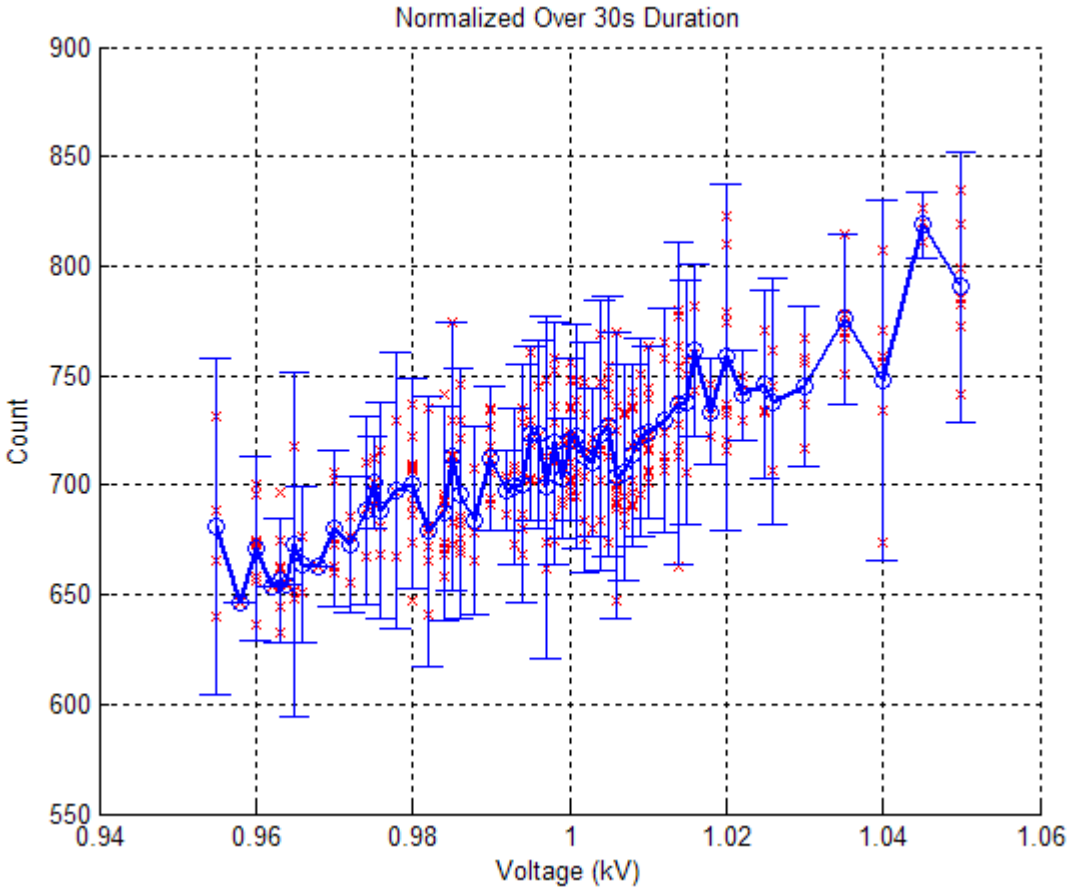

Figure E.2: Zoomed in view of the plateau