

Rotation-vibration transitions in ethyne

Meirin Evans

9214122

School of Physics and Astronomy
The University of Manchester

Second Year Laboratory Report

Mar 2016

Abstract

Through measurements with a spectrometer, the rotation-vibration energy spectra of water and ethyne (acetylene) were obtained. Water was used as calibration and the ethyne spectrum was studied to determine the moments of inertia of its ground and first excited states. The measured ground state moment of inertia was $(2344 \pm 16) \times 10^{-49} \text{ kgm}^{-2}$, whereas $(2401 \pm 17) \times 10^{-49} \text{ kgm}^{-2}$ was obtained for the first excited state. The results' accuracies were limited by the accuracy in determining the spectral peaks' widths.

1. Introduction

Materials such as water and ethyne (acetylene), C_2H_2 , absorb specific radiation wavelengths corresponding to rotational-vibrational energy state transitions. These transitions are quantised and thus are seen as spectral peaks. Measuring the spectroscopic wavenumbers, k , of these peaks allows identification of the transitions. Molecular vibrations can be quantified by the moment of inertia, I , which may be calculated from the transition energies. Knowing the moment of inertia allows calculation of bond length, as the moment of inertia of a rotating mass, m , is given by

$$I = mr^2, \quad (1)$$

where r is the distance of the mass from the rotation centre.

2. Theory

The spectrometer used in this experiment was a differential spectrometer, therefore when channel number is plotted against spectrometer output, minima occur at the start of a transition peak, whilst maxima occur at the end. This is why sharp differential peaks will be seen rather than wider transition peaks. The spectrometer applies a bias voltage of 5V, making 5V the effective 0. The midpoint between maxima and minima in plots such as Figure 1 gives the channel number of the transition peak centre.

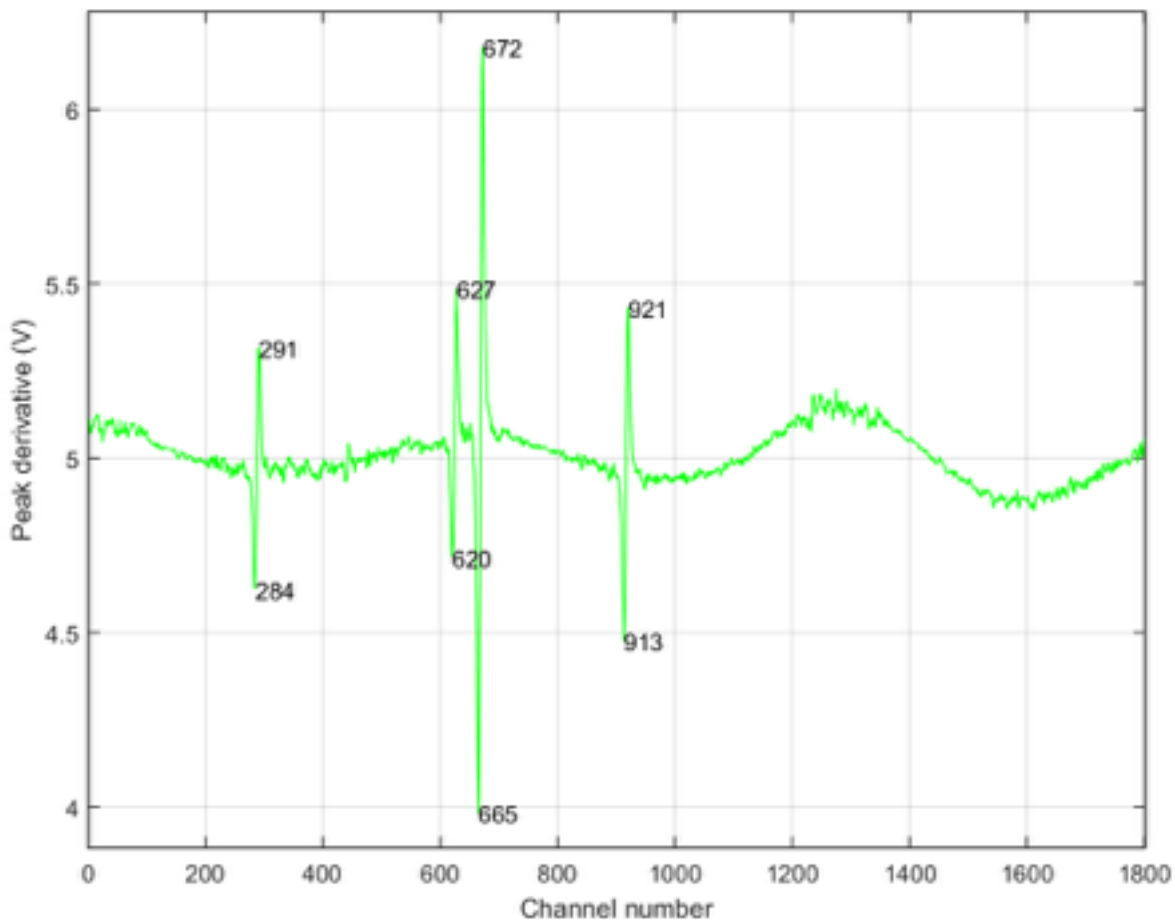


Fig 1. Water spectrum. The channel numbers of the maxima and minima are labelled. Water has 4 transitions peaks and other fluctuation about 5V is noise.

Knowing the relationship between channel number and wavenumber allows conversion to find transition wavenumber. Each transition peak is associated with a quantum number. Knowing the relationship between transition wavenumber and quantum number allows calculation of moment of inertia. If this relationship is linear the moments of inertia of the ground and first excited states of a molecule are the same, I , given by

$$I = \frac{h^2}{4\pi^2 B}, \quad (2)$$

where B is the linear gradient and h is Planck's constant (6.626×10^{-34} Js). If the relationship between wavenumber and quantum number is quadratic the moments of inertia of the ground and first excited states are different, given by

$$I_0 = \frac{h^2}{4\pi^2 (B + C)} \quad (3)$$

and

$$I_1 = \frac{h^2}{4\pi^2 (B - C)}, \quad (4)$$

where I_0 and I_1 are the ground and first excited states moment of inertia respectively and B and C are the linear and quadratic coefficients of the relationship respectively.

For use in Equations 2, 3 and 4 the values of B and C are converted from wavenumber to energy, E , using

$$E = hck, \quad (5)$$

where c is the speed of light (3×10^8 ms⁻¹). Since the masses in ethyne are dominated by carbon, Equation 1 allows estimation of the ethyne C≡C bond length, with a known value of 118 pm [1].

The fact that maxima and minima in plots such as Figure 1 are different means the transition peak has broadened. One cause for this may be Doppler broadening, giving a peak width of

$$\sigma = 2k_0 \sqrt{\frac{k_B T}{Mc^2}}, \quad (6)$$

where k_0 is the wavenumber of the midpoint between maxima and minima, k_B is Boltzmann's constant (1.381×10^{-23} JK⁻¹), T is the temperature at which measurements are made and M is the mass of a molecule. Doppler broadening is a result of the motion of the molecule relative to an observer.

3. Experimental method

Water has known transition wavenumbers therefore was used to calibrate ethyne data. Plotting the midpoint channel number against these wavenumbers gave a linear relationship between these variables, as shown in Figure 2. To test whether the relationship between quantum number and transition wavenumber was quadratic or linear the data was fitted to both and the χ^2 values were compared. This was implemented in a **MATLAB** [2] script.

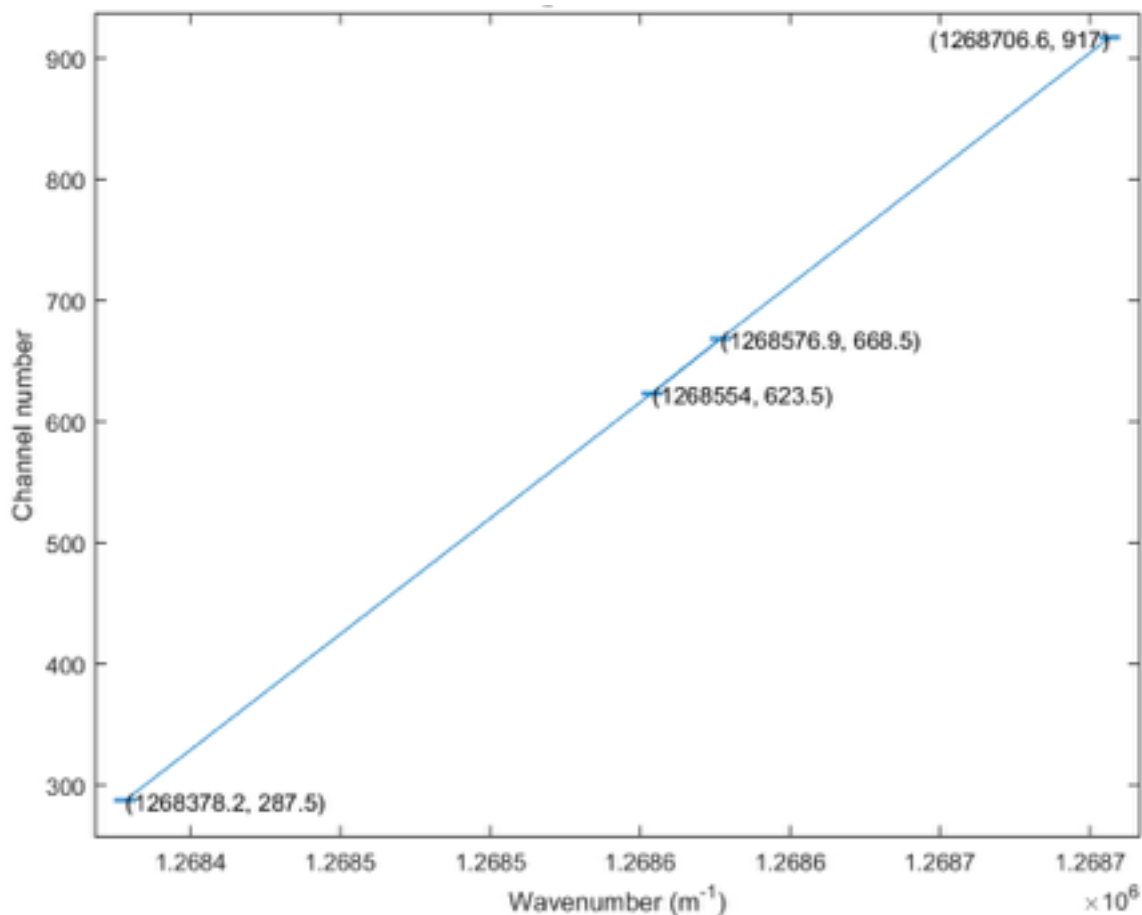


Fig 2. Water calibration. The known wavenumbers for water are plotted against the midpoint of the maxima and minima from Figure 1 to obtain the relationship between channel number and wavenumber. Each point is labelled with wavenumber and channel number.

The wavenumber range used was 939.4 m^{-1} , inverting gave a wavelength range of 0.0011 m . The mid-range wavenumbers and wavelength were 1268698.4 m^{-1} and 788 nm respectively. Using the calibration from Figure 2, the wavenumber step size was 0.52 m^{-1} .

4. Results

The relationship between wavenumber and channel number was found to be linear, as expected, with the residuals shown in Figure 3. Ethyne's spectrum is shown in Figure 4, where five transitions are seen.

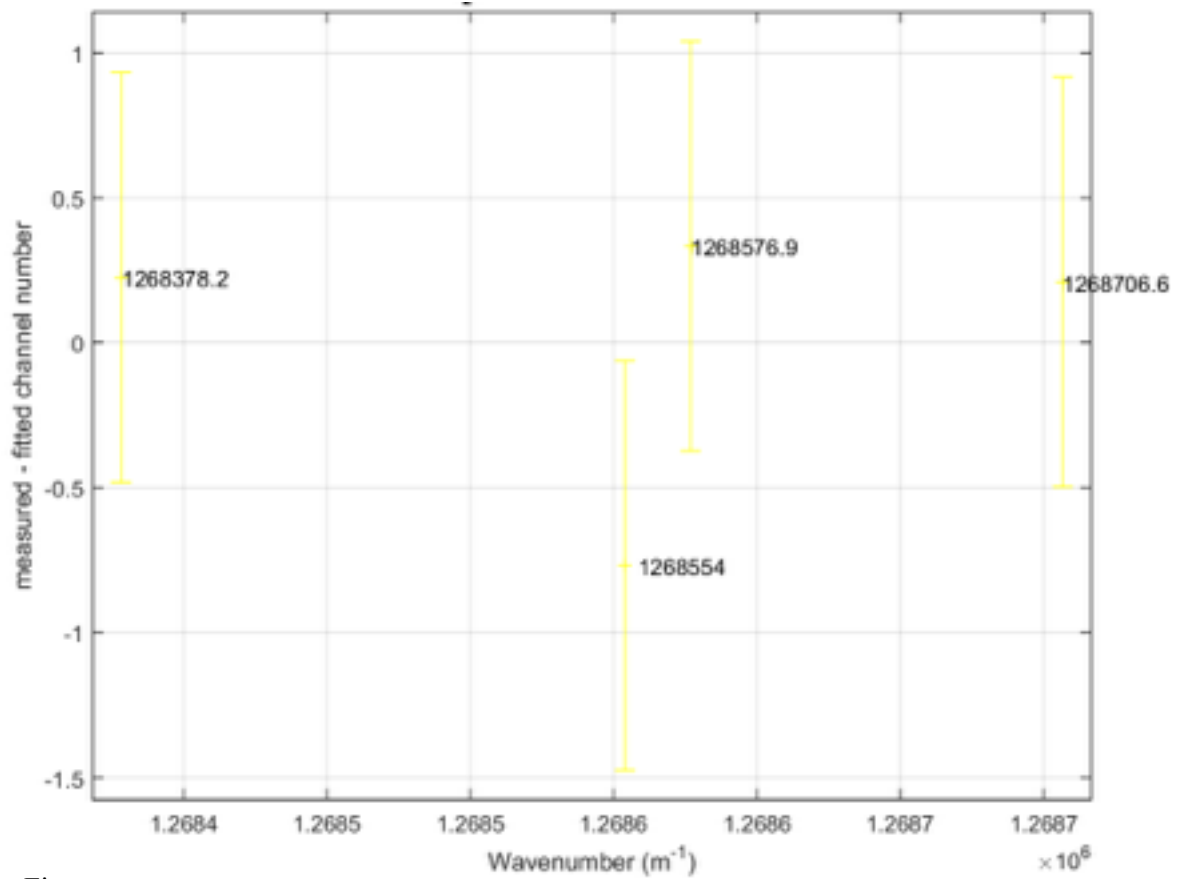


Fig 3. Calibration residuals. The transition wavenumbers are plotted against the difference between measured and fitted channel number. The residuals are sufficiently small (given that the channel number range is 1-1801) to conclude that a linear fit is suitable. The wavenumbers of each transition are labelled.

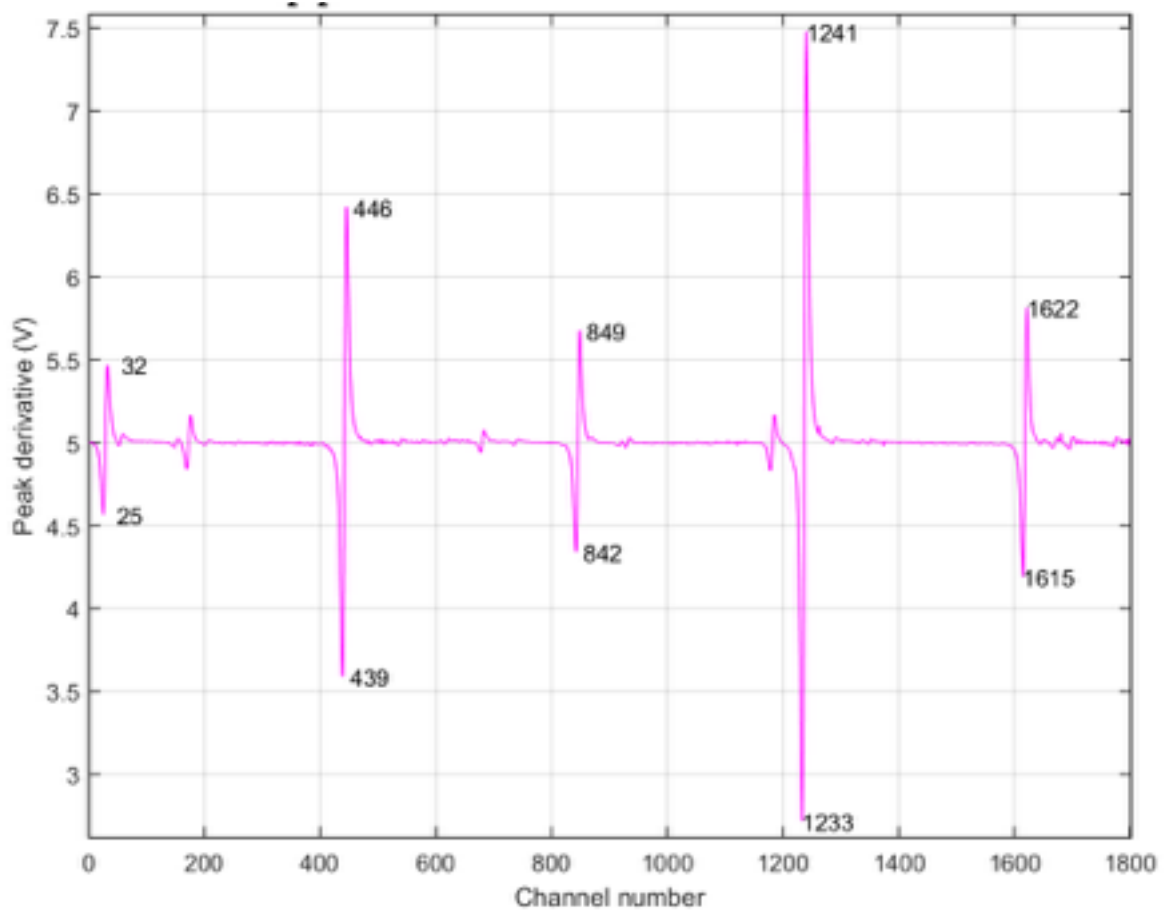


Fig 4. Ethyne spectrum. As Figure 1 but for ethyne rather than water.

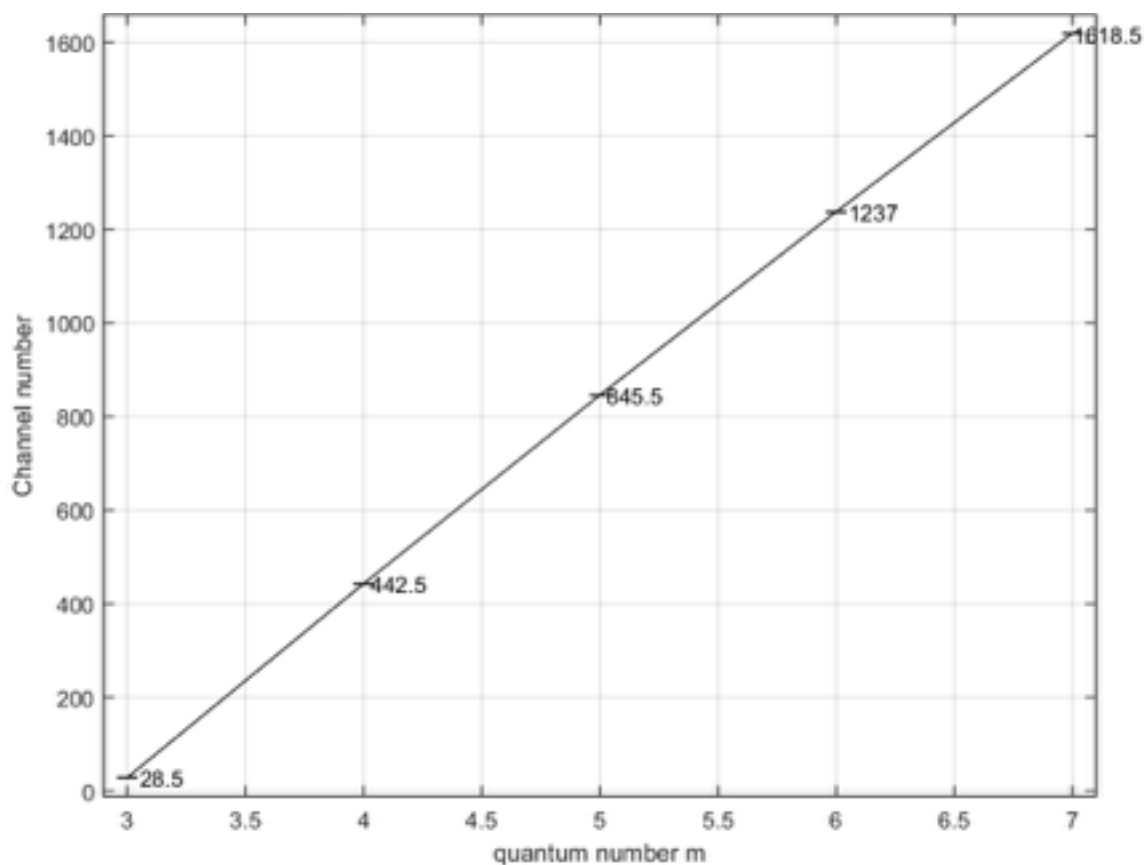


Fig 5. A plot of quantum number against transition channel numbers. The channel numbers for each of ethyne's five transitions are labelled.

Figure 5 does not show whether the relationship between quantum number and transition channel number is linear or quadratic therefore the residuals for both are shown in Figure 6. Since the quadratic gave a better fit than the linear, the quadratic was used. Its B and C values were $(4680 \pm 30) \times 10^{-26}$ J and $(-57 \pm 3) \times 10^{-26}$ J respectively.

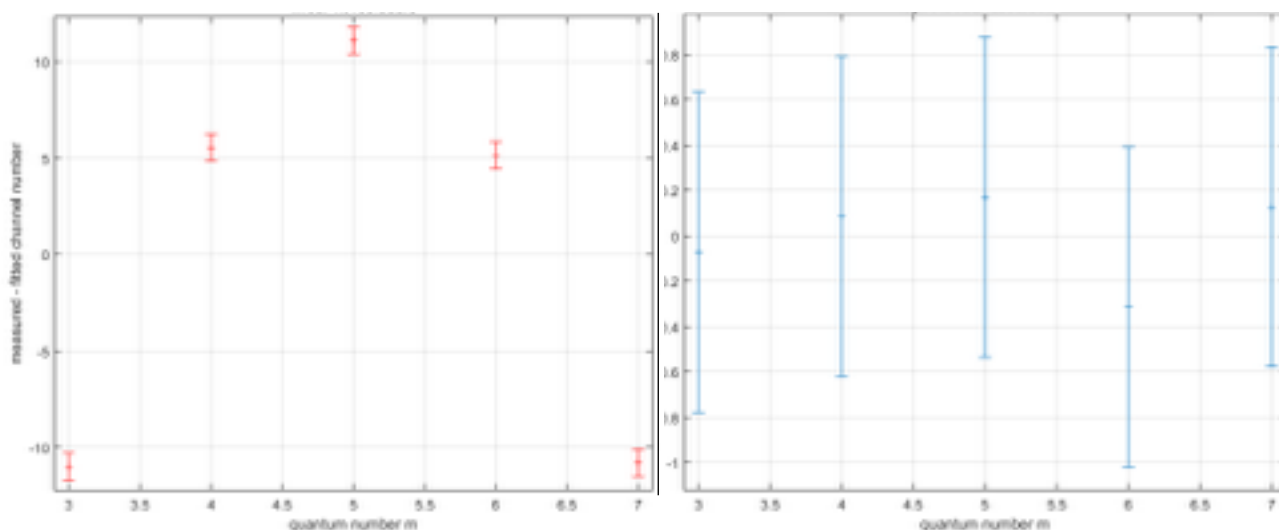


Fig 6. Left are the residuals for a linear relationship between quantum number and transition channel number. Right is the same but quadratic rather than linear. On the x axis is quantum number whilst on the y is the difference between the measured and fitted channel numbers for ethyne's five transitions. The y axes on both plots are different.

I_0 and I_1 were calculated as $(2344 \pm 16) \times 10^{-49} \text{ kgm}^{-2}$ and $(2401 \pm 17) \times 10^{-49} \text{ kgm}^{-2}$ respectively, giving a difference, ΔI , of $(57 \pm 24) \times 10^{-49} \text{ kgm}^{-2}$ through adding errors in quadrature. Equation 1 estimates the ground state bond length, r_0 , as $(153.1 \pm 0.5) \text{ pm}$ and the first excited state bond length, r_1 , as $(154.9 \pm 0.6) \text{ pm}$, giving a stretching, Δr , of $(1.8 \pm 0.8) \text{ pm}$.

5. Discussion

The error assigned to the value of a maximum or minimum was ± 0.5 channel numbers as only integer channel numbers were available. The reduced- χ^2 from Figure 2 was 0.8, confirming the validity of a linear fit between wavenumber and channel number and the suitability of the error value. The reduced- χ^2 of the linear fit was 278.78, whilst the quadratic fit value was 0.105, meaning the quadratic fit was used. The reduced- χ^2 for the quadratic fit was slightly small, suggesting a slight overestimate in errors. However, a higher error value could not be fully justified without extra channel numbers in between those used in this experiment. In calculating errors on the moments of inertia the calibration intercept was not used, as it was insignificant compared to other errors.

Using an approximate room temperature of 300K in Equation 6, σ for water was 3.04 channel numbers. Comparing this with the typical differences between maxima and minima in Figure 1 of seven or eight shows that Doppler broadening cannot be solely responsible for peak width, but was a contributing factor. Other contributors may have been spectrometer instrumental error, such as diffraction.

The estimated bond lengths are different to the true value because the hydrogen atom mass in ethyne were not considered, nor the C-H bond lengths. Taking these into account means the estimated bond lengths were not completely out from the accepted value.

6. Summary

The calculated moments of inertia for the ground and first excited states of ethyne are far enough apart to conclude that they are different. This is seen by comparing the lower limit of the ground state moment of inertia ($2428 \times 10^{-49} \text{ kgm}^{-2}$) to the upper limit of the first excited state moment of inertia ($2418 \times 10^{-49} \text{ kgm}^{-2}$). This justifies the use of a quadratic fit for quantum number against transition wavenumber, as a linear fit would have led to these values being too close together to conclude they were different.

The main results are summarised in Table 1.

Quantity	Value
I_0	$(2344 \pm 16) \times 10^{-49} \text{ kgm}^{-2}$
I_1	$(2401 \pm 17) \times 10^{-49} \text{ kgm}^{-2}$
ΔI	$(57 \pm 24) \times 10^{-49} \text{ kgm}^{-2}$
r_0	$(153.1 \pm 0.5) \text{ pm}$
r_1	$(154.9 \pm 0.6) \text{ pm}$
Δr	$(1.8 \pm 0.8) \text{ pm}$

Table 1. Summary of key results.

References

- [1] Koski, H. & Sandor, E., "Neutron Powder Diffraction Study of the Low-Temperature Phase of Solid Acetylene-d₂", *Acta Crystallog.*, Volume 6, Issue 16, 1975.
- [2] MATLAB, Version 5, The Math Works Inc, Natick, Mass 01760.

Appendix

```
% project1.m
% Data analysis of rotational-vibrational transitions in ethyne (acetylene)
% Meirin Evans, Mar 16
% -----

% -----
% clear previous work
clear all;      % clear variables in workspace
close all;     % close figure windows
% -----

% -----
% read in data files
load h2o.dat;      % load h2o data file
load c2h2.dat;    % load c2h2 data file
% -----

% -----
% locate positions of h2o peaks
h2oinv = -h2o;          % invert h2o
                        % spectrum
[maxh, maxloc] = findpeaks(h2o, 'MinPeakProminence', 0.3); % max, only
                                                              % peaks 0.3 over
                                                              % average
[minh, minloc] = findpeaks(h2oinv, 'MinPeakProminence', 0.3); % min
locs = (maxloc + minloc)/2; % centre
                                                              % point between
                                                              % max & min
% -----

% -----
% plot h2o spectrum
fig1 = figure;          % separate
                        % figure window
u1 = fig1.Name;        % handle
fig1.Name = 'water spectrum'; % rename
                        % figure window
plot(h2o(1:length(h2o)), 'g'); grid % plot in
                                    % green with
                                    % grid
xlabel('Channel number'); % x axis label
ylabel('Peak derivative (V)'); % y axis label
title('H2O spectrum: peak & trough channel number labelled'); % figure title
```

```

xlim([0 length(h2o)+1]); % set x axis
                           limits
ylim([min(h2o)-0.1 max(h2o)+0.1]); % set y axis
                           limits
text(maxloc, maxh, [num2str(maxloc)]); % label max-
                                       ima
text(minloc, -minh, [num2str(minloc)]); % label min-
                                       ima
print('Fig1 water spectrum', '-dpng'); % save figure
                                       as png

% -----

% -----
% least squares fit for h2o
wavenum = [1268378.2; 1268554.0; 1268576.9; 1268706.6]; % wavenumbers for
                                                         h2o peaks
[cal, scal] = polyfit(wavenum, locs, 1); % linear fit for cali-
                                       bration
fig2 = figure; % separate figure
               window
u2 = fig2.Name; % handle
fig2.Name = 'water calibration'; % rename figure
                                   window
errorbar(wavenum, locs, (1/sqrt(2))*ones(size(locs))); % plot with error
                                                         bars in blue
xlabel('Wavenumber (m-1)'); % x axis label
ylabel('Channel number'); % y axis label
title('H2O calibration'); % figure title
xlim([min(wavenum)-10 max(wavenum)+10]); % set x axis limits
ylim([min(locs)-30 max(locs)+20]); % set y axis limits
for n = 1:4 % start of for loop
    if n < 4 % start of if state-
              ment
        text(wavenum(n), locs(n), ['(', num2str(wavenum(n)), ', ', num2str(locs(n)), ')']); % label points
    else % last point
        text(wavenum(n), locs(n), ['(', num2str(wavenum(n)), ', ', num2str(locs(n)), ')'], % label point with
        'HorizontalAlignment', 'right'); % label point with
                                       line to right of text
    end % end if statement
end % end for loop
print('Fig2 water calibration', '-dpng'); % save figure as png
% -----

```

```

% -----
% calibration residuals fit
locsfit = polyval(cal, wavenum); % fitted data
fig3 = figure; % new figure
u3 = fig3.Name; % handle
fig3.Name = 'water calibration residuals'; % name figure

errorbar(wavenum, locs-locsfit, (1/sqrt(2))*ones(size(locs)), 'y+'); grid % plot with
% error bars in
% yellow

xlabel('Wavenumber (m-1); % x axis label
ylabel('measured - fitted channel number'); % y axis label
title('H2O calibration residuals'); % figure title
xlim([min(wavenum)-10 max(wavenum)+10]); % set x axis
% limits

ylim([min(locs-locsfit)-1/sqrt(2)-0.1 max(locs-locsfit)+1/sqrt(2)+0.1]); % set y axis
% limits

text(wavenum, locs-locsfit, [num2str(wavenum)]); % label max-
% ima

print('Fig3 water calibration residuals', '-dpng'); % save figure
% as png

% -----

% -----
% Physics calculations
datarange = length(h2o)/cal(1) - 1/cal(1); % calculate
% wavenumber range

wavrange = 1/datarange; % calculate wave-
% length range

midrangewavenum = (length(h2o) - 2*cal(2) + 1)/(2*cal(1)); % calculate mid
% range wavenumber

midrangewavelen = 1/midrangewavenum; % calculate mid
% range wavelength

step = 2/cal(1) - 1/cal(1); % channel step size

% -----

% -----
% chi squared for h2o
covmcal = sqrt(2)*inv(scal.R)*inv(scal.R)'; % covariance matrix for calibra-
% tion

errcal1 = sqrt(covmcal(1,1)); % error on calibration gradient
errcal2 = sqrt(covmcal(2,2)); % error on calibration intercept
chisqu = 0; % start chi squared sum at 0

```

```

for n = 1:4;                                % loop over 4 values
    d(n) = cal(1)*wavenum(n) + cal(2) - locs(n); % deviation of point from linear fit
                                                line
    doversigmasqu(n) = (d(n)/sqrt(0.5))^2;    % single value of chi squared, er
                                                ror on maxloc is 1
    chisqu = doversigmasqu(n) + chisqu;      % chi squared
end                                           % end for loop
% -----

% -----
% locate positions of c2h2 peaks
c2h2inv = -c2h2;                            % invert c2h2
                                                spectrum
[maxc, maxlocc] = findpeaks(c2h2, 'MinPeakProminence', 0.4); % max, only
                                                peaks 0.4 over
                                                average
[minc, minlocc] = findpeaks(c2h2inv, 'MinPeakProminence', 0.4); % min
loccs = (maxlocc + minlocc)/2;              % centre
                                                point between
                                                max & min
% -----

% -----
% plot c2h2 spectrum
fig4 = figure;                               % separate
                                                figure window
fig4.Name = 'ethyne spectrum';               % rename
                                                figure window
u4 = fig4.Name;                              % handle
plot(c2h2(1:length(c2h2)), 'm'); grid       % plot in ma-
                                                genta with
                                                grid
xlabel('Channel number');                   % x axis label
ylabel('Peak derivative (V)');               % y axis label
title('C_2H_2 spectrum: peak & trough channel number labelled'); % figure title
xlim([0 length(c2h2)+1]);                  % set x axis
                                                limits
ylim([min(c2h2)-0.1 max(c2h2)+0.1]);        % set y axis
                                                limits
text(maxlocc, maxc, [num2str(maxlocc)]);    % label max-
                                                ima
text(minlocc, -minc, [num2str(minlocc)]);    % label min-
                                                ima
print('Fig4 ethyne spectrum', '-dpng');     % save figure
                                                as png
% -----

```

```

% -----
% plot channel number against quantum number
mk = [3; 4; 5; 6; 7];
fig5 = figure;

fig5.Name = 'quantum v channel number';

u5 = fig5.Name;
errorbar(mk, locsc, (1/sqrt(2))*ones(size(mk)), 'k-'); grid

xlabel('quantum number m');
ylabel('Channel number');
title('quantum number m against channel number');
xlim([min(mk)-0.1 max(mk)+0.1]);
ylim([min(locsc)-40 max(locsc)+40]);
text(mk, locsc, [num2str(locsc)]);

print('Fig5 quantum v channel number', '-dpng');
% -----

% -----
% try linear fit
[p, sp] = polyfit(mk, locsc, 1);
fitp = polyval(p, mk);

resid1 = locsc - fitp;

covm1 = sqrt(2)*inv(sp.R)*inv(sp.R)';

errp1 = sqrt(covm1(1,1));

errp2 = sqrt(covm1(2,2));

fig6 = figure;
u6 = fig6.Name;
fig6.Name = 'linear fit residuals';

errorbar(mk, locsc-fitp, (1/sqrt(2))*ones(size(locsc)), 'r+'); grid

xlabel('quantum number m');

```

% quantum numbers
% separate figure window
% rename figure window
% handle
% plot in black with grid
% x axis label
% y axis label
% figure title
% set x axis limits
% set y axis limits
% label wavenumbers
% save figure as png

% linear fit
% column vector for linear data
% residuals for linear fit
% covariance matrix for linear fit
% error on linear gradient
% error on linear intercept
% new figure
% handle
% rename figure
% plot error bars in red with grid
% x axis label

```

ylabel('measured - fitted channel number'); % y axis label
title('linear fit residuals'); % figure title
xlim([min(mk)-0.1 max(mk)+0.1]); % set x axis
limits % limits
ylim([min(locsc-fitp)-1/sqrt(2)-0.5 max(locsc-fitp)+1/sqrt(2)+0.5]); % set y axis
limits % limits
print('Fig6 linear fit residuals', '-dpng'); % save figure
as png

% -----

% -----

% try quadratic fit
[q, sq] = polyfit(mk, locsc, 2); % quadratic
fit % fit
fitq = polyval(q, mk); % column
vector for
quadratic
data
resid2 = locsc - fitq; % residuals
for quadratic
fit
covm2 = sqrt(2)*inv(sq.R)*inv(sq.R)'; % covariance
matrix for
calibration
errq1 = sqrt(covm2(1,1)); % error on
quadratic 2nd
order term
errq2 = sqrt(covm2(2,2)); % error on
quadratic
gradient
errq3 = sqrt(covm2(3,3)); % error on
quadratic in-
tercept
fig7 = figure; % separate
figure window
fig7.Name = 'quadratic fit residuals'; % rename
figure window
u7 = fig7.Name; % handle
errorbar(mk, locsc-fitq, (1/sqrt(2))*ones(size(locsc)), '+'); grid % plot with
error bars in
blue
xlabel('quantum number m'); % x axis label
ylabel('measured - fitted channel number'); % y axis label
title('quadratic fit residuals'); % figure title
xlim([min(mk)-0.1 max(mk)+0.1]); % set x axis
limits % limits

```

```

ylim([min(locsc-fitq)-1/sqrt(2)-0.1 max(locsc-fitq)+1/sqrt(2)+0.1]); % set y axis
                                                                    limits
print('Fig7 quadratic fit residuals', '-dpng'); % save figure
                                                                    as png

% -----

% -----
% chi squared for c2h2
% linear
chisq = 0; % start chi squared sum at
           0
for np = 1:5; % loop over 5 values
    doversigmasq(np) = (resid1(np)*sqrt(2))^2; % single value of chi
                                                squared, error on maxloc is
                                                1
    chisq = doversigmasq(np) + chisq; % linear chi squared
end % end for loop
disp('Residuals for linear fit are in variable resid1'); % display where residuals
                                                         held

% quadratic
chisq = 0; % start chi squared
           sum at 0
for nq = 1:5; % loop over 5 values
    doversigmasquq(nq) = (resid2(nq)*sqrt(2))^2; % single value of chi
                                                  squared, error on
                                                  maxloc is 1
    chisquq = doversigmasquq(nq) + chisquq; % quadratic chi
                                             squared
end % end for loop
disp('Residuals for quadratic fit are in variable resid2'); % display where
                                                            residuals held

disp('Values in resid2 are smaller than those in resid1 therefore quadratic fit is
used'); % tell user which fit
                                                used
disp('resid2 ='); % display resid2
disp(resid2);
% -----

```

```

% -----
% Convert fit coefficients to energy
% C
kC = q(1)/cal(1); % convert channel number to wavenumber

C = kC*6.626e-34*3e+8; % convert wavenumber to energy

sigmaC = (6.626e-34*3e+8/cal(1))*sqrt(errq1^2 + (C*errcal1)^2); % error on C

% B
kB = q(2)/cal(1); % convert channel number to wavenumber

B = kB*6.626e-34*3e+8; % convert wavenumber to energy

sigmaB = (6.626e-34*3e+8/cal(1))*sqrt(errq2^2 + (B*errcal1)^2); % error on B
% -----

% -----
% find moments of inertia
I0 = ((6.626e-34)^2)/((B-C)*4*pi^2); % ground state moment of inertia
I1 = ((6.626e-34)^2)/((B+C)*4*pi^2); % excited state moment of inertia
L0 = 2*sqrt(I0/(2*12*1.667e-27)); % estimate C-C unstretched bond length
L1 = 2*sqrt(I1/(2*12*1.667e-27)); % estimate C-C stretched bond length
compare0 = 120.3e-12 - L0; % compare L0 to known value
compare1 = 120.3e-12 - L1; % compare L1 to known value
% -----

% -----
% estimate errors on moments of inertia
sigmaI0 = ((6.626e-34/(2*pi*(B-C)))^2)*sqrt(sigmaB^2 + sigmaC^2); % error on I0
sigmaI1 = ((6.626e-34/(2*pi*(B+C)))^2)*sqrt(sigmaB^2 + sigmaC^2); % error on I1
% -----

% -----
% calculate difference in moments of inertia
deltaI = I1-I0;
deltaL = L1-L0; % amount of stretching
% -----

```



```

% -----
% consider accuracy of difference in moments of inertia
sigmadeltaI = sqrt(sigmaI0^2 + sigmaI1^2);
% -----

% -----
% print results
fprintf('Ground state moment of inertia I0 = %3.2e +/- %1.0e\n', I0, sigmaI0);
fprintf('1st excited state state moment of inertia I1 = %3.2e +/- %1.0e\n', I1, sigmaI1);
fprintf('Difference in moments of inertia deltaI = %1.0e +/- %1.0e\n', deltaI, sigmadeltaI);
% -----

% -----
% Doppler broadening
sigmaovermean = sqrt(1.381e-23*300/(18*1.661e-27*9e16));
sigma = sigmaovermean.*wavenum;
% -----

```

The number of words in this document is 1481.

This document was last saved on 7/3/2016 at 13:24.