SPECTROSCOPY OF DIATOMIC RARE-EARTH OXIDES: YBO AND EUO

by

Stephen A. McDonald

B.S. University of Virginia

SUBMITTED TO THE DEPARTMENT OF CHEMISTRY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1985

Copyright © 1985 Massachusetts Institute of Technology

Signature of Author Signature redacted

Department of Chemistry
June 3, 1985

Certified by Signature redacted

Robert W. Field
Thesis Supervisor

Accepted by Signature redacted

Dietmar Seyferth
Chairman Departmental Committee on Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 30 1985

Archives

LIBRARIES

MITLibraries

77 Massachusetts Avenue
Cambridge, MA 02139
http://libraries.mit.edu/ask

DISCLAIMER NOTICE

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available.

Thank you.

PAGES CONTAIN BACKGROUND SPECKLING.

Spectroscopy of Diatomic Rare-Earth Oxides: YbO and EuO

by

Stephen A. McDonald

Submitted to the Department of Chemistry on June 3, 1985 in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

High resolution spectra of both YbO and EuO have been recorded in order to test a Ligand Field Theory model for the electronic structure of the rare earth monoxides. The ligand field model distills simplicity from the extremely complex electronic spectra of the rare earth oxides. The states examined in YbO belonged to Hund's case c where the quantum number used to name a state is Ω. Rotational analyses, fluoresence spectra and Ω dependent perturbations were used to assign Ω values and determine the relative energy of five of the expected ten lowest lying electronic states. The ground state of YbO is a $^1\Sigma^+$ state which comes from the $Yb^{2+}(f^{14})\ O^{2-}$ configuration. The other low lying energy levels come from the $Yb^{2+}(f^{13}\sigma)\ O^{2-}$ configuration whose center of gravity lies 8370 cm^{-1} above the ground state. This is the first time that states of both $f^N s$ and f^{N+1} low lying configurations have been observed for the same molecule. In EuO an $^8\Sigma^- - {}^8\Sigma^-$ transition was rotationally analyzed. The $^8\Sigma^-$ ground state of EuO appears to come from the $Eu^{2+}(f^7)\ O^{2-}$ configuration. In addition the data suggest that the upper $^8\Sigma^-$ state may come from the $Eu^+(f^7\sigma)\ O^-(\pi^4\sigma)$ configuration. The data in both YbO and EuO support the ligand field model.

Thesis Supervisor: Robert W. Field
Title: Professor of Chemistry

This doctoral thesis has been examined by a committee of the Department of Chemistry as follows:

Professor James Kinsey — Signature redacted
Chairman

Professor Robert Field — Signature redacted
Thesis Supervisor

Professor Shaoul Ezekiel — Signature redacted

Table of Contents

Abstract 2

Table of Contents 4

List of Figures 10

List of Tables 11

1. Introduction 12

2. Experimental Techniques 17

2.1 Broida Oven 17
2.2 Equipment 18
2.3 Technique 19
2.4 Calibration 19

3. Computerized Data Collection 22

3.1 Introduction 22
3.2 Principles 22
3.3 The Programs 28

4. Angular Momentum in Diatomic Molecules 34

4.1 Introduction 34
4.2 The Rotation Operator 36
4.3 Derivation of Operator for Total Angular Momentum 42
4.4 Parity 54
4.4.1 Electronic and Rotational Parity 54
4.4.2 Parity in Hund's Case a 59
4.4.3 Parity in Hund's Case b 59
4.4.4 Parity in Hund's Case c 59
4.4.5 Parity in a Separated Atom Approach 60
4.5 Conclusion 61

5. Ligand Field Model 66

5.1. Introduction 66
5.2. The Ligand Field Hamiltonian 67
5.3. Predicted Configurations 72
5.4. Qualitative Effect of the Ligand Field 74
5.5. Ligand Field Calculation for YbO 75

6. Europium Oxide 85

6.1 Introduction 85

6.2 Observed Spectra 85
6.2.1 Fluorescence Spectrum 85
6.2.2 High Resolution 86
6.3 $^{8}\Sigma$ Hamiltonian 88
6.3.1 Rotational Hamiltonian 89
6.3.2 Centrifugal Distortion 90
6.3.3 Spin-Spin Hamiltonian 91
6.3.4 Spin-Rotation Hamiltonian 98
6.3.5 Spin-Orbit 100
6.3.6 Hyperfine Hamiltonian 102
6.4 Qualitative Description of Expected Energy Levels 105
6.5 Parity and Naming of the Levels 106
6.5.1 Parity 106
6.5.2 Name 107
6.6 Assignment of the Spectrum 108
6.7 The Fit 114
6.8 Discussion 123
6.8.1 Constants 123
6.8.2 Agreement With Ligand Field Prediction 135
6.9 Conclusion 136

7. Ytterbium Oxide 139

7.1 Introduction 139
7.2 YbO Hamiltonian 141
7.2.1 Rotational Hamiltonian 141
7.2.2 Centrifugal Distortion 144
7.2.3 Hyperfine Hamiltonian 144
7.3 5793, 6032 and 6067 Å Bands 146
7.3.1 Survey 146
7.3.2 High Resolution 149
7.4 5929 A Band 156
7.4.1 High Resolution 156
7.4.2 Fluorescence 157
7.4.3 Summary of 5929 Å Band 165
7.5 5925 A Band 165
7.5.1 High Resolution 165
7.5.2 Fluorescence 168
7.5.3 Summary of 5925 Å Band 174
7.6 6021 A Band 174
7.6.1 High Resolution 174
7.6.2 Fluorescence 175
7.6.3 Summary of 6021 Å Band 179
7.7 4518 A Band 180
7.7.1 Fluorescence 180
7.7.2 Summary of 4518 Å Band 184
7.8 5955 A Band 185
7.9 5952 A Band 185
7.9.1 High Resolution 185
7.9.2 Fluorescence 186
7.9.3 Summary of 5952 Å Band 187
7.10 Discussion 188

7.1. Conclusion 193
8. The Future of Ligand Field Theory **195**
8.1. Conclusions 195
Appendix A. Programs for Data Collection **199**
Appendix B. Hamiltonian for an Octet Sigma State **263**

Acknowledgements

It may seem a bit strange, but one of my primary motivations for finishing this thesis was so that I could write this section. Throughout my life I've come into contact with many people who have taught me a great deal about good scientific research and who, perhaps more important, have been good friends. I'm very happy to have this opportunity to formally thank some of them.

My scientific career began under the supervision of my undergraduate research advisor, Professor Lester Andrews. Lester took me away from the textbooks and put me into a lab. With his help, science became something I did instead of something I studied.

A great deal of the credit for this work goes to my current research advisor, Professor Robert Field. Bob provided superbly equipped labs as well as excellent advice which I occasionally ignored, usually to my detrement. I'm in debt to Bob for all the editing he has done on this thesis. I'll always appreciate the encouragement he gave and the faith he expressed in me throughout my graduate career, even during the most trying periods.

Carter Kittrell taught me the power of a careful, methodical approach to designing and running an experiment. Carter is an unparalleled experimentalist, and I like to think that some of his talent and technique rubbed off on me. Certainly there will always be some of Carter's wisdom in any research I do.

Mike Dulick created Ligand Field Theory, the theoretical framework for

this thesis, and he is still one of the driving forces behind its development. Mike has taught me everything I know about the theory and has given me a great deal of help with the analysis of the EuO spectra. He also did many of the EuO experiments with me, as did Harold Schall.

I learned about using computers in the laboratory from two excellent teachers, Stephen Corteselli and Michael Burns. Stephen was always willing to help me with hardware and software problems of any magnitude, saving me untold hours of frustration. During these fixit sessions, Stephen taught me to read computer circut diagrams and how to wield a soldering iron without doing more damage than good. Mike has been an extremely good starting point for anything I wanted to know about computers. Mike's breadth of knowledge extends from digital filtering to computerized music. He has saved me a tremendous amount of time with his ability to suggest excellent references at the drop of a subject.

John Thomas is the unchallenged angular momentum guru at MIT. My conversations with him always took the same path. "John", I would say, "I think I finally have the right viewpoint for angular momentum. To make sure, I want you to answer the following question yes or no." Then I would repeat the last request slowly and distinctly, "just yes or no this time, o.k.?" He would never, ever answer with yes or no. With a patient sigh John would sit me down for a two hour session on angular momentum theory from the beginning. His message was always the same. Before you make pictures as a shortcut to understanding angular momentum, you must understand the basic principles. After four years, John's words of wisdom finally sunk in. The result is Chapter 4.

By analyzing several transitions in YbO, Professor Colin Linton

demystified the process of reducing transitions to molecular constants. This can be a rather daunting step and without Colin I'd probably still be scratching my head. Colin also directed the early work on YbO, the transitions which involved the ground state, and co-authored a paper about those transitions.

Most of my work on YbO was done with Stephen Rice. Steve has the ability to set his sights on a target and then go after it full speed until he succeeds. This kept us on course through the often perturbed spectra of YbO.

The sometimes unpleasant life of a graduate student has been improved by the friendships I've formed with my fellow sufferers. I'd like to thank Evan Abramson, Harold Schall, and Mike Dulick for always being ready to talk about anything, anytime, and often more intelligently than I could.

Above all, I thank my family for their patience, love and support. My wife, Carol, has been an excellent editor and just all around wonderful.

List of Figures

Figure 6-1:	$^4\Sigma^-$ Energy Levels	109
Figure 6-2:	Strong EuO Lines	111
Figure 7-1:	Excitation Spectrum of the 6032 Å Band	151
Figure 7-2:	YbO 5929 Å Band	162
Figure 7-3:	YbO 5925 Å Band	169
Figure 7-4:	Energy Level Diagram for f^{14} and $f^{13}s$ Configuration in YbO	191

List of Tables

Table 5-I: Calculated Ligand Field Radial Parameters 71
Table 5-II: Predicted Ground States for the Oxides of Ba-Hf 73
Table 5-III: Hamiltonian Matrix Elements for the Yb^{2+} $4f^{13}6s$ Configuration 81
Table 6-I: Diagonal Elements of the Spin-Spin Hamiltonian 97
Table 6-II: Diagonal Elements of the Spin-Rotation Hamiltonian 100
Table 6-III: Names and Parity of the Rotational Levels of an $^{8}\Sigma^{-}$ State 110
Table 6-IV: Hyperfine Splittings of an $^{8}\Sigma$ State 113
Table 6-V: EuO Transition Frequencies 115
Table 6-VI: Combination Differences for EuO Subbands 119
Table 6-VII: EuO Constants 122
Table 6-VIII: EuO Fit to Experimental Data 124
Table 6-IX: Calculated vs. Experimental Centrifugal Distortion Constants 133
Table 7-I: Resolved Fluorescence from the 6032 and 6067 Å Bands 148
Table 7-II: Observed Lines in the 5793, 5032 and 6067 Å Bands 152
Table 7-III: Comparison of Combination Differences for ^{174}YbO Bands with Levels in Common 153
Table 7-IV: 5929 Å Line List 158
Table 7-V: Constants from the 5929 Å Band 161
Table 7-VI: Splitting of Ω-Doubled Lines in the 4968 Å Band 164
Table 7-VII: Bands Observed from the 5929 Å Laser Excitation 165
Table 7-VIII: Direction Cosine Matrix Element for Relative Intensity 167
Table 7-IX: Constants from the 5925 Å Band 170
Table 7-X: 5925 Å Line List 171
Table 7-XI: Comparison of Combination Differences for Lower States of the 5929 and 5925 Å Bands 173
Table 7-XII: Bands Observed from the 5925 Å Laser Excitation 174
Table 7-XIII: Molecular Constants from the 6021 Å Band 175
Table 7-XIV: 6021 Å Line List 176
Table 7-XV: Splitting of Ω-Doubled 4829 Å Band 178
Table 7-XVI: Splitting of Ω-Doubled 5027 Å Band 178
Table 7-XVII: Splitting of Ω-Doubled 5244 Å Band 179
Table 7-XVIII: Bands Observed from the 6021 Å Laser Excitation 180
Table 7-XIX: Bands Observed from the 4518 Å Laser Excitation 185
Table 7-XX: Bands Observed from the 5955 Å Laser Excitation 186
Table 7-XXI: Bands Observed from the 5952 Å Laser Excitation 187
Table 7-XXII: Observed Lower Energy Levels of YbO 189
Table 7-XXIII: YbO Ligand Field Fit 190

Chapter 1
Introduction

Statement of Problem

High resolution spectroscopy on the visible bands of the diatomic rare earth oxides has proved very difficult because of the extreme complexity of the spectra. When an analysis has been performed, the unscrambling of the transitions to produce an energy level diagram, much less to gain physical insight into the molecule, has proved impossible. The difficulties stem from two factors:

1. The molecular energy levels derive from partially filled atomic f orbitals on the metal atom. The high orbital angular momentum and spin multiplicity of the atomic ion-core can lead to a large number of low lying molecular levels, many of which will be thermally populated in an oven. As a result, even the absorption spectra will be very congested.

2. Since the myriad of molecular states formed from the 4f orbitals on the metal have such similar electronic structure, the molecular levels will have very similar constants. Consequently one of the valuable handles a molecular spectroscopists normally uses to distinguish one state from another, its constants, is rendered worthless.

Recently, Dulick [3] developed a model for the diatomic rare earth oxides based on Crystal Field Theory. Dulick's model requires only the energy levels of the parent metal ion and their wavefunctions along with a straightforward electrostatic calculation to predict an energy level diagram and many of the electronic properties of low lying states of the rare earth oxides. For all of

the oxides made up of metals in the Lanthanide series, the model starts with a $M^{2+}O^{2-}$ distribution of the electrons. Since O^{2-} has only filled shells, the relative ordering of the molecular energy levels comes from the 4f and 6s electrons on the M^{2+} parent metal ion. Dulick successfully used his model on the oxides of Cerium and Praseodymium, for which the electron configurations are 4f6s and $4f^2$6s. To examine the applicability of the model to molecules with a large number of 4f electrons, the research described in this thesis was undertaken to check specific predictions of the model for YbO and EuO. The ground states of YbO and EuO are predicted to come from the $4f^{14}$ and $4f^7$ metal ion configurations respectively, with states belonging to the $4f^{13}$6s configuration in YbO lying close to the energy of the ground state.

Prior Investigations

The low dispersion emission spectrum of most rare earth monoxides was systematically catalogued by Gatterer et. al. [5] in 1957. Almost a decade later, the ESR measurements of LaO in rare gas matricies by Weltner, McLeod and Dasai [9] established beyond any doubt that the ground state was $^2\Sigma^+$. Subsequent analyses [6] indicated that the LaO electronic structure was quite similar to that of ScO, YO and the isoelectronic alkaline-earth halide series. The comprehensive spectroscopic study of the rare-earth monoxides in rare gas matrices by DeKock and Weltner [1], followed later by Gabelnick, Reedy and Chasanov [4] established preliminary estimates of ground state vibrational constants for the remaining diatomic rare earth oxides (except YbO). ESR on GdO has identified the ground state as a $^9\Sigma$ [8]. A rotational analysis has been done on NdO [7].

Low resolution chemiluminescence studies have been done on both EuO and YbO [2, 10].

Outline of Thesis

Chapter 2 details the experimental set-up. A computer system was developed to collect and aid in the analysis of the thousands of lines obtained. The programs are explained in Chapter 3. The construction of effective molecular Hamiltonians requires a good understanding of angular momentum in diatomic molecules and Chapter 4 explores the necessary concepts. Chapter 5 describes the Crystal Field Model. Chapter 6 presents the experimental data and analysis for EuO and compares the predictions of the crystal field model to the results while Chapter 7 does the same for YbO.

References

[1] R. L. DeKock and W. Weltner, Jr.
Spectroscopy of Rare Earth Oxide Molecules in Inert Matrices at 4 K.
J. Phys. Chem. 75:514, 1971.

[2] C. R. Dickson and R. N. Zare.
Beam-Gas Chemiluminescent Reactions of Eu and Sm with O_3, N_2O, NO_2 and F_2.
Chem. Phys. 7:361-370, 1975.

[3] M. Dulick.
Spectroscopic Studies of the Diatomic Rare-Earth Oxides.
PhD thesis, M. I. T., 1982.

[4] S. D. Gabelnick, G. T. Reedy and M. G. Chasanov.
Infrared Spectra and Structrue of some Matrix-Isolated Lanthanide and Actinide Oxides.
J. Chem. Phys. 60:1167, 1974.

[5] A. Gatterer, J. Junkes, E. W. Salleter and B. Rosen.
Molecular Spectra of Metallic Oxides.
Vatican Press, Vatican City, 1957.

[6] K. P. Huber and G. Herzberg.
Molecular Spectra and Molecular Structure, IV, Constants of Diatomic Molecules.
Van Nostrand Reinhold Co., New York, 1979.

[7] L. A. Kaledin, E. A. Shenyavskaya and I. Kovacs.
Electronic Spectrum of NdO.
Acta Physica Hungarica 54(1-2):189, 1983.

[8] R. J. Van Zee, R. F. Ferrante, K. J. Zeringue and W. Weltner,Jr.
The Ground State and Zero-Field Splitting of the GdO Molecule.
J. Chem Phys. 75(11):5297, 1981.

[9] W. Weltner, Jr., D. McLeod, Jr. and P. H. Kasai.
ESR and Optical Spectroscopy of ScO, YO and LaO in Neon and Argon Matrices;Establishment of Their Ground Electronic States.
J. Chem. Phys. 46:3172, 1966.

[10] A. Yokozeki and M. Menzinger.
Molecular Beam Chemiluminescence. VIII: Pressure Dependence and Kinetics of $Sm^+(N_2O, O_3, F_2, Cl_2)$ and Yb^+ (O_3, F_2, Cl_2) Reactions. Dissociation Energies of the Diatomic Reaction Products.
Chem. Phys. 14:427-439, 1976.

Chapter 2
Experimental Techniques

2.1 Broida Oven

The monoxides are made in a device called a Broida Oven. Pieces of metal, 99.9% pure, are placed in an Al_2O_3 crucible which is surrounded by a tungsten heating basket. A very small hollow pipe shaped into a ring surrounds the top of the crucible and delivers the carrier gas to the system. The ring is capped by a stainless steel cone that confines the gas/metal mixture flowing from the crucible to a stream about 7 mm in diameter. A stainless steel water-cooled base supports the assembly. The base also supports a cylindrical Al_2O_3 shield which confines the heat to the crucible region. The tip of the cone pokes through the top of the shield where the tip is encircled by a another ring which provides the oxidant gas.

To do an experiment, the following procedure is used. First a clean sample of metal, one to three grams, is placed in the crucible. The system is then closed and pumped down to $\sim 10^{-2}$ torr. The carrier gas, always Ar for the experiments described in this thesis, is adjusted such that the pressure in the oven is from two to five torr. Voltage is applied to the wire basket, heating the metal. The oxidant gas is bled into the system and a flame, a thin stream of excited gas which fluoresces in the visible, is observed. The metal temperature (vapor pressure), carrier gas and oxidant gas pressure are all adjusted to produce the maximum laser excited fluorescence. This fluorescence

is focussed onto the slits of a Spex spectrometer, model 1802, 1200 grooves/mm, grating blazed at 1.2 μ. A Hamamatsu R928 photomultiplier detects the light exiting from the monochromator.

Little is known about the exact details of the flame reactions, but the overall reaction is quite simple. The reaction of N_2O, 70 kcal/mole N_2-O bond dissociation energy, with the metal is a highly exothermic reaction, $\Delta H = 30$ kcal/mole for YbO, $\Delta H = 60$ kcal/mole for EuO. The extra energy can go into electronic excitation of the molecule. For example, levels 4300 cm^{-1} above the ground state of YbO had enough population to produce an excitation signal.

2.2 Equipment

Three continuous wave dye lasers were used, Coherent models 599-21, 699-21 and 699-29. The first is a standing wave laser used for broad band scans (laser $\sim cm^{-1}$ wide) and single cavity mode scans with stilbene dye. For all single mode scans the resolution was limited by the Doppler width of the molecules, $\sim$ 0.03 cm^{-1}. Both Coherent model 699 lasers were used single mode. The 699-21 is a ring laser that was used exclusively with rhodamine 6G dye. Its main advantage over the 599 is the increase in power by a factor of 5 to 10. The 699-29 is a 699-21 with an Apple IIe computer which controls the laser and takes data. The dye lasers were pumped with Ar^+ ion lasers.

For high resolution (single mode) experiments which did not use the 699-29 and its attached Apple computer, scans were recorded with a Digital LSI-11/23 based computer. The computer data collection system is detailed in Chapter 3.

2.3 Technique

All experiments were single laser excitations with the fluorescence monitored at right angles to the beam through a monochromator. Three types of spectra were recorded. Broadband excitation scans were taken with the model 599 laser operating multimode, bandwidth $\sim$1 cm^{-1}, and the spectrometer operating in zero order. Thus the monochromator was being used as a spatial filter only. The laser was chopped at about 1 kHz and the fluorescence was detected with a lock in amplifier, PAR model HR-8, to remove the background chemiluminescence. The bands observed by broadband excitation were then probed in single mode high resolution scans with the spectrometer set to a strong fluorescence line, not necessarily in the same band as excited by the laser. The frequency selection provided by this technique removes most of the overlapping transitions. The resolution was limited by the Doppler width of the molecules, $\sim$0.03 cm^{-1}. After the excitation scans, the laser frequency was held constant while fluorescence scans were taken with the monochromator. These scans were good for relative frequencies to $\sim$1 cm^{-1}. The fluorescence scans told which branch the laser was exciting and gave information about fluorescence to other electronic states.

2.4 Calibration

For the high resolution single mode scans taken in the rhodamine 6G region, the iodine atlas of Gerstenkorn and Luc was used [2]. As suggested in the atlas, the absolute wave number of all transitions have been corrected by subtracting 0.0056 cm^{-1} from the frequencies. The lines should be accurate to

about 0.008 cm^{-1}. The stilbene work was calibrated with a Te_2 atlas [1]. The accuracy should also be about 0.008 cm^{-1}.

References

[1] J. Cariou and P. Luc.
Atlas du Spectre d'Absorption de la Molecule de Tellure.
Technical Report, Editions du C.N.R.S., 1980.
Paris, France.

[2] S. Gerstenkorn and P. Luc.
Atlas du Spectre d'Absorption de la Molecule d'Iode.
Technical Report, Editions du C.N.R.S., 1978.
Paris, France.

Chapter 3
Computerized Data Collection

3.1 Introduction

This chapter is divided into two parts. The first section is a mini-tutorial on the common hardware and software used for real time data collection. A limited knowledge of computers is necessary for this section. In the second part I shall present the programs I have written for data collection. Since the programs are in C and PDP-11 assembler, a knowledge of these languages is necessary.

3.2 Principles

The first requirement for computerized data collection is a clean, computer readable signal. This may take many forms: an analog voltage, a set of TTL compatible voltages or data in the IEEE bus format. It all depends on the input board in the computer. Care must be taken with analog voltages since a computer can often cause radio frequency and ground loop problems. If you suspect these problems, I suggest you read Horowitz [4], Hoenig [3] and Morrison [6].

Suppose you have a good analog voltage and an analog to digital (A/D) board in the computer. How does the program tell the A/D to start a conversion? In most computers the A/D is accessed by a technique called

memory mapping. As far as the computer knows, the A/D looks just like a couple of memory locations. Suppose I have an A/D that is set to respond to memory locations 10 and 11. Futhermore suppose a 1 in memory location 10 tells the A/D to start a conversion. When the A/D is finished, the A/D may change the number in location 10 to a 2 and the digital number corresponding to the input analog voltage will now be at location 11. All the computer has to do is put a 1 in location 10, read 10 until a 2 appears and then read the value from location 11. The A/D is "mapped" into the memory space of the computer. In a computer with memory mapping, all input/output (I/O) is done with this technique including disk reads and writes and terminal handling. The details of how to control a given board are described in the technical literature on the board. The memory locations that a given controller responds to can be set with small switches on the board. You must realize that certain boards have to respond to specific memory locations in order for canned programs to work. For example the operating system assumes that the disk I/O is done with certain locations following a prescribed format. So always check for compatibility of a board before making a purchase and make sure its location in memory agrees with any programs which use the board.

I have actually gone a little ahead of the typical series of events because something has to tell the program that it is time to take a data point. Here is where the "real" in "real time data collection" comes from. Often the program waits until a certain amount of time has passed since it took the last data point. To keep time, the program uses a real time clock board which has several memory locations associated with it. One function of the clock is to run for a specified amount of time after which it stops and raises a flag by putting a number into one of its memory locations. If the program continually

tests that location, it will detect the flag being raised and can restart the clock before taking a data point, thereby ensuring evenly spaced data points. However, suppose that every tenth point you want to do some data analysis that will take longer than the time in between points. After doing the extra analysis, the computer will test the "done" flag and find that the clock has already finished timing the desired period. If the computer ignores this and continues on with the data collection, the points will not be evenly spaced in time. To prevent this, the computer could test the flag while doing its analysis, but this is unsatisfactory since there will always be some time delay between tests. A better answer is to use the interrupt capability of the computer.

Let me explain what an interrupt is. Normally the central processing unit (CPU) of the computer is a classic example of a one-track mind. It works only on one piece of code at a given time and is completely oblivious to anything else. However a device with interrupt capability can reach out, grab the CPU and tell it, "You will immediately drop what you're doing and start executing the program whose starting memory location is stored in memory location X", where X depends on a set of switches on the board of the interrupting device. The interrupt program which the CPU is forced to execute must be supplied by the programmer. After executing the special program, called an interrupt routine, the CPU returns to the original program and continues as if nothing had happened.

Back to the clock problem. The real time clock has interrupt capability so that it can force the CPU to execute an interrupt routine whenever it times out. The interrupt routine will reset the clock and record a data point irrespective of the program being executed at the time of the interrupt. The

points will be evenly spaced in time.

Before leaving this discussion on interrupts, I would like to give you a warning. They can be hazardous to your sanity during debugging. By their very nature, interrupts can occur anytime during a program. Several times I have run into the problem that an interrupt routine will cause the main program to bomb only if the interrupt occurs during a specific part of the main program. The program will run successfully 999 times and then fail on the next run. No only is the problem very hard to track down, but the attempts are usually half-hearted because such strange behavior always leads a programmer to suspect the hardware. Therefore, if you must use interrupts, keep them very short and simple.

Evenly spaced data have now been collected from the A/D board and stored in computer memory. A common problem at this stage is a lack of enough memory space to store all of the data points taken during one run. The simplest solution is to flush the memory buffer to a disk or other mass storage device while you are taking data. As usual there is a catch. You may not want to stop taking data during the time you write onto the disk. However you cannot write new, incoming, data into the memory buffer until you're positive that all of the old data in the buffer has been flushed. Where do the new data points go in the meantime? The answer is a technique called double buffering. At any given time one of two buffers is being filled with new data while the other holds the data being written to the disk. When the input buffer is full, it becomes the output buffer and the output buffer becomes the input buffer. You must make sure that the output buffer has been completely emptied before before switching its identity to the input buffer. If not, an error message should be sent to the user and the program

terminated. The standard solution to this problem is to slow down the data rate or increase the size of the buffers.

At last you have the digitized data safely stored on your disk. The data can be considered to be made up of sine waves of differing frequencies and phase. For example, if it takes about 1 second to scan over a transition, a Fourier analysis would show a lot of amplitude for the sine wave at 1 Hz. Hamming [2] shows how digital techniques used to remove noise can be analyzed by their effect on the frequencies which make up the data. If you're planning to do any kind of smoothing with the data you must read Hamming's book first. You must also consider the effects of aliasing. This occurs when your sampling rate is less than twice the frequency of the highest frequency component in your signal. The classic example of aliasing involves the wheels of a stagecoach in an old Western. If the motion picture camera happens to take a picture at the same rate as the revolution of the wheel, the wheel will appear to be motionless. If the camera frequency is slightly slower than the wheel's revolutions, the wheel will seem to move foward very slowly. As the stagecoach starts in a motion picture, the wheel rotation speeds up and then seems to slow, finally stopping and then going backwards. All these effects are due to the high rate of rotation being folded into lower frequencies because the sampling rate is too low. You may have the same problem with your signal. For example, if you sample at 60 Hz and you have lots of 60 Hz noise, you will seem to have a dc component in your data which is not really there. The solution is to use an external filter on the data before it goes into the computer that will strongly attenuate any signals with frequencies at half the sampling rate or higher.

Before discussing my programs I would like to digress a little and pass

on some of the wisdom that I have acquired in the last five years about programming languages. I'll discuss the three languages I have used, PDP-11 Assembler, C and Fortran.

I think that all programmers who use assembly language go through three distinct phases.

1. First there is the ignorant phase. The programmer doesn't really know much about assembly language except that it is reserved for hackers only. However a function that Fortran cannot handle sends the programmer into the second phase.

2. This phase is characterized by a heady sense of power. Programming in assembler seems to be about as difficult as programming a hand held calculator. The guts of the computer are laid bare. Instead of a giant black box compiler which produces executable code located someplace in computerland, every single action of the computer can be completely understood and controlled. Programs can be written that are more efficient and faster than those created with the slow, ponderous higher level languages. About a year later phase three sets in.

3. I call this the older and wiser phase. After many attempts, the programmer realizes that while he can do anything he wants to the computer, the computer seems to find lots more bugs to throw at him. The final product may be fast and efficient, but the amount of time spent programming outweighs the advantage of final execution speed.

Programming in assembler really will teach you how the computer works, and that can be critical in real time data collection. Futhermore, there are times when speed and complete knowledge of the computer are absolutely necessary. Just as with interrupts, when you must use assembler, nothing else will do. But you should not need it very often. Try to reach phase three quickly where you understand the power but also the pitfalls of assembly language programming.

The language called C is a high level language with the ability to do low level operations such as reading a specific memory location. It encourages structured programming [5], where a giant monster of a program is broken into small understandable pieces. C has two problems. First of all, in most implementations all floating point math is done in double precision. On some machines there is hardware support of single precision math only, so all C math will be done in software. This problem is becoming less important as more computers have hardware to do double precision. The other problem is that math functions such as the trigonometric functions or square roots are not supported in the standard version of C. The encouragement of structured programming along with the ability to access specific memory locations make C an excellent language for data collection programs. However its usually limited math capability makes it less desirable for data analysis.

Finally, the old reliable language, Fortran. There is one very good reason to learn Fortran: it is everywhere. Almost all scientific programmers know it, just about any computer you use will have a Fortran compiler, and there are lots and lots of scientific routines written in Fortran. Other than that, forget it. Fortran programs tend to be convoluted monsters that are rife with GO TO statements. Futhermore, there isn't a standard way to access individual memory locations, a detriment for data collection.

3.3 The Programs

I'll describe the overall structure of the programs which are listed in Appendix A. To learn the language C, read the book by Kernighan and Ritchie [5]. The best place to learn PDP-11 assembly language is the book by

Cooper [1]. In the scanning programs there will be numerous references to subroutines whose code is not included in the program. There are several places to look. The first place is the C compiler manual where many useful routines are defined. These are pulled from the object file CLIB.OBJ during the linking process. I wrote a number of small subroutines, both in C and assembler, that do a number of tasks and have been built into a library called SAMLIB.OBJ. If you are still, lost look at the header files STD.H, GRAPH.LIB, PROG.REQ and ADRESS.LIB. If you don't know what a header file is look at Kernigan and Ritchie's book.

Here is what the scanning program has to do. The laser can be scanned 1 cm^{-1} with an external analog voltage ramp provided by the digital to analog converter (D/A) in the computer. The maximum resolution of the 12 bit D/A is used, giving 4096 points per scan. At each step of the D/A, three channels of data are needed: one for the Fabry-Perot etalon calibration pips, one for the iodine calibration scan and one for the data scan.

The supervisor scanning program is SCNCOM, short for scan command, which has the job of calling the other routines that actually do something. In addition, the variables which pass information among the routines are defined in SCNCOM.

SCNCOM first calls the module HARDSE. This program, along with the routines it calls, determines the hardware setup of the system.

The next routine called by SCNCOM is HEADER, so called because it creates the header section of the scan file. The header section consists of one block, 256 sixteen bit words, of the parameters in binary format followed by any number of blocks with the parameters and comments in ASCII. The

exact number of blocks depends on how much information on the scan you record. The exact format of the first block of the header file is described in the routine QANDA. If the data file is typed out with TYPE command from the operating system RT-11, the first block will appear to be nonsense followed by the text of the ASCII blocks. The data to be collected by other subroutines will be placed after the header section. The very first word in the binary block contains the number of the first data block in the file. Thus if there are two blocks of ASCII text, the first word in the file will be a binary three since the binary header block is block number zero followed by blocks one and two of ASCII information.

Upon entry, HEADER checks to see if values for the scan parameters have been saved from the last run in a file called SETUP.SCN. If they have, then HEADER will read the old values, ask the user which ones should be changed and pass this information to the routine QANDA. If SETUP does not exist, HEADER will tell QANDA to ask for new values for all the parameters.

The parameter values are actually obtained from the user by the program QANDA, short for questions and answers. The parameters can be anything the programmer wants: for example the number of channels, the scan time or the data file name. QANDA queries the user for the value of any parameter marked by HEADER. QANDA is carefully constructed such that it is trivial to add or remove parameters to the file. Thus the header section of the file can be tailored to each type of scan. QANDA then returns to HEADER which checks to make sure the new values are satisfactory to the user. If they are, the binary and ASCII header section of the data file is written to a disk. The file SETUP.SCN is also written and closed (the program's

interaction with SETUP.SCN is terminated) so that the new parameters are available the next time HEADER is called. Control then passes back to SCNCOM.

The next program called by SCNCOM is TAKDAT, short for take data. TAKDAT sets up the two buffers, called b2 and b3, used for double buffering. The interrupt routine and real time clock are also initalized. In addition, a circular buffer called b1 is set up. I call it a circular buffer because the interrupt routine fills b1 from start to finish and then rolls around and fills at the start again. In the meantime another routine, called BUFFIL, is clearing out b1 and using the data to fill b2 or b3. I use the third buffer, b1, because I do not want the interrupt routine to worry about double buffering. Also, an interrupt has a hard time doing disk I/O. All the interrupt routine has to do is fill b1 while BUFFIL worries about double buffering and writing the data to disk. TAKDAT then calls DIGEST.

DIGEST, so named because it digests the data, does all the analysis on the incoming data. At present the only analysis consists of storing the data with double buffering and plotting it on the screen. However with the flexibility provided by the use of the interrupt routine, any analysis routines desired may be added to DIGEST.

ADIR.MAC, short for analog to digital interrupt routine, is an assembly language interrupt routine. ADIR is short and very well documented so I will not describe it any further here.

BUFFIL, which stands for buffer fill, is called by DIGEST and performs two important functions. It takes the new data from the interrupt buffer b1 and fills b2 or b3 with them. When one of these buffers is full BUFFIL will

write its contents to the disk.

DISPLY, short for display, displays the data on a graphics terminal. Right now that terminal must be a Digital VT-55 or VT-105. To use the VT-105, DISPLY should be compiled with the header file GRAPH.LIB containing the VT-105 commands. To use the VT-55 terminal, DISPLY should be compiled with the file GRAPH.LIB containing the VT-55 commands.

PRTTIM prints the time remaining in the scan on the screen while the data are being taken.

That is all I have to say about the scanning programs. Remember that Appendix A contains the listings of these programs.

References

[1] J. W. Cooper.
The Mini-Computer in the Laboratory with Examples Using the PDP-11.
John Wiley and Sons, New York, 1977.

[2] R. W. Hamming.
Digital Filters.
Prentice-Hall, Englewood Cliffs, New Jersey, 1977.
An absolutely superb book for signal processing.

[3] S. A. Hoenig.
How to Build and Use Electronic Devices Without Frustration, Panic, Mountains of Money, or an Engineering Degree.
Little Brown and Company, Boston, 1980.

[4] P. Horowitz and W. Hill.
The Art of Electronics.
Cambridge University Press, Cambridge, 1980.
This is the book to read for an introduction to electronics.

[5] B. W. Kernighan and D. M. Ritchie.
The C Programming Language.
Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[6] R. Morrison.
Grounding and Shielding Techniques in Instrumentation.
John Wiley and Sons, New York, 1977.

Chapter 4

Angular Momentum in Diatomic Molecules

4.1 Introduction

At first blush, the theory of angular momentum in diatomic molecules seems fairly simple and well laid out. After all, Herzberg's outline of Hund's cases a-e [2] supposedly provides everything a diatomic molecule spectroscopist could want to know about angular momentum. In my case, Herzberg's treatment raised more questions than it answered. I will list some of those questions in an attempt to whet your appetite for this chapter.

1. Why do people always talk about *spherical* harmonics in connection with diatomic molecules, since diatomic molecules are hardly a spherically symmetric system?

2. Why are you allowed to know two projections of the total angular momentum, projections into both the molecular and space fixed systems when, according to the rules of quantum mechanics, the projection on only one axis at a time can be known?

3. How does the angular momentum operator magically appear in the rotation operator $D(\omega)$ and what is this operator ever useful for?

4. Why is so much known about angular momenta and their projections in Hund's case a where the individual uncoupled angular momenta and their projections ($|\Lambda S\Sigma>$) are known as well as the total angular momentum and its projections ($|JM\Omega>$)? After all, when two angular momenta are described in a coupled system, detailed information about the initial, uncoupled, projections is lost. For example, if I couple $|\ell m_\ell>$ with $|s m_s>$, the coupled wavefunction is written $|\ell s j m_j>$. Except for $m_j = \pm j$, there are several values of m_ℓ and m_s in each coupled wave

function.

5. Since Hund's case a is called the "uncoupled" basis set why can't simple angular momentum coupling equations be used to write case b in terms of case a?

6. Think of evaluating the elements of $\mathbf{R}^2$ in Hund's case c. I can write $\mathbf{R}^2$ as $(\mathbf{J}-\mathbf{J}_a)^2$. When I write $\mathbf{J}\cdot\mathbf{J}_a$ in terms of ladder operators, I will have $J^+J_a^-$. Because of anomalous angular momentum rules J^+ is a lowering operator. J_a^- is also a lowering operator, so both J^+ and J_a^- will lower the value of Ω. Therefore why is the following sequence incorrect?

$$J^+J_a^-|JJ_a\Omega> = (\text{number})J^+|JJ_a\Omega-1> =$$

$$(\text{another number})|JJ_a\Omega-2>$$

7. Where do the anomalous commutation rules, that always make raising and lowering operations with **J** so confusing, come from?

8. Herzberg's description of Hund's case a [3] contains the following statement:

> Ω and the angular momentum **N** of nuclear rotation form the resultant **J**.

Note that Herzberg's **N**, the angular momentum coming only from nuclear rotation, is now called **R**. This statement sounds as if one should add the projection of **L**, Λ, with the projection of **S**, Σ, to give the projection Ω and then add this *projection* to the rotational angular momentum **R** to give the total angular momentum **J**. How do you add a projection to an angular momentum? What are the appropriate Clebsch-Gordan coefficients?

9. Where do the formulas for the determination of absolute parity come from?

For the answers to these and many other questions, read on.

4.2 The Rotation Operator

The rotation operator is a source of confusion for many reasons. First, the appearance of angular momentum in the operator can be somewhat surprising unless the derivation specifically shows how its inclusion is very natural. Once that barrier is hurdled, a reader has to contend with a confusing number of definitions for the supposedly standard Euler angles. Various authors differ as to the direction of rotation, the axes of rotation, and whether to use passive or active rotations. The conventions used by different authors are summarized in Larsson [11] and Wolf [14]. I am using the definitions in Silver [12]. Even when the conventions for rotation are clearly stated, a subtle trap remains. It is very easy to use the wrong form of the rotation operator by not paying close attention to the coordinate systems. Avoiding this trap requires a very good understanding of exactly what the rotation operator does. I will discuss this last pitfall in some detail.

The first step in convincing someone that the rotation operator is really a rational and useful construct rather than magic is to go through a detailed derivation of the operator starting with the infinitesimal rotation operator. This is done very well in many texts, Silver [12] for example, and I need not copy their work. There are, however, two important points that I wish to emphasize from the derivation

1. The angular momentum in the rotation operator appears very naturally. Basically, the infinitesimal rotation of a function requires the derivatives of the function with respect to the angle about the axis of rotation. When these derivatives are determined they turn out to equal the component of the angular momentum about the axis of rotation. For example, the infinitesimal rotation about the z axis uses

$x(\partial/\partial y) - y(\partial/\partial x) = (i/h)\ l_z$.

2. The rotation about one axis, say the z axis, by the angle θ is denoted $R_z(\theta)$ and performed by the operator $\exp[i\theta l_z]$.

The next important step is to do several rotations in succession, for example to relate a fixed axis system to a rotating axis system. This is actually a very tricky operation. Wolf [14] details the correct procedure which I will follow here.

The rotation of a physical system or function about fixed axes is called an active rotation. The rotation of a coordinate system while the physical system remains fixed is called a passive rotation. In either case a function originally described by f(r) becomes F(r') after the rotation.

$$F(r') = f(r) \tag{4.1}$$

For active rotation, this equation says that the value of the function at a particular r is carried along when the function is rotated to the new location r'. Consider a p orbital. If I mark a point in the p orbital and then rotate the orbital, the marked point of the p orbital keeps its same value since the marker moves along with the function. The coordinates of the marker are r and r' before and after the rotation. The same point in the physical space of the function must have the same value. However, the value of f(r') will not necessarily be equal to f(r). Therefore a new function F(r') is needed to describe the function after rotation. For passive rotation r and r' correspond to the same point in space described in different coordinate systems. At the same point in space the function must have the same value irrespective of the coordinate system. In general F() and f() will not have the same functional

form since they must give the same value for different coordinates. The next step is to find a rotation operator **R** such that equation (4.1) is true.

$$F(r') = \mathbf{R}f(r') = f(r). \tag{4.2}$$

This can be interpreted in two ways. The equation

$$F(r') = \mathbf{R}f(r') \tag{4.3}$$

says that the operator **R** transforms the function f() operating on the final coordinate system r' into the final function F(). It is extremely important to note that the coordinate system is the same on both sides of the equality sign and that the operator **R** will have to be given in terms of the final coordinate system. The other interpretation of equation (4.2) is summarized by

$$\mathbf{R}f(r') = f(r). \tag{4.4}$$

This equation states that **R** leaves the function unchanged but changes the coordinate system.

Which interpretation you use, that of equation (4.3) or (4.4), depends on your needs. If you want the final function in the final coordinates, use equation (4.3). If you want to express the initial function in terms of functions in the rotated system, for example $J_p = \Sigma_q J_q D^*_{pq'}$ then use equation (4.4).

Next, successive Eulerian rotations will be considered. I will use equation (4.3) to find the rotated function. For this derivation, f'() refers to a new function and not to the derivative of f().

1. First rotate about the z axis by α to obtain the new frame x'y'z'.

$$f'(r') = \exp[i\alpha L_{z'}]f(r')$$

Note that the final coordinates are inserted into the original function before the rotation and that the final coordinates are used for the rotation operator. In other words, the rotated coordinates, r', are placed in the orginal function, f(), and the rotation operator is specified with the variables of the rotated system.

2. Next rotate about the y' axis by β to arrive at the frame x''y''z''.

$$f''(r'') = \exp[i\beta L_{y''}]f'(r'')$$

$$= \exp[i\beta L_{y''}]\exp[i\alpha L_{z''}]f(r'')$$

3. Finally rotate about the z'' axis by γ to obtain the final frame x'''y'''z'''.

$$f'''(r''') = \exp[i\gamma L_{z'''}]f''(r''')$$

$$= \exp[i\gamma L_{z'''}]\exp[i\beta L_{y'''}]\exp[i\alpha L_{z'''}]f(r''')$$

Thus the rotation operator can be written

$$D(\alpha\beta\gamma) = \exp[i\gamma L_{z'''}]\exp[i\beta L_{y'''}]\exp[i\alpha L_{z'''}]. \tag{4.5}$$

where the triple primes specify the final, rotated coordinate system. In general the primes are not explicitly written. Then we have

$$D(\alpha\beta\gamma) = \exp[i\gamma L_z]\exp[i\beta L_y]\exp[i\alpha L_z]. \tag{4.6}$$

Always remember that axes in this rotation operator are the final, rotated axes.

Now consider the following ***active*** rotation.

1. Rotate about the z axis by $-\alpha$
2. Rotate about the y axis by $-\beta$
3. Rotate about the z axis by $-\gamma$

This set of active rotations has the same effect as the passive rotations just discussed. The relation between the function and the final coordinate system is the same for both series of rotations. Therefore, the same operator can be used for both types of rotation.

To further confuse matters, the passive rotation described by equation (4.5) will do the exact same thing as the following set of ***passive*** rotations [12]:

1. Rotate about the fixed z axis by $+\gamma$.
2. Rotate about the fixed y axis by $+\beta$.
3. Rotate about the fixed z axis by $+\alpha$.

In this case the operator will be

$$D(\omega) = \exp[i\alpha L_z]\exp[i\beta L_y]\exp[i\gamma L_z] \tag{4.7}$$

where the rotation operator is written in terms of the original coordinate system as opposed to equation (4.6) where the operator is written in terms of the final coordinate system. The operator in equation (4.7) is not particularly useful since we usually want the rotation operator to be expressed in terms of the final coordinate system. However, you must be careful because some authors incorrectly use the rotation operator given in equation (4.7) to do the series of Eulerian rotations discussed above, as opposed to the correct form given in equation (4.6) and more explicitely in equation (4.5). For example, Judd [5] performs the following set of passive rotations:

1. Rotate about the fixed z axis by $+\gamma$.
2. Rotate about the fixed y axis by $+\beta$.
3. Rotate about the fixed z axis by $+\alpha$.

He then gives the rotation operator as

$$D(\omega) = \exp[-i\alpha J_z]\exp[-i\beta J_y]\exp[-i\gamma J_z].$$

This operator is incorrect both because the angles have the wrong sign and because Judd wants to use this operator to perform transformations described by equation (4.3). The correct form of the operator, the one which corresponds to the Judd's description of the rotations, is given in equation (4.6). Obvioulsy it is absolutely critical to use the correct form of the rotation operator.

4.3 Derivation of Operator for Total Angular Momentum

It may appear to be overkill to derive the specific form of the operator for angular momentum in a diatomic molecule. After all, if as the rules for the result of the operators on their eigenfunctions are known, the specific forms of the operators are not needed. However, I found working through the derivation to be very helpful in understanding some of the finer points of angular momenta in diatomic molecules. The derivation will follow the procedure in Judd [7].

In a diatomic molecule with two nuclei, labeled A and B, and n electrons the total orbital angular momentum can be written as

$$(i/h)\mathbf{N} = \mathbf{r}_A \times \nabla_A + \mathbf{r}_B \times \nabla_B + \Sigma_{t=1}^{n} \mathbf{r}_t \times \nabla_t. \tag{4.8}$$

where $\mathbf{r}_A$ is the position of nucleus A in the laboratory. ∇_A is shorthand for the vector $(\partial/\partial x, \partial/\partial y, \partial/\partial z)$. Now let

$$\mathbf{r}_C = (M_A\mathbf{r}_A + M_B\mathbf{r}_B)/(M_A+M_B)$$

and

$$\mathbf{r}_O = (M_A\mathbf{r}_A + M_B\mathbf{r}_B + m\Sigma_{t=1}^{n} \mathbf{r}_t)/(M_A+M_B+nm)$$

where C is the center of mass of the nuclear frame excluding the electrons and O is the center for the entire molecule. M_A is the mass of nucleus A and m is the mass of each electron. I will need the additional definitions

$$\mathbf{r}_{BA} = \mathbf{r}_B - \mathbf{r}_A$$

$$\mathbf{r}_{tC} = \mathbf{r}_t - \mathbf{r}_C.$$

At this point I want to change variables such that all electronic motion is referred to the center of mass of the nuclear frame and the motion of the molecule as a whole is removed. In shorthand, the variable change will be

$$(\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_1 \ldots\ldots \mathbf{r}_n) \rightarrow (\mathbf{r}_O, \mathbf{r}_{BA}, \mathbf{r}_{1C}, \mathbf{r}_{2C} \ldots\ldots \mathbf{r}_{nC}).$$

By using the above definitions I get

$$\nabla_A = [M_A/(M_A+M_B+nm)]\ \nabla_O - [M_A/(M_A+M_B)]\ \Sigma_t\ \nabla_{tC} - \nabla_{BA}$$

$$\nabla_B = [M_B/(M_A+M_B+nm)]\ \nabla_O - [M_B/(M_A+M_B)]\ \Sigma_t\ \nabla_{tC} + \nabla_{BA}$$

$$\nabla_t = [m/(M_A+M_B+nm)]\ \nabla_O + \nabla_{tC}.$$

These equations are derived by using the chain rule. For example,

$$\nabla_A = \partial/\partial\mathbf{r}_A = (\partial\mathbf{r}_O/\partial\mathbf{r}_A)(\partial/\partial\mathbf{r}_O) + (\partial\mathbf{r}_{tC}/\partial\mathbf{r}_A)(\partial/\partial\mathbf{r}_{tC}) + (\partial\mathbf{r}_{BA}/\partial\mathbf{r}_A)(\partial/\partial\mathbf{r}_{BA}).$$

Substituting the expressions for ∇_A, ∇_B and ∇_t into equation (4.8) gives

$$(i/h)\mathbf{N} = \mathbf{r}_O \times \nabla_O + \mathbf{r}_{BA} \times \nabla_{BA} + \Sigma_t\ \mathbf{r}_{tC} \times \nabla_{tC}. \qquad (4.9)$$

The terms on the right refer to the angular momentum of the molecule as a whole, the relative nuclear motion and the angular momentum of the electrons referred to the center of mass of the nuclear frame. If translational motion is not of interest, the first term may be ignored. Therefore, the orbital angular

momentum is given by

$$(i/h)\mathbf{N} = \mathbf{r}_{BA}\times\nabla_{BA} + \Sigma_t \mathbf{r}_{tC}\times\nabla_{tC}. \qquad (4.10)$$

The velocity of the electrons is usually much greater than that of the nuclei. Hence it makes more sense to describe the electron motions in a coordinate system which moves with the nuclei. The lab fixed coordinate system has the coordinates (x,y,z) while the coordinates for the molecule fixed system are (ξ,η,ς). The origin for both systems is at the center of mass of the nuclei, C. The ς axis is aligned with the inter-nuclear axis AB and runs from A to B. You might think that the specification of the molecular system relative to the space fixed system would require all three Euler angles. However, the linear molecule is a special case. Once the ς axis is aligned with the internuclear axis, which requires only two Euler angles, there is no way to determine the third angle. To put it another way, I am not aligning two coordinate systems but only two vectors. This requires two rotations about arbitrary coordinate axes, for example the two angles of a polar coordinate system. The change of variables can be represented as

$$(\mathbf{r}_{1C}, \mathbf{r}_{2C}, \ldots\mathbf{r}_{nC}, \theta_{BA}, \phi_{BA}) \rightarrow (\rho_1, \rho_2, \ldots\rho_n, \theta, \phi)$$

where θ_{BA} and ϕ_{BA} are the polar coordinate angles of the vector $\mathbf{r}_{BA}$. Before the change of variables, θ_{BA} and ϕ_{BA} refer to the polar coordinate angles of the internuclear axis. After the change, the electron motion has been locked to that axis and the angles describe a new coordinate system. Therefore, the subscript limiting the angles to the nuclei no longer applies, especially since the angles take on a new meaning. For example, if I examine $\partial/\partial\theta_{BA}$ before the rotation, the derivative operates only on those functions which contain the

coordinates of the nuclei. *After the rotation $\partial/\partial\theta$ also operates on functions which contain the coordinates of the electrons because the positions of the electrons are now a function of the angle θ, among other things.*

I want to write equation (4.9) in terms of the new variables. $\mathbf{r}_{tC}\times\nabla_{tC}$, which equals $(i/h)l_{tC}$, is fairly easy to transform. Both $\mathbf{r}_{tC}$ and ∇_{tC} are vectors with respect to N, and thus their cross product will also be a vector. The rotation operator can be used to transform a vector defined in a fixed coordinate system to one defined in a rotating system. Therefore

$$l_p = \Sigma_q \; l_q D^{1}_{pq}{}^{*}(\omega). \tag{4.11}$$

where l_q is the component of l in the molecule while l_p is the component in space.

The total electron orbital angular momentum, the sum of the individual electron angular momenta, is given by

$$\mathbf{L} = \Sigma_t \; l_t$$

so I can write

$$L_p = \Sigma_q \; L_q D^{1}_{pq}{}^{*}(\omega). \tag{4.12}$$

Remember, the purpose of this exercise is to find an expression for $\Sigma_t \; \mathbf{r}_{tC}\times\nabla_{tC}$ in terms of operators in the rotating frame. Since

$$\Sigma_t \; \mathbf{r}_{tC}\times\nabla_{tC} = \; (i/h)\mathbf{L}, \tag{4.13}$$

I can write

$$(\Sigma_t \mathbf{r}_{tC}\times\nabla_{tC})_p = \Sigma_q L_q D^{1}_{pq}{}^{*}(\omega).$$

The term $\mathbf{r}_{BA}\times\nabla_{BA}$ is more difficult. The vector $\mathbf{r}_{BA}$ is not transformed into the new coordinate system, it *defines* the new system. If I used the same trick that I used for the electrons, the orbital angular momentum of the nuclei would appear to be zero since they are motionless in a coordinate system which rotates with the nuclei. Obviously this is wrong. The reason why this error occurs can be seen if I write

$$\mathbf{r}_{BA}\times\nabla_{BA} = i\mathbf{l}_{BA}$$

and then use the fact that in polar coordinates [5]

$$l_z = -i(\partial/\partial\phi)$$

$$l_{\pm} = \exp[\pm i\phi](\pm(\partial/\partial\theta)+i\cot\theta(\partial/\partial\phi)). \qquad (4.14)$$

Remember that I have already shown that $\partial/\partial\theta_{BA}$ and $\partial/\partial\phi_{BA}$ are different from $\partial/\partial\theta$ and $\partial/\partial\phi$. If I can find expressions for the derivatives $(\partial/\partial\theta_{BA})$ and $(\partial/\partial\phi_{BA})$ in terms of $(\partial/\partial\theta)$ and $(\partial/\partial\phi)$, I will be able to transform l_{BA} into the rotating system. Judd [7] outlines a method which involves evaluating the partial derivatives, and suggests an alternative method using the rotation operator. However, the alternative method is impossible using his form of the rotation operator which, as I have already discussed, is incorrect. While the partial derivative method works for orbital angular momentum, the rotation operator must be used once spin has been included. There are no specific operators for spin so there are not any partial derivatives to take. I will show how to perform the change of coordinates using the rotation operator with the

rotation op< ator given by

$$D(\omega) = \text{xp}[i\gamma L_\zeta]\exp[i\theta L_\eta]\exp[i\phi L_\zeta]$$

where $\mathbf{L} = \Sigma_t\, l_t$ is the sum of orbital angular momentum operators for one electron. I fore I start, it is important to understand that in the same way as for funct ons, the operator being rotated from the original coordinate system is expresse< in terms of the position vectors in the final coordinate system. Thus I will perate on $\partial/\partial\theta$ instead of $\partial/\partial\theta_{BA}$.

$$D(\omega)(\partial/\partial)D^{-1}(\omega) =$$

$$\text{p}[i\gamma L_\zeta]\exp[i\theta L_\eta]\exp[i\phi L_\zeta](\partial/\partial\theta)\exp[-i\phi L_\zeta]\exp[-i\theta L_\eta]\exp[-i\gamma L_\zeta]$$

Since $\partial/\partial\theta$ oes not operate on $\exp[-i\phi L_\zeta]$, I can write

$$D(\omega) = \text{xp}[i\gamma L_\zeta]\exp[i\theta L_\eta](\partial/\partial\theta)\exp[-i\theta L_\eta]\exp[-i\gamma L_\zeta]. \qquad (4.15)$$

I need to d termine the effect of $\partial/\partial\theta$ on $\exp[-i\theta L_\eta]$. Using the commutator,

$$[(\partial/\partial\theta),\text{e p}[-i\theta L_\eta]] = (\partial/\partial\theta)\exp[-i\theta L_\eta] - \exp[-i\theta L_\eta](\partial/\partial\theta)$$

$$= (-iL_\eta)\exp[-i\theta L_\eta] + \exp[-i\theta L_\eta](\partial/\partial\theta) - \exp[-i\theta L_\eta](\partial/\partial\theta).$$

The last st p may be a little confusing if you forget that I am working with operators a l not just functions. The result is

$$[(\partial/\partial\theta),\text{e p}[-i\theta L_\eta]] = (-iL_\eta)\exp[-i\theta L_\eta].$$

so

$$(\partial/\partial\theta)\exp[-i\theta L_\eta] = \exp[-i\theta L_\eta][(\partial/\partial\theta) - iL_\eta].$$

Putting this into equation (4.15) gives

$$D(\omega)(\partial/\partial\theta)D^{-1}(\omega) = \exp[i\gamma L_\varsigma](-iL_\eta)\exp[-i\gamma L_\varsigma] + (\partial/\partial\theta). \tag{4.16}$$

Now I am stuck with evaluating the operator $\exp[i\gamma L_\varsigma](-iL_\eta)\exp[-i\gamma L_\varsigma]$ or at least putting it into a more tractable form. The effect of this operator is to multiply something by a function of gamma, $f(\gamma)$ where

$$f(\gamma) = \exp[i\gamma L_\varsigma](-iL_\eta)\exp[-i\gamma L_\varsigma].$$

I would like to replace this function of gamma by another equivalent function that is easier to use. The key is to determine just what I mean by equivalent. I can replace $f(\gamma)$ by another function if that new function, used in conjunction with the angular momentum operators, produces the same number as $f(\gamma)$. I need to find a new function of γ that behaves the same way as $f(\gamma)$ with respect to differentiation. As far as the variable γ is concerned, $f(\gamma)$ is just the product of two functions of γ, $\exp[i\gamma L_\varsigma]$ and $\exp[-i\gamma L_\varsigma]$. To take its derivative I use the rule for the derivative of a product. If

$$f(\gamma) = \exp[i\gamma L_\varsigma](-iL_\eta)\exp[-i\gamma L_\varsigma] \tag{4.17}$$

then

$$f'(\gamma) = \partial/\partial\gamma\ f(\gamma)$$

$$= \partial/\partial\gamma\ \exp[i\gamma L_\varsigma](-iL_\eta)\exp[-i\gamma L_\varsigma]$$

$$= (\partial/\partial\gamma)[\exp[i\gamma L_\zeta]](-iL_\eta)\exp[-i\gamma L_\zeta] +$$

$$\exp[i\gamma L_\zeta](-iL_\eta)(\partial/\partial\gamma)[\exp[-i\gamma L_\zeta]]$$

$$= \exp[i\gamma L_\zeta](L_\zeta L_\eta - L_\eta L_\zeta)\exp[-i\gamma L_\zeta].$$

With the commutation rules for angular momenta this becomes

$$f'(\gamma) = \exp[i\gamma L_\zeta](-iL_\xi)i\exp[-i\gamma L_\zeta].$$

Following the same series of steps for the second derivative

$$f''(\gamma) = \exp[i\gamma L_\zeta](iL_\eta)\exp[-i\gamma L_\zeta]. \tag{4.18}$$

A comparison of equations (4.17) and (4.18) shows that I can set up the differential equation

$$f(\gamma) + f''(\gamma) = 0.$$

A possible solution is

$$f(\gamma) = A\sin\gamma + B\cos\gamma.$$

Solving for the initial values,

$$f(0) = B = -iL_\eta$$

$$f'(0) = A = -iL_\xi$$

the new, equivalent function is

$$g(\gamma) = -iL_\xi \sin\gamma - iL_\eta \cos\gamma.$$

Putting $g(\gamma)$ into equation (4.16)

$$D(\omega)(\partial/\partial\theta)D^{-1}(\omega) = (\partial/\partial\theta) - iL_\xi \sin\gamma - iL_\eta \cos\gamma. \tag{4.19}$$

The same procedure for $\partial/\partial\phi_{BA}$ gives

$$\partial/\partial\phi_{BA} = \partial/\partial\phi + iL_\xi \cos\gamma \sin\theta - iL_\eta \sin\gamma \sin\theta - iL_\zeta \cos\theta. \tag{4.20}$$

It is now straightforward to construct l_{BA} in the new coordinate system by putting equations (4.19) and (4.20) into equation (4.14). The result is:

$l_{BA,\,z}$ in the rotated system $= D(\omega) l_z D^{-1}(\omega)$

$$= D(\omega)[-i(\partial/\partial\phi)]D^{-1}(\omega)$$

$$= -i(\partial/\partial\phi) + L_\xi \cos\gamma \sin\theta - L_\eta \sin\gamma \sin\theta - L_\zeta \cos\theta,$$

$l_{BA,\,\pm}$ in the rotated system $= D(\omega) l_\pm D^{-1}(\omega)$

$$= D(\omega)[\exp[\pm i\phi](\pm(\partial/\partial\theta) + i\cot\theta(\partial/\partial\phi))]D^{-1}(\omega)$$

$$= \exp[\pm i\phi][\pm(\partial/\partial\theta) \mp iL_\xi \sin\gamma \mp iL_\eta \cos\gamma + i\cot\theta(\partial/\partial\phi +$$

$$iL_\xi \cos\gamma \sin\theta - iL_\eta \sin\gamma \sin\theta - iL_\zeta \cos\theta)]. \tag{4.21}$$

Equation (4.21) is combined with equation (4.13) and placed in equation (4.10). If the components of **L** in equation (4.13) are written explicitly, as outlined in (4.12), they will cancel most of the terms involving **L** from equation (4.21). I

can then write

$$\mathbf{N} = \mathit{l} + \mathbf{T}, \tag{4.22}$$

where

$$T_x \pm iT_y = L_\varsigma \exp[\pm i\phi]\csc\theta, \quad T_z = 0$$

and

$$l_z = -i(\partial/\partial\phi)$$

$$l_\pm = \exp[\pm i\phi](\pm(\partial/\partial\theta) + i\cot\theta(\partial/\partial\phi)).$$

Writing the components of **N**

$$N_x \pm iN_y = \exp[\pm i\phi][\pm(\partial/\partial\theta) + i\cot\theta(\partial/\partial\phi) + L_\varsigma\csc\theta]$$

$$N_z = -i(\partial/\partial\phi). \tag{4.23}$$

You should notice two points in these equations. Most references to the electrons have disappeared into the operator l. The derivatives implicit in this operator, such as $\partial/\partial\theta$, operate on the electrons as well as the nuclei because the electrons are now referenced to the nuclear framework. That is easy to understand, but if it is true, why is there still an operator, **T**, that operates exclusively on the electrons? ***This operator is unique to linear molecules.*** If another atom is included off axis it disappears, as I will show next.

Suppose you have a non-linear molecule with three atoms, where the "extra" atom has zero nuclear mass such that all the equations already derived

for the center of mass remain correct. All three Euler angles will now be necessary to align the two coordinate systems. In this case, the angular momentum of the nuclei can be written in terms of the Euler angles [10].

$$l_{BA,x} \pm i l_{BA,y} = i e^{\pm i\phi}[\cot\theta(\partial/\partial\phi) \mp i(\partial/\partial\phi\theta) - (1/\sin\theta)(\partial/\partial\gamma)]$$

$$l_{BA,z} = -i(\partial/\partial\phi) \qquad (4.24)$$

Comparing this to the previous definition of l_{BA} given in equation (4.14) shows that the new ability to specify an angle of rotation about the BA axis has added a $(-1/\sin\gamma)(\partial/\partial\gamma)$ term which corresponds to the angular momentum of the nuclei about the internuclear axis.

I have already shown how to handle the first two partial derivatives, but what about $\partial/\partial\gamma$? The procedure is the same as for $(\partial/\partial\gamma)$.

$$D(\omega)(\partial/\partial\gamma)D^{-1}(\omega) = \exp[i\gamma L_\zeta]\exp[i\theta L_\eta]\exp[i\phi L_\zeta](\partial/\partial\gamma)$$

$$\exp[-i\phi L_\zeta]\exp[-i\theta L_\eta]\exp[-i\gamma L_\zeta]$$

$$= \exp[i\phi L_\zeta](\partial/\partial\gamma)\exp[-i\phi L_\zeta]$$

$$= -iL_\zeta + \partial/\partial\gamma$$

Putting this expression for $\partial/\partial\gamma$ into equation (4.24) and then putting the new l_{BA} into equation (4.10) gives

$$\mathbf{N} = \mathbf{l}$$

The mysterious **T** operator has conveniently disappeared. The reason for its

appearance in a diatomic molecule is now obvious. In a non-linear molecule the angular momentum of the electrons about the ς axis is determined by the $(1/\sin\theta)\partial/\partial\gamma$ term in l. However, in a diatomic molecule, the angle about this axis is not specified, so the operator which determines the angular momentum about the ς inter-nuclear axis is not included in l_{BA}. Thus the orbital angular momentum of the electrons about the ς axis must be determined independently of the operator l_{BA}. The operator **T** does just that.

So far, everything has been done in a spherically symmetric system. I will now impose a boundry condition for diatomic molecules on the system. The basis set will be restricted to those states $|\psi>$ such that

$$(N_\varsigma - L_\varsigma)|\psi> = 0.$$

In other words, all the angular momentum about the internuclear axis arises from the electrons.

The previous analysis suffices for Hund's case b. However, in case a, the spins of the electrons must also be referred to the rotating coordinate system. It is easy to include this by using the rotation operator. The only difference is that the angular momentum in $D(\omega)$ will now include the spin angular momentum.

$$D(\omega) = \exp[i\gamma(L_\varsigma+S_\varsigma)]\exp[i\theta(L_\eta+S_\eta)]\exp[i\phi(L_\varsigma+S_\varsigma)]$$

4.4 Parity

4.4.1 Electronic and Rotational Parity

For EuO and YbO, the parity of a Σ state and several case c $\Omega = 0$ states must be determined. Herzberg [2] gives rules for determining parity but gives no explanations, instead referring the reader to a paper written in German by Wigner and Witmer [13]. Judd [6] gives the reasoning behind the rules for cases a and b. I will derive the parity rules for Hund's cases a, b and c for a very special class of diatomic molecules, those in which L is a good quantum number. I will assume that the electronic configurations which the molecular states come from are known.

The action of the inversion operator, E^*, on a basis function which is a parity eigenfunction determines the parity of the eigenfunction. Hougen [4] shows how to break E^* down to the operators σ_v and $C_2(y)$. Specifically,

$$E^*|\eta\Lambda S\Sigma JM\Omega> = \sigma_v(xz)|\eta\Lambda S\Sigma> \quad C_2(y)|JM\Omega>.$$

The effect of σ_v on the single electron orbitals which make up the molecular orbitals must be determined. The single electron orbitals are chosen to be eigenfunctions of the operator l_z, , the projection of l, the orbital angular momentum for a single electron, on the internuclear axis. Therefore, they will have an azimuthal dependence of the type exp[imϕ] [9]. If I choose σ_v to be a reflection in the xz plane of the molecule, the effect of σ_v on the angle ϕ is to change it into the angle 2π - ϕ. Since m is an integer, this changes exp[imϕ] into exp[-imϕ], or the eigenfunction $|lm>$ into $|l\text{-}m>$. The phases of the $|lm>$, however, have not yet been specified. Two possible phase choices would

be

$\sigma_v|lm> = |l\text{-}m>$

or

$\sigma_v|lm> = -|l\text{-}m>$.

Since we want the result to pass smoothly over into the united atom picture, we will borrow the result from atomic states as given by Condon and Shortley [1]

$$\sigma_v Y_{lm}(\theta,\phi) = Y_{lm}(\theta,-\phi) = Y^*_{lm}(\theta,\phi) = (-1)^m Y_{l-m}(\theta,\phi)$$

where $Y_{lm}(\theta,\phi)$ is a spherical harmonic. Therefore, for the single electron molecular orbital, we choose phases so that

$$\sigma_v|lm> = (-1)^m|l\text{-}m>. \tag{4.25}$$

We must also deal with the action of σ_v on the single electron spin eigenfunctions $|s\alpha>$ and $|s\beta>$, where $|s\alpha>$ is the eigenfunction with a projection of $+1/2$ in the molecular frame, while $|s\beta>$ has a projection of $-1/2$. The rules are derived by Judd [8].

$$\sigma_v|s\alpha> = |s\beta>, \quad \sigma_v|s\beta> = -|s\alpha>$$

I can rewrite this as

$$\sigma_v|sm_s> = (-1)^{m_s-1/2}|s\text{-}m_s>, \tag{4.26}$$

where m_s is the projection of spin in the molecular coordinate system.

The next step is to determine the effect of σ_v on the many electron

eigenfunctions $|L\Sigma>$ and $|S\Sigma>$. The many electron eigenfunctions can be written in terms of the single electron eigenfunctions. I will work through an eigenfunction $|L\Lambda>$ made up of three electrons as a demonstration of how to relate $\sigma_v|L\Lambda>$ and $|L-\Lambda>$ for any number of electrons. The angular momenta of electrons 1, 2 and 3 will be designated l_1, l_2 and l_3. $L_2 = l_1 + l_2$. $L = L_2 + l_3$. The projection of l_1 is λ_1 and $\Lambda = \Sigma_t \lambda_t$.

$$|L\Lambda> = |l_3 L_2 L\Lambda> = (-1)^{-l_3+L_2+\Lambda} (2L+1)^{1/2} \Sigma_{\lambda_3\Lambda_2} |l_3\lambda_3> |L_2\Lambda_2>$$

$$\times \begin{pmatrix} l_3 & L_2 & L \\ \lambda_3 & \Lambda_2 & -\Lambda \end{pmatrix}$$

$$|l_1 l_2 L_2\Lambda_2> = (-1)^{-l_1+l_2+\Lambda_2} (2L_2+1)^{1/2} \Sigma_{\lambda_1\lambda_2} |l_1\lambda_1> |l_2\lambda_2>$$

$$\times \begin{pmatrix} l_1 & l_2 & L_2 \\ \lambda_1 & \lambda_2 & -\Lambda_2 \end{pmatrix}.$$

Putting the two equations together gives

$$|L\Lambda> = |l_3 L_2 L\Lambda> = (-1)^{-l_3+L_2-l_1+l_2+\Lambda+\Lambda_2} [(2L+1)(2L_2+1)]^{1/2}$$

$$\Sigma_{\lambda_3\Lambda_2\lambda_1\lambda_2} |l_1\lambda_1> |l_2\lambda_2> |l_3\lambda_3>$$

$$\times \begin{pmatrix} l_3 & L_2 & L \\ \lambda_3 & \Lambda_2 & -\Lambda \end{pmatrix} \begin{pmatrix} l_1 & l_2 & L_2 \\ \lambda_1 & \lambda_2 & -\Lambda_2 \end{pmatrix}.$$

Now operate on this decoupled function with σ_v. I will use equation (4.25).

$$\sigma_v|L\Lambda> = |l_3 L_2 L\Lambda> = (-1)^{-l_3+L_2-l_1+l_2+\Lambda+\Lambda_2} [(2L+1)(2L_2+1)]^{1/2}$$

$$\times \Sigma_{\lambda_3\Lambda_2\lambda_1\lambda_2} (-1)^{\lambda_1+\lambda_2+\lambda_3} |l_1-\lambda_1> |l_2-\lambda_2> |l_3-\lambda_3>$$

$$\times \begin{pmatrix} l_3 & L_2 & L \\ \lambda_3 & \Lambda_2 & -\Lambda \end{pmatrix} \quad \begin{pmatrix} l_1 & l_2 & L_2 \\ \lambda_1 & \lambda_2 & -\Lambda_2 \end{pmatrix} \tag{4.27}$$

$(-1)^{\lambda_1+\lambda_2+\lambda_3}$ equals $(-1)^{\Lambda}$ because $\Sigma\lambda_i = \Lambda$. The next step is to compare equation (4.27) to the expansion for $|L\text{-}\Lambda>$.

$$|L\text{-}\Lambda> = (-1)^{-l_3+L_2-l_1+l_2-\Lambda-\Lambda_2} \quad [(2L+1)(2L_2+1)]^{1/2}$$

$$\times \Sigma_{\lambda_3\Lambda_2\lambda_1\lambda_2} \quad |l_1\text{-}\lambda_1> \; |l_2\text{-}\lambda_2> \; |l_3\text{-}\lambda_3>$$

$$\times \begin{pmatrix} l_3 & L_2 & L \\ -\lambda_3 & -\Lambda_2 & \Lambda \end{pmatrix} \quad \begin{pmatrix} l_1 & l_2 & L_2 \\ -\lambda_1 & -\lambda_2 & \Lambda_2 \end{pmatrix} \tag{4.28}$$

Comparing equations (4.27) and (4.28) and using the following property of 3-j symbols

$$\begin{pmatrix} j_1 & j_2 & j \\ m_1 & m_2 & m \end{pmatrix} = (-1)^{j_1+j_2+j} \begin{pmatrix} j_1 & j_2 & j \\ -m_1 & -m_2 & -m \end{pmatrix}$$

gives

$$\sigma_v|L\Lambda> = (-1)^{\Sigma l+L+2\Lambda+2(L_2+\Lambda_2)} \; (-1)^{\Lambda} \; |L\text{-}\Lambda>.$$

Note that the first power of (-1) comes completely from the properties of the expansion and 3-j coefficients; therefore it will be the same for $|S\Sigma>$, while the $(-1)^{\Lambda}$ term depends on the action of σ_v on the single electron eigenfunctions. The equation can be simplified slightly to give

$$\sigma_v|L\Lambda> = (-1)^{L+\Sigma l-\Lambda} \; |L\text{-}\Lambda>, \tag{4.29}$$

where I have used the fact that $(-1)^{3\Lambda} = (-1)^{-\Lambda}$, which is true for Λ integer or ,if I use these results for spin, half-integer.

Even though this example has been worked out for three electrons, the result will be the same for any number of electrons since the only change is a factor of $(-1)^{2(L_i+\Lambda_i)}$ for each intermediate angular momentum and this always gives a factor of +1.

For $|S\Sigma>$ the term which comes from the expansion of the 3-j coefficients is equivalent to that for L, $(-1)^{\Sigma s+S+2\Sigma+2(S_2+\Sigma_2)} = (-1)^{S+2\Sigma+n/2}$, where n is the number of electrons. Next I must find the term equivalent to $(-1)^{\Lambda}$ which results from the operation of reflection on the single electron eigenfunctions. Equation (4.26) gives the result for a single electron eigenfunction.

$$\sigma_v|sm_s> = (-1)^{m_s-1/2}|s-m_s>$$

When this is summed over n single electron orbitals, the factor $(-1)^{\Sigma - n/2}$ results (note that in the exponent Σ is not the summation symbol but the sum of the m_s). Putting these two terms together

$$\sigma_v|S\Sigma> = (-1)^{S+3\Sigma}|S-\Sigma> = (-1)^{S-\Sigma}|S-\Sigma>. \tag{4.30}$$

That takes care of the parity of the electronic part of the eigenfunctions, but the rotational term must still be determined. Hougen [4] and Larsson [11] show how to derive the rotational parity. For cases a and c the results are

$$C_2(y)|JM\Omega> = (-1)^{J-\Omega}|JM-\Omega> \tag{4.31}$$

while for case b

$$C_2(y)|NM_N\Lambda> = (-1)^{N-\Lambda}|NM_N-\Lambda>. \tag{4.32}$$

4.4.2 Parity in Hund's Case a

Applying equations (4.29), (4.30) and (4.31) to the basis set functions of case a gives

$$E^*|L\Lambda S\Sigma J\Omega> = (-1)^{L+\Sigma\ell-\Lambda+S-\Sigma+J-\Omega} \quad |L-\Lambda S-\Sigma J-\Omega>$$

The parity eigenfunctions can constructed by taking a linear combination of the basis set functions,

$$\text{Parity eigenfunction} = |L\Lambda S\Sigma J\Omega> \pm (-1)^{L+\Sigma\ell-\Lambda+S-\Sigma+J-\Omega}$$

$$|L-\Lambda S-\Sigma J-\Omega>.$$

4.4.3 Parity in Hund's Case b

Using the results of equations (4.29) and (4.32) on the basis set of case b gives

$$E^*|L\Lambda N\Lambda SM_s JM_j> = (-1)^{L+\Sigma\ell+N} |L-\Lambda N-\Lambda SM_s JM_j>.$$

The parity eigenfunctions are

$$|L\Lambda N\Lambda SM_s JM_j> \pm (-1)^{L+\Sigma\ell+N} |L-\Lambda N-\Lambda SM_s JM_j>.$$

4.4.4 Parity in Hund's Case c

Case c couples $|S\Sigma>$ and $|L\Lambda>$ to produce $|J_a\Omega>$. To determine the action of σ_v on $|J_a\Omega>$ I will use the same procedure as was used for building up multi-electron eigenfunctions from single electron eigenfunctions.

$$\sigma_v|SLJ_a\Omega\rangle = \sigma_v[(-1)^{-S+L+\Omega}\ (2J_a)^{1/2}\ \Sigma_{\Lambda\Sigma}\ |S\Sigma\rangle\ |L\Lambda\rangle$$

$$\times \begin{pmatrix} S & L & J_a \\ \Sigma & \Lambda & -\Omega \end{pmatrix}]$$

$$= (-1)^{-S+L+\Omega}\ (-1)^{S+L+J_a}\ (-1)^{L+\Sigma l+\Lambda+S-\Sigma}\ (2J_a)^{1/2}$$

$$\times \Sigma_{\Lambda\Sigma}\ |S\text{-}\Sigma\rangle|L\text{-}\Lambda\rangle \quad \begin{pmatrix} S & L & J_a \\ -\Sigma & -\Lambda & \Omega \end{pmatrix}$$

$$= (-1)^{2L+\Sigma l+2S+J_a+\Omega}|J_a\text{-}\Omega\rangle$$

$$= (-1)^{\Sigma l+2S+J_a+\Omega}|J_a\text{-}\Omega\rangle \qquad (4.33)$$

This gives the electronic parity. The rotational parity is determined by equation (4.31) so the total parity is determined by

$$E^*|J_a\Omega\rangle = (-1)^{\Sigma l+2S+J_a+J}|J_a\text{-}\Omega\rangle$$

The parity eigenfunctions are

$$|J_a\Omega\rangle \pm (-1)^{\Sigma l+2S+J_a+J}|J_a\text{-}\Omega\rangle$$

While the 2S term may appear alarming since in case c S is not known, it is actually relatively harmless. This term will always be even for J = integer systems and odd for J = half-integer systems.

4.4.5 Parity in a Separated Atom Approach

By using the results from above for the united atom approach the parity in the separated atom approach is trivial. Determine the parity of each atom

using the rules for the united atom approach. Since the eigenfunctions in the separated atom picture are of the type |atom 1>|atom 2>, the parity of the molecule will then be given by (parity of atom 1)(parity of atom 2). Remember that this is so simple only because I have assumed L is a good quantum number, which is an approximately valid assumption for the states f^{N+1} and $F^N s$ states examined in EuO and YbO.

4.5 Conclusion

I think the best way to end this chapter is to show how I have answered the questions I listed at the beginning.

1. All the angular momentum theory for a diatomic molecule is carried out for a spherically symmetric system. There are several practical concessions made for a diatomic molecule. It is admitted that in general the electron orbital angular momentum, **L**, is not well known. Only its projection on the internuclear axis is well determined. Futhermore basis functions are restricted to those for which

$$[J_\varsigma - (L_\varsigma + S_\varsigma)]|\psi> = 0$$

for cases a and c and to

$$[N_\varsigma - L_\varsigma]|\psi> = 0$$

for case b.

2. The total angular momentum, or any other angular momentum for that matter, is allowed to have two projections, one in the lab and one in the molecule, provided we do not attempt to specify the Euler angles which relate the two coordinate systems. Since we have no idea how the two projections are oriented with respect to

each other, we are not violating any quantum mechanical principle.

3. The angular momentum operator does not appear magically in the rotation operator. It appears because it is the correct operator for infinitesimal rotations.

4. In case a, the true uncoupled basis set would be $|L\Lambda;S\Sigma;RR_\zeta\rangle$ where **R** is the rotational angular momentum. Instead, we choose an "uncoupled" basis set where the "total" angular momentum is given by

$$\mathbf{R} = \mathbf{J} - \mathbf{L} - \mathbf{S}$$

R becomes the total angular momentum, while **J**, **L** and **S** are the uncoupled angular momenta.

5. In case a, spin is referred to the molecular coordinates, while in case b it is quantized in the lab. In order to go from case a basis sets to case b, the spin coordinate system must first be transformed from the molecule to the lab after which the angular momenta can be recoupled. Going from case a to case c does not present this problem because spin is defined in the molecular system in both cases.

6. The danger here is in writing the basis function as $|J_a J\Omega\rangle$ as opposed to the correct way $|J_a\Omega;J\Omega\rangle$. The two projections "happen" to be the same (actually, we forced this to be true (see point 1)) but they are independent. Therefore, the operation of $J^+J_a^-$ on $|J_a J\Omega\rangle$ should be written in the following way:

$$J^+J_a^-|J_a J\Omega\rangle = J^+|J\Omega\rangle J_a^-|J_a\Omega\rangle =$$

$$(\text{number})|J\Omega\text{-}1\rangle|J_a\Omega\text{-}1\rangle$$

7. The anomalous commutation rules come about because, in the molecule fixed system, the correct operator for the total angular momentum is $D(\omega)JD^{-1}(\omega)$ rather than **J**.

8. You never add projections of angular momenta to angular

momenta. If you think in terms of doing so, sooner or later you will run into trouble.

9. The rules for the determination of absolute parity when **L** is a good quantum number have been derived, in English.

I hope I have been able to show you that the theory of angular momentum in diatomic molecules is quite understandable provided you start with the basic theory and work your way up very carefully.

References

[1] E. U. Condon and G. H. Shortley.
The Theory of Atomic Spectra.
University Press, Cambridge, England, 1963.

[2] G. Herzberg.
Molecular Spectra and Molecular Structrue.
Van Nostrand Reinhold Company, New York, 1950.

[3] G. Herzberg.
Molecular Spectra and Molecular Structrue.
Van Nostrand Reinhold Company, New York, 1950.
page 219.

[4] J. T. Hougen.
The Calculation of Rotational Energy Levels and Rotational Line Intensities in Diatomic Molecules.
NBS Monograph 115, National Bureau of Standards, June, 1970.

[5] B. R. Judd.
Angular Momentum Theory for Diatomic Molecules.
Academic Press, New York, 1975.

[6] B. R. Judd.
Angular Momentum Theory for Diatomic Molecules.
Academic Press, New York, 1975.
page 128.

[7] B. R. Judd.
Angular Momentum Theory for Diatomic Molecules.
Academic Press, New York, 1975.
Chapter 6.

[8] B. R. Judd.
Angular Momentum Theory for Diatomic Molecules.
Academic Press, New York, 1975.
page 129.

[9] B. R. Judd.
Angular Momentum Theory for Diatomic Molecules.
Academic Press, New York, 1975.
page 130.

[10] B. R. Judd.
Angular Momentum Theory for Diatomic Molecules.
Academic Press, New York, 1975.
page 9.

[11] M. Larsson.
Phase Conventions for Rotating Diatomic Molecules.
Physica Scripta 23:835, 1981.

[12] B. L. Silver.
Irreducible Tensor Methods.
Academic Press, New York, 1976.

[13] E. Wigner and E. E. Witmer.
Uber die Struktur der Zweiatomigen Molekelspektren nach der Quantenmechanik.
Z. Physik 51:859, 1928.

[14] A. A. Wolf.
Rotation Operators.
American Journal of Physics 37(5), 1969.

Chapter 5
Ligand Field Model

5.1 Introduction

This chapter provides an outline of the ligand field model which has been derived to predict the energy levels of rare-earth oxide molecules. I will demonstrate how to apply the model to EuO and YbO. For a more thorough derivation and discussion of the model, the reader is refered to Dulick [3] or Field [6].

The theory behind the model is very straightforward. For a diatomic metal oxide, MO, one starts with the energy levels of the free metal ion M^{2+}. Then a doubly negative point charge, representing the O^{2-} ion, is placed the same distance away from the M^{2+} as the oxygen atom is in the molecule. The axially symmetric electric field from this point charge splits the energy levels of the M^{2+} ion in a way that can be predicted by an electrostatic calculation based on Crystal Field Theory, which neglects the overlap of M^{2+} and O^{2-} centered orbitals.

The main assumption is that the O^{2-} ion can be represented as a polarizable point charge, that the overlap of the orbitals on the M^{2+} ion with the O^{2-} orbitals is small. For 4f orbitals on the metal this is an excellent assumption since they are very small. The valence orbitals, with the principal quantum number equal to six or greater, are another matter. These orbitals should significantly overlap those on the oxygen. However, if the radial

parameters for these electrons are treated as adjustable parameters, the model is still very useful. For example, with a $4f^N6s$ configuration, the splitting of the free metal ion levels will be done by the interaction of the ligand field with the 4f electrons. The 6s electron contributes a term which is a constant for the entire configuration. Since the overlap approximation is excellent for 4f orbitals, the calculation should give reasonable results.

5.2 The Ligand Field Hamiltonian

The tensor form of the ligand field Hamiltonian is [18]

$$H_{lf} = \Sigma_{kqi}\, B^k_q (C^k_q)_i,$$

where the sum over i is the sum over electrons and the sum over k comes from using a multipole expansion for the separation of the electron from the location of the perturbing point charge (the O^{2-} in this case). As will be demonstrated, the number of terms in the sum over k is limited by angular factors. The C^k_q are defined by the following equation [14].

$$C^k_q = [4\pi/(2k+1)]^{1/2}\ \ Y^k_m(\theta,\phi)$$

where $Y^k_m(\theta,\phi)$ is a spherical harmonic. B^k_q is a radial term which can be calculated from a knowledge of the radial behavior of the M^{2+}-centered orbitals.

The symmetry of the $C_{\infty v}$ point group permits the use of only the axial C^k_0 components, all others are zero. Therefore the ligand field Hamiltonian for a diatomic molecule is

$$H_{lf} = \Sigma_{ki} B_0^k (C^k{}_0)_i$$

with

$$B_0^k = Ze^2 \int_0^\infty \frac{r_<^k}{r_>^{k+1}} R^2(r)\, dr \qquad (5.1)$$

where

R_e = the distance from the metal nucleus to the oxygen atom
Z = the charge of the ligand = -2 for $M^{2+}O^{2-}$
e = charge of electron
$r_<$ = the integration variable r if the electron is within a sphere centered at the nucleus of radius R_e
= R_e if the electron is outside of the sphere
$r_>$ = R_e if electron is within sphere of radius R_e
= r if electron is outside of this sphere
$R(r)$ = the function which describes the radial behavior of the electron.

As a demonstration, I will apply this Hamiltonian to single electron orbitals of s and f electrons. For the 6s electron,

$$\langle 00|H_{cf}|00\rangle = \Sigma_k B_0^k(6s)\langle 00|C_0^k|00\rangle$$

using the Wigner-Eckart theorem [16],

$$\langle 00|C_0^k|00\rangle = (-1)^0 \begin{pmatrix} 0 & k & 0 \\ 0 & 0 & 0 \end{pmatrix} \langle 0||C^k||0\rangle.$$

A formula for the reduced matrix elements of C^k is given by Edmonds [5].

$$\langle 0||C^k||0\rangle = \begin{pmatrix} 0 & k & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$= 1 \text{ for } k = 0$$

$= 0$ for $k \neq 0$.

For a single electron in a 6s orbital

$$\langle 6s|H_{lf}|6s\rangle = \Sigma_k \, B_0^k(6s)\langle 00|C_0^k|00\rangle$$

$$= B_0^0(6s).$$

For a 4f electron, $l = 3$, so

$$\langle 3m'|H_{lf}|3m\rangle = \Sigma_k \, B_0^k(4f)\langle 3m'|C_0^k|3m\rangle$$

$$\langle 3m'|C_0^k|3m\rangle = (-1)^{3-m'} \begin{pmatrix} 3 & k & 3 \\ m' & 0 & m \end{pmatrix} \langle 3||C^k||3\rangle$$

$$\langle 3||C^k||3\rangle = (-1)^3 (7) \begin{pmatrix} 3 & k & 3 \\ 0 & 0 & 0 \end{pmatrix}$$

For a 3-j symbol,

$$\begin{pmatrix} j_1 & j_2 & j \\ m_1 & m_2 & -m \end{pmatrix} = (-1)^{j_1+j_2+j} \begin{pmatrix} j_1 & j_2 & j \\ -m_1 & -m_2 & m \end{pmatrix} ,$$

so for the 3-j symbol with all m's equal to zero, the sum of the j's must be even. This restriction, in addition to the triangle condition which limits k to values between 0 and 6, restricts the possible values of k to 0, 2, 4, and 6. Using the numerical values for the 3-j symbols tabulated in Rotenberg et. al. [13],

$$\langle 3||C^0||3\rangle = [7]^{1/2}$$

$$\langle 3||C^2||3\rangle = -2[7/15]^{1/2}$$

$$\langle 3||C^4||3\rangle = [14/11]^{1/2}$$

$$\langle 3||C^6||3\rangle = -10[7/429]^{1/2}.$$

I will evaluate the other 3-j term for m = 2,

$$\begin{pmatrix} 3 & 0 & 3 \\ -2 & 0 & 2 \end{pmatrix} = -[1/7]^{1/2}$$

$$\begin{pmatrix} 3 & 2 & 3 \\ -2 & 0 & 2 \end{pmatrix} = 0$$

$$\begin{pmatrix} 3 & 4 & 3 \\ -2 & 0 & 2 \end{pmatrix} = (1/3)[7/22]^{1/2}$$

$$\begin{pmatrix} 3 & 6 & 3 \\ -2 & 0 & 2 \end{pmatrix} = [3/1001]^{1/2}.$$

Putting all the terms together for m = 2,

$$\langle lm|H_{lf}|lm\rangle = \langle 3\pm2|H_{lf}|3\pm2\rangle = B_0^0 - (7/33)B_0^4 + (10/143)B_0^6.$$

As I have shown, the angular terms are fairly easy to calculate. The radial terms, the B_0^k, are not quite so easy. The integral given in equation (5.1) can be divided into two parts, $r \leq R_e$ and $r > R_e$. Writing this in detail gives

$$B_0^k = Ze^2 \int_0^\infty \frac{r_<^k}{r_>^{k+1}} R^2(r)\, dr$$

$$= Ze^2 \int_0^{R_e} \frac{r^k}{R_e^{k+1}} R^2(r)\, dr + Ze^2 \int_{R_e}^\infty \frac{R_e^k}{r^{k+1}} R^2(r)\, dr.$$

To evaluate the integrals numerically, the form of the radial distribution of the electron orbital, R(r), is needed. A numerical distribution for R(r) in M^{2+} can

be obtained from a Hartree-Fock SCF calculation. Herman and Skillman [8] present a program for doing the calculation. The results of a ligand field calculation on YbO and EuO using Hartree-Fock wavefunctions [12] are listed in Table 5-I.

Table 5-I: Calculated Ligand Field Radial Parameters

EuO

$B_0^0(4f) = 129.043\ cm^{-1}$

$B_0^0(6s) = 115.736\ cm^{-1}$

YbO

$B_0^0(4f) = 129.043\ cm^{-1}$

$B_0^0(6s) = 115.736\ cm^{-1}$

$B_0^2(4f) = 7003\ cm^{-1}$

$B_0^4(4f) = 1068\ cm^{-1}$

$B_0^6(4f) = 382\ cm^{-1}$

With the integral B_0^0 and the energy levels of the ion, the relative ordering of the configurations in the molecule can be predicted. For example the center of gravity of levels in the $4f^{14}$ and $4f^{13}6s$ configurations of YbO and will have the following energies:

$$E(f^{14}) = (\text{energy of } f^{14} \text{ in ion}) + 14 \times B_0^0(4f)$$

$$E(f^{13}s) = (\text{energy of } f^{13}s \text{ in ion}) + 13 \times B_0^0(4f) + B_0^0(6s).$$

Thus,

$$E(f^{13}s) - E(f^{14}) = \Delta E_{ion} + B_0^0(6s) - B_0^0(4f).$$

The $4f^{14}$ configuration gives rise to one level which is the ground state of the Yb^{2+} ion [11]. The center of gravity of the $4f^{13}6s$ configuration is 39,369 cm^{-1} above the ground state in the ion. Therefore, in YbO Ligand Field Thoery predicts that the $4f^{14}$ configuration will lie about 26,062 cm^{-1} below the center of gravity of the $4f^{13}6s$ configuration. The ionic levels of the $4f^{13}6s$ configuration will be split by the higher order terms in the ligand field, but these terms should not shift the center of gravity of the configuration.

A similar calculation on EuO predicts that the $4f^7$ configuration will lie 32,789 cm^{-1} below the lowest $4f^6 6s$ level.

5.3 Predicted Configurations

The spherically symmetric term in the ligand field Hamiltonian (B_0^0) is by far the largest. Its effect is to shift the relative energies of the electon configurations in the ion. The predicted ground states of the Ba-Hf monoxides are shown in Table 5-II. In the table the molecular orbitals σ is located mainly on the oxygen atom while the σ^*, formed from the 6s and 6p orbitals on the metal ion and the $2p_z$ orbital on the oxygen ion, is more polarized towards the metal atom. In all but EuO and YbO, the most stable molecular configuration of the rare earth monoxides becomes $4f^N s$ [6] where N + 1 is the number of electrons outside filled shells in the metal ion. EuO and YbO are exceptions because of the stability of the lowest L-S term of the half-filled EuO (f^7) and filled YbO (f^{14}) configurations. The location and parent configuration of the low energy levels of YbO and EuO is one of the crucial questions my experiments were designed to answer.

Table 5-II brings out another important feature of the rare-earth oxides.

Table 5-II: Predicted Ground States for the Oxides of Ba-Hf

Molecule	$\Delta G(1/2)$ cm^{-1}	Predicted Configuration	Predicted ground state
BaO	666[a]	σ^2	$^1\Sigma^1$
LaO	808[a]	$\sigma^2\sigma^* f^0$	$^2\Sigma^+$
CeO	824[b].820[f].808[g]	$\sigma^2\sigma^* f^1$	$\Omega = 2$
PrO	828[c].817[d].818[g]	$\sigma^2\sigma^* f^2$	$\Omega = 3\ 5$
NdO	823[f].815[g]	$\sigma^2\sigma^* f^3$	$\Omega = 4$
PmO	--	$\sigma^2\sigma^* f^4$	$\Omega = 0\ 5$
SmO	819[f].808[g]	$\sigma^2\sigma^* f^5$	$\Omega = 0^-$
EuO	668[d]	$\sigma^2 f^7$	$^8\Sigma^-$
GdO	825[e].824[f].813[g]	$\sigma^2\sigma^* f^7$	$^9\Sigma^-$
TbO	824[d.g]	$\sigma^2\sigma^* f^8$	$\Omega = 6\ 5$
DyO	829[g]	$\sigma^2\sigma^* f^9$	$\Omega = 8$
HoO	829[g]	$\sigma^2\sigma^* f^{10}$	$\Omega = 8\ 5$
ErO	829[g]	$\sigma^2\sigma^* f^{11}$	$\Omega = 0^-$
TmO	832[g]	$\sigma^2\sigma^* f^{12}$	$\Omega = 0\ 5$
YbO	680[h]	$\sigma^2 f^{14}$	$\Omega = {}^1\Sigma^+$
LuO	836[a.f].825[g]	$\sigma^2\sigma^* f^{14}$	$\Omega = {}^2\Sigma^+$
HfO	968[a]	$\sigma^2\sigma^{*2} f^{14}$	$\Omega = {}^1\Sigma^+$

a. Ref [9]. gas phase
b Ref [10]. gas phase
c Ref [4]. gas phase
d. Ref [7]. Ar Matrix
e Ref [19]. gas phase
f Ref [2]. Ne Matrix
g. Ref [2]. Ar Matrix
h. This work

The states which come from a $4f^{N-1}6s$ configuration all have a vibrational frequency, $\Delta G(1/2)$, of $\sim$820 cm^{-1}, while those that come from $4f^N$ have a $\sim$670 cm^{-1}. This is fairly easy to explain with the ligand field model. The orbital which comes from the 6s is much larger than the 4f orbital [6], in fact a significant fraction of the electron density in the σ^* orbital lies at $r > R_e$.

Consequently, the doubly negative oxygen ion will see a metal ion with a charge of about +5/2. When the configuration is $4f^{N+1}$ the electron is now in a small 4f orbital which effectively shields the O^{2-} from the metal ion, making the effective charge of the metal ion = +2. One other point to make about vibrational frequencies is that, for both EuO and YbO, we suspect, from tentative experimental vibrational assignments, that some of the configurationally excited states have a vibrational frequency of $\sim$550 cm^{-1}. Based on this frequency we suggest that the states come from a M^+O^- configuration. The reasoning goes as follows. An electron is removed from an molecular orbital which is primarily on the oxygen atom to one that is primarily on the metal atom. This decreases the electrostatic attraction between the two ions and thus lowers the vibrational frequency.

5.4 Qualitative Effect of the Ligand Field

Before deriving the Hamiltonian, it is helpful to gain a qualitative understanding of the effect of the ligand field. The closer the metal ion electrons are to the O^{2-}, the higher the energy. Therefore the higher the projection of electron angular momentum on the internuclear axis, the lower the energy, while a low projection means the electron is in an orbital which brings it close to the oxygen. In the $f^{13}s$ configuration, one can think in terms of a hole instead of an electron. An orbital which brings the hole close to the negatively charged oxygen will be lower in energy. Thus, $\Omega = 0$ will be the lowest energy orbital.

5.5 Ligand Field Calculation for YbO

The atomic basis set must be chosen before doing any calculations. In the $f^{13}s$ comfiguration of YbO the spin-orbit splitting is far greater than the electrostatic, so j-j coupling is a better choice than ls coupling for the single electron orbitals. Consequently, the basis set will be $|J_c j_s J_a \Omega>$.

In addition to the ligand field Hamiltonian, the Hamiltonian for the unperturbed ion must also be included. There are two interactions to be included, the Coulomb and spin-orbit. The Hamiltonian for the Coulomb term is given by [14]

$$H_c = e^2/r_{12} = e^2\Sigma_k(r_<^k/r_>^{k+1}) \quad (C_1^k \cdot C_2^k) \tag{5.2}$$

while the spin-orbit Hamiltonian is [14]

$$H_{so} = \Sigma_i \, \zeta(r) l_i \cdot s_i.$$

I've already shown how the B_0^0 ligand field term affects the energy of the $f^{13}s$ configuration relative to the f^{14} configuration. The matrix elements for the splitting of levels within the $f^{13}s$ configuration will be derived for an fs configuration, which will be identical to those of an $f^{13}s$ configuration except for the appropriate sign changes. The first term to consider, the one which causes the largest splitting, is the spin-orbit energy due to the f electron. First, the Wigner-Eckart theorem is used on the matrix elements. [14]

$$<j_f j_s J_a \Omega| \; H_{so} \; |j_f' j_s J_a' \Omega'> = \Sigma_i \, <j_f' j_s J_a' \Omega'| \; \zeta l_i \cdot s_i \; |j_f' j_s J_a \Omega>$$

$$= (-1)^{J_a' - \Omega'} \, \Sigma_i \begin{pmatrix} J_a' & 0 & J_a \\ -\Omega & 0 & \Omega \end{pmatrix} \quad <j_f' j_s J_a' || \; \zeta l_i \cdot s_i \; || j_f' j_s J_a>$$

The spin-orbit operator, $l \cdot s$, will yield zero when it operates on the s electron. To operate on the f electron, j_f must be decoupled from j_s.

$$\langle j_f' j_s J_a' || \; l_i \cdot s_i \; || j_f j_s J_a \rangle = (-1)^{j_f + j_s + J_a'} \left[(2J_a+1)(2J_a'+1)\right]^{1/2}$$

$$\times \begin{Bmatrix} j_f & J_a & j_s \\ J_a' & j_f' & 0 \end{Bmatrix} \langle j_f || l_i \cdot s_i || j_f' \rangle$$

Next l and the spin of the f electron must be decoupled:

$$\langle j_f || l_i \cdot s_i || j_f' \rangle = \langle \ell j || l_i \cdot s_i || \ell j' \rangle$$

$$= \langle 3 \; 1/2 \; j || l_i \cdot s_i || 3 \; 1/2 \; j' \rangle$$

$$= (-1)^{l+s+j} \; (2j+1)^{1/2} \begin{pmatrix} 3 & 1/2 & j \\ 3 & 1/2 & 1 \end{pmatrix} \langle 3 || l || 3 \rangle \; \langle 1/2 || s || 1/2 \rangle$$

$$\langle 3 || l || 3 \rangle = [l(l+1)(2l+1)]^{1/2} = 2[21]^{1/2}$$

$$\langle 1/2 || s || 1/2 \rangle = [3/2]^{1/2}.$$

I will demonstrate how to calculate the matrix elements only for $j=j'=7/2$, $J_a=J_a'=3$. A complete listing of all the elements is given in Table 5-III. I need the values of the following 3-j coefficients:

$$\begin{pmatrix} 3 & 0 & 3 \\ -\Omega & 0 & \Omega \end{pmatrix} = -1/[7]^{1/2}$$

$$\begin{pmatrix} 3 & 1/2 & 7/2 \\ 3 & 1/2 & 1 \end{pmatrix} = -1/[56]^{1/2}$$

$$\begin{pmatrix} 7/2 & 3 & 1/2 \\ 3 & 7/2 & 0 \end{pmatrix} = -1/[56]^{1/2}.$$

Putting it all together

$$<7/2\ 1/2\ 3\ 0|\ H_{so}\ |7/2\ 1/2\ 3\ 0> = (-1)^3\ (-1/[7]^{1/2})\ \ (7)\ (-1)^{7/2+1/2+3}$$

$$(-1/[56]^{1/2})\ (-1)\ [8]^{1/2}\ (-1/[56]^{1/2})\ 2[21]^{1/2}\ [3/2]^{1/2}$$

$$= (3/2)\varsigma.$$

The next term to consider is the Coulomb term. The form of the Coulomb Hamiltonian is given by [15]

$$H_{el} = e^2\ \Sigma_k\ r_<^k/r_>^{k+1}\ \ C_1^k.C_2^k$$

where the 1 and 2 refer to electrons 1 and 2. The matrix elements can be expressed in the following form [1]

$$<\eta j_f j_s J_a \Omega|H_{el}|\eta j_f' j_s J_a' \Omega'> = \ f_k F^k - g_k G^k$$

where

$$F^k(n^a l^{\ a}, n^b l^{\ b})$$

$$= e^2 \int_0^\infty \int_0^\infty (r_<^k/r_>^{k+1})\ R_1^{\ 2}(n^a l^{\ a}) R_2^{\ 2}(n^b l_b) dr_1 dr_2$$

$$G^k(n^a l^{\,a}, n^b l^{\,b})$$

$$= 2e^2 \int_0^\infty dr_2 \int_0^{r_2} (r_1^k / r_2^{k+1})\; R_1(n^a l^{\,a}) R_1(n^b l^b) R_2(n^a l^{\,a}) R_2(n^b l_b) dr_1$$

$$f_k = \langle (j_f)_1 (j_s)_2 J_a \Omega |\; C_1^k \cdot C_2^k \;| (j_f)_1' (j_s)_2 J_a' \Omega' \rangle$$

and

$$g_k = \langle (j_f)_1 (j_s)_2 J_a \Omega |\; C_1^k \cdot C_2^k \;| (j_f)_2' (j_s)_1 J_a' \Omega' \rangle .$$

Applying the Wigner-Eckart theorem to f_k, [17]

$$f_k = \delta_{J_a J_a'} \delta_{\Omega\Omega'} \; (-1)^{j_f + j_s + J_a} \begin{Bmatrix} j_f & j_s & J_a \\ j_s & j_f' & k \end{Bmatrix} \langle j_f || C^k || j_f \rangle$$

$$\langle j_s || C^k || j_s \rangle ,$$

while the exchange integral term is given by

$$g_k = \delta_{J_a J_a'} \delta_{\Omega\Omega'} \; (-1)^{2j_s + J_a} \begin{Bmatrix} j_f & j_s & J_a \\ j_f' & j_s & k \end{Bmatrix} \langle j_f || C^k || j_s \rangle \;\; \langle j_s || C^k || j_f' \rangle .$$

Before the reduced matrix elements can be evaluated, l and s must be decoupled:

$$\langle l s j || C^k || l' s' j' \rangle = (-1)^{l + s + j' + k} \; [(2j+1)(2j'+1)]^{1/2} \begin{Bmatrix} l & j & s \\ j' & l' & k \end{Bmatrix}$$

$$\langle l || C^k || l' \rangle$$

$$\langle l||C^k||l'\rangle = (-1)^l [(2l+1)(2l'+1)]^{1/2} \begin{pmatrix} l & k & l' \\ 0 & 0 & 0 \end{pmatrix} \quad (5.3)$$

For the exchange integral, $l = 3$ and $l' = 0$. The 3-j symbol then requires that k = 3. Putting these values into equation (5.3) gives

$$\langle 3||C^k||0\rangle = 1.$$

The complete values for g_k are given in Table 5-III.

The matrix elements of f_k are all zero except for f_0. Since the Coulomb term, f_0F^0, will be the same for every level in the configuration, this term in the Hamiltonian will be combined with the other constant terms.

Finally the ligand field terms must be evaluated. The matrix element is

$$\langle j_f j_s J_a|C_0^k|j_f' j_s J_a'\rangle = (-1)^{J_a-\Omega} \begin{pmatrix} J_a & k & J_a' \\ -\Omega & 0 & \Omega' \end{pmatrix} \langle j_f j_s J_a||C^k||j_f' j_s J_a'\rangle$$

$$\langle j_f j_s J_a||C^k||j_f' j_s J_a'\rangle = (-1)^{j_f+j_s+J_a'+k} [(2J_a+1)(2J_a'+1)]^{1/2} \begin{Bmatrix} j_f & J_a & j_s \\ J_a' & j_f' & k \end{Bmatrix}$$

$$\times \langle j_f||C^k||j_f'\rangle$$

$$\langle ls j_f||C^k||ls j_f'\rangle = (-1)^{l+s+j_f'+k} [(2j_f+1)(2j_f'+1)]^{1/2} \begin{Bmatrix} l & j_f & j_s \\ j_f' & l & k \end{Bmatrix}$$

$$\times \langle l||C^k||l\rangle$$

$$\langle l||C^k||l\rangle = (-1)^l (2l+1) \begin{pmatrix} l & k & l \\ 0 & 0 & 0 \end{pmatrix}$$

Since $l = 3$, k = 0, 2, 4 or 6. Using these values along with $j_s = 1/2$, the

coeffecients can be calculated. They are listed in Table 5-III with a change of sign to account for the $f^{13}s$ vs. fs configuration.

Table 5-III: Hamiltonian Matrix Elements for the Yb^{2+} $4f^{13}6s$ Configuration

Electrostatic Interaction

$$G3 = G_3(4f^{13}6s) = G^3(4f^{13}6s)/7$$

Spin-Orbit Interaction

$$Zeta = \zeta_{4f}$$

Ligand Field Interaction

$$B2 = B_0^2(4f)/420$$

$$B4 = B_0^4(4f)/2772$$

$$B6 = B_0^6(4f)\cdot 5/1716$$

Multiply all off-diagonal Ligand Field matrix elements by the square root of the number in the ODME column of the same row.

Hamiltonian Matrix for the f13s Configuration

State									G3	ZETA	B2	B4	B6	ODME
5/2	1/2	2	2	5/2	1/2	2	2		1	2	96	-44	0	
5/2	1/2	2	1	5/2	1/2	2	1		1	2	-48	176	0	
5/2	1/2	2	0-	5/2	1/2	2	0-		1	2	-96	-264	0	
5/2	1/2	3	3	5/2	1/2	3	3		1/7	2	120	-132	0	
5/2	1/2	3	2	5/2	1/2	3	2		1/7	2	0	308	0	
5/2	1/2	3	1	5/2	1/2	3	1		1/7	2	-72	-44	0	
5/2	1/2	3	0+	5/2	1/2	3	0+		1/7	2	-96	-264	0	
7/2	1/2	4	4	7/2	1/2	4	4		-1	-1.5	140	-252	4	
7/2	1/2	4	3	7/2	1/2	4	3		-1	-1.5	35	378	-17	
7/2	1/2	4	2	7/2	1/2	4	2		-1	-1.5	-40	198	22	
7/2	1/2	4	1	7/2	1/2	4	1		-1	-1.5	-85	-162	1	
7/2	1/2	4	0-	7/2	1/2	4	0-		-1	-1.5	-100	-324	-20	
7/2	1/2	3	3	7/2	1/2	3	3		-1/7	-1.5	125	-162	1	
7/2	1/2	3	2	7/2	1/2	3	2		-1/7	-1.5	0	378	-6	
7/2	1/2	3	1	7/2	1/2	3	1		-1/7	-1.5	-75	-54	15	
7/2	1/2	3	0+	7/2	1/2	3	0+		-1/7	-1.5	-100	-324	-20	
5/2	1/2	3	3	7/2	1/2	3	3		4/7		-10	60	-2	3
5/2	1/2	3	2	7/2	1/2	3	2		4/7		0	-140	12	3
5/2	1/2	3	1	7/2	1/2	3	1		4/7		6	20	-30	3
5/2	1/2	3	0+	7/2	1/2	3	0+		4/7		8	120	40	3
5/2	1/2	2	2	5/2	1/2	3	2				24	-88	0	5
5/2	1/2	2	1	5/2	1/2	3	1				24	220	0	2
5/2	1/2	2	2	7/2	1/2	4	2				4	-108	8	5
5/2	1/2	2	1	7/2	1/2	4	1				4	18	-16	10
5/2	1/2	2	0-	7/2	1/2	4	0-				8	120	40	3
5/2	1/2	2	2	7/2	1/2	3	2				12	-44	0	15
5/2	1/2	2	1	7/2	1/2	3	1				12	110	0	6
5/2	1/2	3	3	7/2	1/2	4	3				10	-60	2	21
5/2	1/2	3	2	7/2	1/2	4	2				40	180	-28	1
5/2	1/2	3	1	7/2	1/2	4	1				10	108	14	5
7/2	1/2	4	3	7/2	1/2	3	3				15	-90	3	7
7/2	1/2	4	2	7/2	1/2	3	2				20	90	-14	3
7/2	1/2	4	1	7/2	1/2	3	1				5	54	7	15

References

[1] E. U. Condon and G. H. Shortley.
The Theory of Atomic Spectra.
University Press, Cambridge, England, 1963.

[2] R. L. DeKock and W. Weltner, Jr.
Spectroscopy of Rare Earth Oxide Molecules in Inert Matrices at 4 K.
J. Phys. Chem. 75:514, 1971.

[3] M. Dulick.
Spectroscopic Studies of the Diatomic Rare-Earth Oxides.
PhD thesis, M. I. T., 1982.

[4] M. Dulick, R. W. Field and J. Cl. Beaufils.
The Electronic Spectrum of PrO. Energy Linkages, Rotational Analysis, and Hyperfine Structure for Systems XVII and XXI.
J. Mol. Spec. 87:278, 1981.

[5] A. R. Edmonds.
Angular Momentum in Quantum Mechanics.
Princeton University Press, Princeton, New Jersey, 1974.
page 76, eq. 5.4.6.

[6] R. W. Field.
Diatomic Molecule Electronic Structure beyond Simple Molecular Constants.
Ber. Bunsenges. Phys. Chem. 86:771-779, 1982.

[7] S. D. Gabelnick, G. T. Reedy and M. G. Chasanov.
Infrared Spectra and Structrue of some Matrix-Isolated Lanthanide and Actinide Oxides.
J. Chem. Phys. 60:1167, 1974.

[8] F. Herman and S. Skillman.
Atomic Structure Calculations.
Prentice Hall, Englewood Cliffs, N.J, 1963.

[9] K. P. Huber and G. Herzberg.
Molecular Spectra and Molecular Structure, IV, Constants of Diatomic Molecules.
Van Nostrand Reinhold Co., New York, 1979.

[10] C. Linton, M. Dulick, R. W. Field, P. Carette and R. F. Barrow.
Low Lying Electronic States of CeO.
J. Chem. Phys. 74:189, 1981.

[11] W. C. Martin, R. Zalubas and L. Hagen.
Atomic Energy Levels-The Rare-Earth Elements.
Technical Report NSRDS-NBS 60, Institute for Basic Standards.
National Bureau of Standards, 1978.

[12] Michael Dulick, private communication.

[13] M. Rotenberg, N. Metropolis, R. Bivins and J. K. Woolen, Jr.
The 3-j and 6-j Symbols.
The Technology Press, MIT, Cambridge, 1981.

[14] B. L. Silver.
Irreducible Tensor Methods.
Academic Press, New York, 1976.

[15] B. L. Silver.
Irreducible Tensor Methods.
Academic Press, New York, 1976.
page 100.

[16] B. L. Silver.
Irreducible Tensor Methods.
Academic Press, New York, 1976.
page 69, equation 6.7.

[17] B. L. Silver.
Irreducible Tensor Methods.
Academic Press, New York, 1976.
page 92.

[18] B. G. Wybourne.
Spectroscopic Properties of Rare-Earths.
John Wiley and Sons, Inc., New York, 1965.

[19] B. R. Yadav, S. B. Rai and D. K. Rai.
High Resolution Spectrum of GdO.
Can. J. Phys. 54:2429, 1976.

Chapter 6
Europium Oxide

6.1 Introduction

The Ligand Field Model applied to EuO predicts a ground state configuration of Eu^{2+} ($4f^7$) O^{2-}($\pi^4\sigma^2$) (see Chapter 5). The lowest energy molecular state which can be made from this configuration is an $^8\Sigma^-$ state. This state should have very small hyperfine splittings (nuclear spin = 5/2) because there will not be any Fermi-contact splitting and the magnetic effects from the f electrons will cancel since they are in a half-filled shell. The strongest EuO transition in the Rhodamine 6G region was analyzed to test the predictions of the theory.

6.2 Observed Spectra

6.2.1 Fluorescence Spectrum

A transition was excited with a laser at 16735 cm^{-1}. The fluorescence spectrum contained lines to the red of the laser with a P-R separation the same as the transition excited with the laser, hereafter called the laser transition, at 15377, 14709 and 13391 cm^{-1}. This leads to vibrational separations of 2×679, 668 and 2×659. A fit to these data gives ω_e' = 688.8±0.4 cm^{-1} and $\omega_e x_e'$ = 3.3±0.8 cm^{-1}, in good agreement with previous work [8] for the ground state vibrational frequency.

6.2.2 High Resolution

A transition was excited with a single mode dye laser using Rhodamine 6G dye. The excitation was monitored by a monochromator set to pass the P branch for R branch excitation and set to pass the R branch for P branch excitation. The spectrum is red degraded. The most prominent feature in both the P and R branches (no Q branch was observed) is a group of 4 lines whose relative separation does not change with N. The group of four lines roughly follows the pattern expected for levels with energy predicted by BN(N+1). The combination differences obtained from these lines, listed in Table 6-VI, show that the lower state is unperturbed, as opposed to the situation for the upper state in which level shifts of up to 1 cm^{-1} are observed. For several values of N, some or all of these strongest four lines disappear. A perturbation which split the six hyperfine components beyond the Doppler width could cause this effect since the intensity of the line would be reduced by six. In addition to the strongest four lines, for each N there are many broader, weaker lines in which the hyperfine splitting of the lines causes them to broaden beyond the 0.03 cm^{-1} Doppler width. In several of these subbranches the six hyperfine components are resolved. As with the perturbed stronger lines, hyperfine splitting could reduce the intensity of these subbranches by a factor of six relative to the stronger, unsplit lines. In addition, the monitor wavelength was peaked to optimize the fluorescence signal from the strongest lines. Since the fluorescence from the broader subbands could be up to 13 cm^{-1} away from the stronger lines for a given excitation region, their already small signals were reduced to below the noise.

It proved impossible to track any of the subbands below $N'' = 11$. At

low N, the hyperfine interactions of the spin sub-states causes splittings in all the lines. With Doppler limited resolution, the spectrum becomes a quasi-continuum.

One of the weaker subbands was successfully tracked by using the following tedious experimental technique. First the oven current was raised by about 15% which resulted in much shorter runs but an improvement in the signal to noise by a factor of about five. Then the spectrum of a line with resolved hyperfine structure was recorded, say a P line. The monochromator was peaked on the corresponding R line fluorescence, after which the laser was tuned until the signal from scattered laser light through the monochromator was maximized. This placed the laser at the same frequency as the desired R line. After noting the frequency the was laser tuned back to the P line. Next the frequency of the monochromator was adjusted until it was centered at the laser frequency. Finally the laser was returned to the R line region and a 0.6 cm^{-1} wide laser excitation scan was recorded. The R line was usually found within 0.1 cm^{-1} of the expected frequency. This method of recording the spectrum automatically performs some of the data analysis since, for each line, the lower state P-R separation is known to within 0.01 cm^{-1}. It should also be noted that this technique would be almost impossible to apply without the scanning capabilities of the Coherent 699-29 computerized dye laser. Even with the power this method provides, I was unable to track the subband below a perturbation at N" = 36 or above one at N" = 57,; in fact, the lines near those values of N are somewhat suspect since it is extremely easy to start tracking a different subband by accident. This occurs because all subbands have very similar lower state combination differences.

One important feature of the lines obtained by this method is that the

hyperfine structure for lines with the same upper state but different lower states is exactly the same, while the opposite is not true. This shows that changes in hyperfine width are taking place only in the upper state. However, the changes appear to be due to perturbations of the upper state. As I will show later in this chapter, the hyperfine splitting should not change significantly with N for the range of N's examined. Thus, it is not possible to state whether the splitting resides in only the upper or lower state. The subband tracked with this method has been identified as the F_7, J=N-5/2, e lines from the 151 isotope. I will explain the name and assignment after deriving and discussing the $^8\Sigma$ Hamiltonian.

6.3 $^8\Sigma$ Hamiltonian

This section works out the Hamiltonian for a Hund's case b $^8\Sigma$ state in detail. In a case b molecule the useful angular momenta are

L, the electron orbital angular momentum, described in the molecular coordinate system.

R, the rotational angular momentum of the molecule.

N, where $\mathbf{N} = \mathbf{R} + \mathbf{L}$.

S, the sum of the spins of each electron, described in the lab fixed coordinates.

J, where $\mathbf{J} = \mathbf{N} + \mathbf{S}$

I, the angular momentum due to the spin of the nuclei, described in the lab fixed coordinates.

F, where $\mathbf{F} = \mathbf{I} + \mathbf{J}$.

6.3.1 Rotational Hamiltonian

The rotational Hamiltonian is given by $B\mathbf{R}^2$ [4].

$$\mathbf{N} = \mathbf{R} + \mathbf{L}$$
$$\mathbf{R} = \mathbf{N} - \mathbf{L}$$
$$B\mathbf{R}^2 = B[\mathbf{N}^2 + \mathbf{L}^2 - \mathbf{N}\cdot\mathbf{L} - \mathbf{L}\cdot\mathbf{N}].$$

Since **N** operates on the rotational angular momentum of the nuclei while **L** operates on the orbital angular momentum of the electrons, **N** and **L** commute and the equation becomes

$$B\mathbf{R}^2 = B[\mathbf{N}^2 + \mathbf{L}^2 - 2\mathbf{N}\cdot\mathbf{L}].$$

The diagonal elements will be

$BN(N+1)$ from $B\mathbf{N}^2$

$BL(L+1)$ from $B\mathbf{L}^2$.

The dot product will also give a diagonal term which includes Λ, the projection of **L** on the internuclear axis. Since only Σ states will be considered in this derivation of the Hamiltonian, $\Lambda = 0$, so the diagonal element of the dot product term will be 0. Usually **L** is not known in a diatomic molecule, so the term $BL(L+1)$ is commonly included in the electronic energy of the state.

Consequently, the diagonal element of the rotational Hamiltonian is

$$\langle \eta\Lambda NSJ|H_r|\eta\Lambda NSJ\rangle = BN(N+1). \qquad (6.1)$$

6.3.2 Centrifugal Distortion

Centrifugal Distortion can be approximated as a second order perturbation correction which involves matrix elements of the operator B(r) between states with different vibrational quantum numbers. Using second order perturbation theory, the correction to the energy of state n, E_n, is written

$$E_n^{(2)} = \Sigma_{n',\, n' \neq n} \; |\langle v_n; e_n | B(r) \mathbf{R}^2 | v_{n'}; e_{n'} \rangle|^2 / (E_n^{(0)} - E_{n'}^{(0)}).$$

where the vibrational wavefunctions are abbreviated v_n and the electronic e_n. $E_n^{(0)}$ are the unperturbed energies of state n. B(r) operates only on the vibrational wavefunctions while $\mathbf{R}^2$ operates on the electronic. Thus the second order correction can be written as

$$E_n^{(2)} = \Sigma_{n',\, n' \neq n} \; |\langle v_n | B(r) | v_{n'} \rangle|^2 |\langle e_n | \mathbf{R}^2 | e_{n'} \rangle|^2 / (E_n^{(0)} - E_{n'}^{(0)}).$$

The non-zero matrix elements of $\mathbf{R}^2$ of interest are

$$\langle {}^8\Sigma^- | \; \mathbf{R}^2 \; | {}^8\Sigma^- \rangle$$

and

$$\langle {}^8\Sigma^- | \; \mathbf{R}^2 \; | {}^8\Pi \rangle.$$

As I will discuss later in this chapter, both the lower and upper ${}^8\Sigma^-$ states involved in the transition appear to be well isolated from any ${}^8\Pi$ states. The energy difference in the denominator will make the ${}^8\Pi$ term insignificant. Thus the second order correction is reduced to

$$E_n^{(2)} = \Sigma_{n',\, n' \neq n} \; |\langle v_n|B(r)|v_{n'}\rangle|^2 \; |\langle {}^8\Sigma^-| \; \mathbf{R}^2 \; |{}^8\Sigma^-\rangle|^2 \; /(E_n^{(0)} - E_{n'}^{(0)}).$$

The centrifugal distortion constant is defined as

$$D = -\Sigma_{n',\, n' \neq n} \; |\langle v_n|B(r)|v_{n'}\rangle|^2 \; /(E_n^{(0)} - E_{n'}^{(0)}).$$

so the energy correction becomes

$$E_n^{(2)} = -D|\langle {}^8\Sigma^-| \; \mathbf{R}^2 \; |{}^8\Sigma^-\rangle|^2.$$

The matrix element $\langle {}^8\Sigma^-| \; \mathbf{R}^2 \; |{}^8\Sigma^-\rangle$ has already been calculated. Using the results of equation (6.1), the matrix element of the centrigufugal distortion constant becomes

$$\langle \eta \Lambda N S J|H_{cd}|\eta \Lambda N S J\rangle = -D[N(N+1)]^2 \qquad (6.2)$$

6.3.3 Spin-Spin Hamiltonian

The Spin-Spin interaction is given by equations 3.55a and 3.88 in Carrington et. al. [4].

$$\langle |H_{ss}| \rangle = -g^2\beta^2 \langle \eta \Lambda N S J| \; \Sigma_{i,j>i}[6]^{1/2} r_{ij}^{-3} \; T^2(\mathbf{s}_i,\mathbf{s}_j)\cdot C^2(\omega_{ij}) \; |\eta' \Lambda' N' S' J'\rangle \qquad (6.3)$$

where

Λ = the projection of **L** on the internuclear axis of **L**

N, **J** and **S** are as previously defined

η = all other quantum numbers

r_{ij} = the distance between electron i and electron j

$\mathbf{s}_i$ = spin of electron i

$T^2(\mathbf{s}_i,\mathbf{s}_j)$ is the second rank tensor formed from

$T^1(\mathbf{s}_i)$ and $T^1(\mathbf{s}_j)$

ω_{ij} = the angular coordinates of electrons i and j

and

$C^2(\omega_{ij})$ = Racah's normalized spherical harmonics where

$$C^k_q(\theta,\phi) = [4\pi/(2k+1)]^{1/2}\ Y^k_m(\theta,\phi). \quad [12]$$

The dot product $T^2 \cdot C^2$ must be handled with care since $T^2(\mathbf{s}_i,\mathbf{s}_j)$ is defined in terms of space fixed coordinates while $C^2(\omega_{ij})$ is defined relative to molecule fixed coordinates. Following the prescription of Brown and Howard [3]

$$T^2(\mathbf{s}_i,\mathbf{s}_j)\cdot C^2(\omega_{ij}) = \Sigma_p\ T^2_p(\mathbf{s}_i,\mathbf{s}_j)C^2_{-p}(\omega_{ij}) \tag{6.4}$$

$$C^2_{-p} = \Sigma_q\ C^2_q D^{2\ *}_{-pq}, \tag{6.5}$$

where p and q refer to space and molecule coordinates respectively and $D^{2\ *}_{-pq}$ is the maxtrix element of the rotation operator between the space and molecule coordinates. Therefore

$$T^2(\mathbf{s}_i,\mathbf{s}_j)\cdot C^2(\omega_{ij}) = T^2(\mathbf{s}_i,\mathbf{s}_j)\cdot(C^2_q(\omega_{ij})D^{2\ *}_{.q}). \tag{6.6}$$

Notice that when writing a dot product, the projection of the angular momentum that is the summation index for the dot product (see equation (6.4) for example) is not written. Since the dot product in equation (6.6) involves the space projections of the operators, the p in $D^{2\ *}_{pq}$ is replaced by a dot. Using the Wigner-Eckart theorem on the above dot product for the

matrix element of H_{ss} [13] yields

$$<|H_{ss}|> = \delta_{JJ'}\ \delta_{MM'}\ -g^2\beta^2\Sigma_{i,j>i}\ (-1)^{N'+S+J} \begin{Bmatrix} N & S & J \\ S' & N' & 2 \end{Bmatrix}$$

$$\times\ <S||\ T^2(\mathbf{s}_i,\mathbf{s}_j)\ ||S'>\Sigma_q\ [6]^{1/2}r_{ij}^{-3}<\eta\Lambda N||\ C_q^2(\omega_{ij})D^{2\ *}_{.q}\ ||\eta'\Lambda'N'>. \quad (6.7)$$

$C_q^2(\omega_{ij})$ operates only on the electrons and thus affects only the quantum numbers L and Λ. $D^{2\ *}_{.q}(\omega)$ operates on the Euler angles of the molecule and thus affects the quantum number N and its projection, Λ. I can write

$$<\eta\Lambda N||C_q^2(\omega_{ij})D^{2\ *}_{.q}||\eta'\Lambda'N'> = <\eta\Lambda|C_q^2(\omega_{ij})|\eta'\Lambda'><\Lambda N||D^{2\ *}_{.q}||\Lambda'N'>. \quad (6.8)$$

I need to find an expression for $<\Lambda N||\ D^{2\ *}_{.q}\ ||\Lambda'N'>$. From Carrington [5]

$$<J'M'\Omega'|\ D^{k\ *}_{-pq}\ |JM\Omega> = [(2J+1)(2J'+1)]^{1/2}\ \ 1/(8\pi^2)$$

$$\times \int D^{J'}_{M'\Omega'}\ D^{k\ *}_{-pq}\ \ D^{J\ *}_{M\Omega}d\omega.$$

Edmonds gives [7]

$$D^{J\ *}_{M'M} = (-1)^{M'-M}D^{J}_{-M'-M}.$$

Therefore

$$<J'M'\Omega'||\ D^{k\ *}_{-pq}\ ||JM\Omega> = [(2J+1)(2J'+1)]^{1/2}\ \ 1/(8\pi^2)(-1)^{-p-q+M-\Omega}$$

$$\times \int D^{J'}_{M'\Omega'}\ \cdot D^{k}_{p-q}\ D^{J}_{-M\Omega}d\omega,$$

which becomes

$$= [(2J+1)(2J'+1)]^{1/2}\ (-1)^{-p-q+M-\Omega} \begin{pmatrix} J' & k & J \\ M' & p & -M \end{pmatrix} \begin{pmatrix} J' & k & J \\ \Omega' & -q & -\Omega \end{pmatrix} .$$

The 3-j symbols will be zero unless $M'+p-M = 0 \Rightarrow M = M'+p$, and $\Omega'-q-\Omega = 0 \Rightarrow \Omega = \Omega'-q$. Putting these results into the exponent of the (-1) term in the above equation gives

$$\langle J'M'\Omega'||\ D^{k}_{-pq}{}^{*}\ ||JM\Omega\rangle = [(2J+1)(2J'+1)]^{1/2}\ (-1)^{M'-\Omega'} \times \begin{pmatrix} J' & k & J \\ -M' & -p & M \end{pmatrix} \begin{pmatrix} J' & k & J \\ -\Omega' & q & \Omega \end{pmatrix} . \tag{6.9}$$

I can also derive an expression for this matrix element using the Wigner-Eckart theorem.

$$\langle J'M'\Omega'|\ D_{-pq}{}^{k}{}^{*}\ |JM\Omega\rangle = (-1)^{J'-M'}\ \langle J'\Omega'||\ D^{k}_{.q}{}^{*}\ ||J\Omega\rangle \begin{pmatrix} J' & k & J \\ -M' & -p & M \end{pmatrix} \tag{6.10}$$

Comparing equations (6.9) and (6.10) gives

$$\langle J'\Omega'||\ D^{k}_{.q}{}^{*}\ ||J\Omega\rangle = (-1)^{\Omega'-J'}\ [(2J+1)(2J'+1)]^{1/2} \begin{pmatrix} J' & k & J \\ -\Omega' & q & \Omega \end{pmatrix} . \tag{6.11}$$

Putting this result into equation (6.8)

$$\langle \eta\Lambda N||\ C^{2}_{q}(\omega_{ij}) D^{2}_{.q}{}^{*}\ ||\eta'\Lambda'N'\rangle = (-1)^{N-\Lambda}\ \langle \eta\Lambda|\ C^{2}_{q}(\omega_{ij})\ |\eta'\Lambda'\rangle$$

$$[(2N+1)(2N'+1)]^{1/2} \begin{pmatrix} N & 2 & N' \\ -\Lambda & q & \Lambda' \end{pmatrix}. \tag{6.12}$$

Along with equation (6.7) this gives

$$\begin{aligned} \langle|H_{ss}|\rangle = (-1)^{N'+S+J+N-\Lambda} & [(2N+1)(2N'+1)]^{1/2} [6]^{1/2} \begin{Bmatrix} N & S & J \\ S' & N' & 2 \end{Bmatrix} \\ & \times \Sigma_q \begin{pmatrix} N & 2 & N' \\ -\Lambda & q & \Lambda' \end{pmatrix} \Sigma_{i,j>i} \langle S|| T^2(\mathbf{s}_i,\mathbf{s}_j) ||S'\rangle \\ & \times -g^2\beta^2 r_{ij}^{-3} \langle \eta\Lambda| C_q^2(\omega_{ij}) |\eta'\Lambda'\rangle. \end{aligned} \tag{6.13}$$

I am considering only an isolated $^8\Sigma$ state, so $S = S'$ and $\Lambda = \Lambda' = 0 \Rightarrow q = 0$. In addition, the terms in the summation over i and j depend only on S, η, η' and Λ. Since I am calculating the matrix elements for an unperturbed $^8\Sigma$ state, these quantum numbers will be constants. They will be replaced with the constant λ', such that

$$\begin{aligned} \lambda' = & -g^2\beta^2 r_{ij}^{-3} [6]^{1/2} \Sigma_{i,i>j} \langle S|| T^2(\mathbf{s}_i,\mathbf{s}_j) ||S'\rangle \\ & \times \langle \eta\Lambda| C_q^2(\omega_{ij}) |\eta'\Lambda'\rangle. \end{aligned} \tag{6.14}$$

Equation (6.13) becomes

$$\begin{aligned} \langle|H_{ss}|\rangle = (-1)^{N'+S+J+N} & [(2N+1)(2N'+1)]^{1/2} \begin{Bmatrix} N & S & J \\ S & N' & 2 \end{Bmatrix} \\ & \times \Sigma_q \begin{pmatrix} N & 2 & N' \\ 0 & 0 & 0 \end{pmatrix} \lambda'. \end{aligned} \tag{6.15}$$

The 3-j symbol

$$\begin{pmatrix} N & 2 & N' \\ 0 & 0 & 0 \end{pmatrix} = 0$$

unless $N + 2 + N'$ is even $\Rightarrow N' = N$ or $N' = N \pm 2$.

All the following 3-j and 6-j symbols are evaluated using the formulas from Edmonds [6].

$\underline{N' = N}$

$$\begin{pmatrix} N & 2 & N \\ 0 & 0 & 0 \end{pmatrix} = (-1)^{N+1} \left[\frac{N(N+1)}{(2N+3)(2N+1)(2N-1)}\right]^{1/2}$$

$$\begin{Bmatrix} N & S & J \\ S & N' & 2 \end{Bmatrix} = (-1)^{N+J+S} \times \frac{2[3X(X-1) - 4N(N+1)S(S+1)]}{[4N(2N-1)(2N+1)(N+1)(2N+3)4S(2S-1)(2S+1)(S+1)(2S+3)]^{1/2}}$$

where $X = N(N+1) + S(S+1) - J(J+1)$. Putting these results into equation (6.15)

$$<|H_{ss}|> = -\lambda' \frac{X(X-1) - 21N(N+1)}{(2N+3)(2N-1)4[210]^{1/2}} \quad . \tag{6.16}$$

For the eight spin components, $J = N \pm 7/2$, $N \pm 5/2$, $N \pm 3/2$ or $N \pm 1/2$. The expression for $<|H_{ss}|>$ can be simplified by making this substitution. The results are listed below:

J	$<NSJ\|H_{ss}\|NSJ> \times 4[210]^{1/2}$
N + 7/2	$-14\lambda' N/(2N+3)$
N - 7/2	$-14\lambda' (N+1)/(2N-1)$
N + 5/2	$-2\lambda' (N-21)/(2N+3)$
N - 5/2	$-2\lambda' (N+22)/(2N-1)$

N + 3/2	$6\lambda'\ (2N^2+15N-22)/[(2N+3)(2N-1)]$
N - 3/2	$6\lambda'\ (2N^2-11N-35)/[(2N+3)(2N-1)]$
N + 1/2	$6\lambda'\ (2N^2+5N-21)/[(2N+3)(2N-1)]$
N - 1/2	$6\lambda'\ (2N^2-N-24)/[(2N+3)(2N-1)]$

I would like to have these elements in the form

(constant)·λ' ·(constant + function of N).

For the term in N + 7/2, this is equivalent to solving the equation

$$a + \frac{b}{2N+3} = \frac{N}{2N+3}$$

for a and b. The solution is

$$-14\lambda'\ N/(2N+3) = -7\lambda'[1 - 3/(N+3)].$$

I will tabulate the results for the others with one additional change. Just to simplify matters, I will absorb the $4[210]^{1/2}$ into the constant λ such that $\lambda = \lambda'/4[210]^{1/2}$.

Table 6-I: Diagonal Elements of the Spin-Spin Hamiltonian

J	$\langle NSJ\|H_{ss}\|NSJ\rangle$
N + 7/2	$-7\lambda[1 - 3/(2N+3)]$
N - 7/2	$-7\lambda[1 + 3/(2N+1)]$
N + 5/2	$-\lambda[1 - 45/(2N+3)]$
N - 5/2	$-\lambda[1 + 45/(2N+1)]$
N + 3/2	$3\lambda[1 + 20/(2N+3) - 7/(2N-1)]$
N - 3/2	$3\lambda[1 + 7/(2N+3) - 20/(2N-1)]$
N + 1/2	$5\lambda[1 + 12/(2N+3) - 9/(2N-1)]$
N - 1/2	$5\lambda[1 + 9/(2N+3) - 12/(2N-1)]$

Now for the off-diagonal elements. The task is somewhat simplified because, since the Hamiltonian is a Hermitian operator, I only have to find the

matrix elements for one half of the matrix.

Elements with N' = N-2

$$\begin{pmatrix} N & 2 & N-2 \\ 0 & 0 & 0 \end{pmatrix} = (-1)^N \left[\frac{3N(N-1)}{2(2N+1)(2N-1)(2N+3)}\right]^{1/2}$$

$$\begin{Bmatrix} N & S & J \\ S & N-2 & 2 \end{Bmatrix} = (-1)^M \times \left[\frac{6M(M+1)(M-2J-1)(M-2J)(M-2S-1)(M-2S)(M-2N+1)(M-2N+2)}{4S(2S-1)(2S+1)(S+1)(2S+3)4N(2N-3)(N-1)(2N-1)(2N+1)}\right]^{1/2} \quad (6.17)$$

where M = N+J+S. Therefore

$$\langle NSJ|H_{ss}|N\text{-}2\ SJ\rangle = (-1)^{N-2+S+J+N}\ [(2N+1)(2N-3)]^{1/2}\ \lambda \times \begin{Bmatrix} N & S & J \\ S & N-2 & 2 \end{Bmatrix} \begin{pmatrix} N & 2 & N-2 \\ 0 & 0 & 0 \end{pmatrix}$$

$$= (1/2)\ [(J+N+7/2)\ (J+N+9/2)\ (-J+N+5/2)\ (-J+N+7/2)$$

$$\times\ (J+N-9/2)\ (J+N-7/2)\ (J-N+9/2)\ (J-N+11/2)]\ /\ [(2N-1)^2$$

$$\times\ (2N-3)\ (2N+1)]^{1/2}.$$

As mentioned above, $\langle N\text{-}2\ SJ|H_{ss}|NSJ\rangle = \langle NSJ|H_{ss}|N\text{-}2\ SJ\rangle$.

6.3.4 Spin-Rotation Hamiltonian

$$H_{sr} = \gamma \mathbf{R}\cdot\mathbf{S} = \gamma(\mathbf{N}-\mathbf{L})\cdot\mathbf{S} = \gamma T^1(\mathbf{N})\cdot T^1(\mathbf{S}) - \gamma T^1(\mathbf{L})\cdot T^1(\mathbf{S})$$

The matrix element of $\gamma T^1(\mathbf{L})\cdot T^1(\mathbf{S})$ will have a term of the following form:

$$<\eta\Lambda|T^1(\mathbf{L})|\eta'\Lambda'>.$$

For $\Lambda = \Lambda' = 0$, this term will be zero. I am not considering any off-diagonal elements with $\Delta\Lambda = \pm 1$. Thus the contribution of $\gamma T^1(\mathbf{L})\cdot T^1(\mathbf{S})$ to the matrix element

$$<\eta\Lambda NSJ|H_{sr}|\eta'\Lambda'N'S'J'>$$

is zero. Working through the other term,

$$<\eta\Lambda NSJ|\ \gamma T^1(\mathbf{N})\cdot T^1(\mathbf{S})\ |\eta'\Lambda'N'S'J'> = (-1)^{N'+S+J}\ \delta_{JJ'}$$

$$\times\ <\eta N||\ T^1(\mathbf{N})\ ||\eta'N'><S||\ \gamma T^1(\mathbf{S})\ ||S'> \begin{Bmatrix} N & S & J \\ S' & N' & 1 \end{Bmatrix}$$

$$<\eta N||\ T^1(\mathbf{N})\ ||\eta'N'> = [N(N+1)(2N+1)]^{1/2}\ \delta_{\eta\eta'}\ \delta_{NN'}$$

$$<S||\ \gamma T^1(\mathbf{S})\ ||S'> = [S(S+1)(2S+1)]^{1/2}\ \delta_{SS'}$$

$$\begin{Bmatrix} N & S & J \\ S & N & 1 \end{Bmatrix} = (-1)^{N+S+J+1}\ \frac{N(N+1) + S(S+1) - J(J+1)}{2[N(2N+1)(N+1)S(2S+1)(S+1)]^{1/2}}$$

Putting it all together

$$<\eta\Lambda NSJ|\ H_{sr}\ |\eta'\Lambda'N'S'J'> = (1/2)\gamma\ [(J(J+1) - N(N+1)$$

$$- S(S+1)]\delta_{NN'}\delta_{SS'}\delta_{\eta\eta'}.$$

As for the rotational Hamiltonian, I have ignored all terms off-diagonal in Λ.

Table 6-II: Diagonal Elements of the Spin-Rotation Hamiltonian

J	$\langle\eta\Lambda NSJ\| H_{sr} \|\eta'\Lambda'N'S'J'\rangle$
N + 7/2	$(7/2)\gamma$ N
N - 7/2	$-(7/2)\gamma(N+1)$
N + 5/2	$(1/2)\gamma(5N-7)$
N - 5/2	$-(1/2)\gamma(5N-12)$
N + 3/2	$(3/2)\gamma(3N-12)$
N - 3/2	$-(3/2)\gamma(3N+15)$
N + 1/2	$(1/2)\gamma(N-15)$
N - 1/2	$-(1/2)\gamma(N+16)$

6.3.5 Spin-Orbit

The diagonal spin orbit elements will be zero in a Σ state. However I am interested in the contributions of off-diagonal spin-orbit to spin-spin and spin-rotation. Consequently when I calculate the spin-orbit matrix elements, at each step I will bunch all constants into a term called "Υ".

I will use the form of the spin-orbit Hamiltonian given by Hougen [10].

$$H_{so} = T^1(\mathbf{L})\cdot T^1(\mathbf{S})$$

Evaluating the matrix element in exactly the same way as for the spin-spin operator gives

$$\langle\eta\Lambda NSJ|H_{so}|\eta'\Lambda'N'S'J'\rangle = (-1)^{N+N'+S+J-\Lambda}\ \Upsilon\ [(2N+1)(2N'+1)]^{1/2} \begin{Bmatrix} N & S & J \\ S' & N' & 1 \end{Bmatrix} \begin{pmatrix} N & 1 & N' \\ -\Lambda & q & \Lambda' \end{pmatrix}.$$

For $N = N'$ but $\Lambda' = \Lambda+1 = \Pi$ state, this equation becomes

$$<\eta\Lambda NSJ|H_{so}|\eta'\Lambda'N'S'J'> = (-1)^{N+N'+S+J-\Lambda} \Upsilon X/[N(N+1)]^{1/2}$$

where $X = N(N+1) + S(S+1) - J(J+1)$. Second order perturbation theory states that the contribution of this off-diagonal term to the diagonal matrix element will be [1]

$$|<|H_{so}|>|^2/(E_\Sigma - E_\Pi)$$

$$= [\Upsilon X]^2/[N(N+1)(E_\Sigma - E_\Pi)]. \qquad (6.18)$$

where Υ contains the strength of the spin-orbit interaction. Comparing equation (6.18) to equation (6.16), you can see that the terms have a very similar funtional form.

Next I will derive the matrix elements of H_{so} for $N' = N+1$ and $\Lambda' = \Lambda+1 = \Pi$. This time K will be my all purpose constant.

$$<\eta\Lambda NSJ|H_{so}|\eta\Lambda+1N+1SJ'> = (-1)^J$$

$$\times K [[(M+1)(M-2J)(M-2N)(M-2S+1)]/[N(2N+1)^2(2N+3)]]^{1/2} \qquad (6.19)$$

where $M = N+J+S$. From second order perturbation theory the matrix element in equation (6.19) will contribute to matrix elements with $N' = N+2$ in the following manner:

$$<N|H_{pert}|N+2> = <N|H_{pert}|N+1><N+1|H_{pert}|N+2>/\Delta E. \qquad (6.20)$$

Using equations (6.19) and (6.20) gives the following second order correction to the matrix element off-diagonal by two in N:

$$-K[[(M+1)(M-2J)(M-2N)(M-2S+1)(M+2)(M-2J+1)$$

$$(M-2N+1)(M-2S+2)]/[N(2N+1)(2N+2)(2N+3)(2N+4)] \quad [E_{\Sigma} - E_{\Pi}]]. \quad (6.21)$$

Comparing equations (6.17) and (6.21) shows that the perturbation due to spin-orbit has essentially the same form as the off-diagonal element of the spin-spin operator.

Since both the diagonal and off-diagonal elements of the spin-spin operator have essentially the same functional form as the perturbation due to spin-orbit, the spin-spin constant may include a significant contribution from spin-orbit.

6.3.6 Hyperfine Hamiltonian

$$H_{hfs} = aT^1(I)\cdot T^1(L) + \quad bT^1(I)\cdot T^1(S) + \quad cT^1(I)\cdot T^1(S,C^2) \ [4]$$

I am only considering $\Lambda = 0$, $\Delta\Lambda = 0$ matrix elements. Given these conditions the first term is zero. The second term is due to the Fermi contact interaction, while the third comes from the electron-nuclear dipolar interaction.

$$\langle \eta\Lambda NSJIF|\ bT^1(I)\cdot T^1(S)|\ \eta'\Lambda'N'S'J'I'F'\rangle \quad = \delta_{FF'}\ (-1)^{J'+I+F} b$$

$$\times\ \langle \eta NSJ||\ T^1(S)\ ||\eta'N'S'J'\rangle\ \ \langle I||\ T^1(I)\ ||I'\rangle \quad \begin{Bmatrix} J & J' & 1 \\ I & I & F \end{Bmatrix} \quad (6.22)$$

$$\langle I||\ T^1(I)\ ||I'\rangle = [I(I+1)(2I+1)]^{1/2}\ \delta_{II'}$$

$$\langle \eta NSJ \| \, T^1(\mathbf{S}) \, \| \eta' N'S'J' \rangle = \delta_{NN'} \; (-1)^{N+S+J'+1} \; [(2J+1)(2J'+1)]^{1/2}$$

$$\times \langle S \| \, T^1(\mathbf{S}) \, \| S' \rangle \quad \begin{Bmatrix} J & J' & 1 \\ S & S' & N \end{Bmatrix} \; \delta_{\eta\eta'}$$

$$\langle S \| \, T^1(\mathbf{S}) \, \| S' \rangle = [S(S+1)(2S+1)]^{1/2} \; \delta_{SS'}$$

Putting all the above equations together

$$\langle \eta \Lambda NSJIF | \; bT^1(\mathbf{I}) \cdot T^1(\mathbf{S}) | \; \eta' \Lambda' N'S'J'I'F' \rangle \quad = b(-1)^{2J'+I+F+N+S+1}$$

$$[I(I+1)(2I+1)S(S+1)(2S+1)(2J+1)(2J'+1)]^{1/2} \quad \begin{Bmatrix} F & I & J \\ 1 & J & I \end{Bmatrix}$$

$$\begin{Bmatrix} N & S & J \\ 1 & J & S \end{Bmatrix} \quad \delta_{\eta\eta'} \; \delta_{\Lambda\Lambda'} \; \delta_{SS'} \; \delta_{II'} \quad \delta_{FF'}. \tag{6.23}$$

There is no reason to work through the algebra for the off-diagonal elements because the result will be too complex to provide insight into the spectrum. However it is useful to derive the explicit formulas for the diagonal matrix elements in order to develop a feel for the size of the hyperfine splitting. In addition to the diagonal elements, there will be terms off-diagonal in J with $J' = J \pm 1$.

$$\begin{Bmatrix} F & I & J \\ 1 & J & I \end{Bmatrix} = (-1)^{F+I+J} \; \frac{F(F+1) - I(I+1) - J(J+1)}{2[I(I+1)(2I+1)J(J+1)(2J+1)]^{1/2}}$$

$$\begin{Bmatrix} N & S & J \\ 1 & J & S \end{Bmatrix} = (-1)^{N+S+J+1} \; \frac{J(J+1) + S(S+1) - N(N+1)}{2[S(S+1)(2S+1)J(J+1)(2J+1)]^{1/2}}$$

Putting these results into equation (6.23)

$$\langle \eta\Lambda NSJIF|\ bT^1(\mathbf{I})\cdot T^1(\mathbf{S})|\ \eta\Lambda NSJIF\rangle \quad = (-1)^{F+I+J} b$$

$$\times \frac{[F(F+1) - I(I+1) - J(J+1)]\ [J(J+1) + S(S+1) - N(N+1)]}{2J(J+1)}. \qquad (6.24)$$

Last are the matrix elements of the electron-nuclear dipole interaction:-

$$\langle \eta\Lambda NSJIF|\ c'T^1(\mathbf{I})\cdot T^1(\mathbf{S},C^2)\ \ |\eta'\Lambda'N'S'J'I'F'\rangle \quad = \delta_{FF'}\ (-1)^{J'+I+F}$$

$$\times\ c'\langle I||\ T^1(\mathbf{I})\ ||I'\rangle\ \ \langle \eta\Lambda NSJ||\ T^1(\mathbf{S},C^2)\ ||\eta'\Lambda'N'S'J'\rangle$$

$$\times \begin{Bmatrix} J & I & F \\ I' & J' & 1 \end{Bmatrix}.$$

As was the case with the spin-spin operator, care must be taken with $T^1(\mathbf{S},C^2)$ because the operator $\mathbf{S}$ is defined in terms of the lab fixed coordinates while C^2 is defined with molecular coordinates. Following the procedure used in section 6.3.3,

$$\langle \eta\Lambda NSJ||\ T^1(\mathbf{S},C^2)\ ||\eta'\Lambda'N'S'J'\rangle =$$

$$\langle \eta\Lambda NSJ||\ \Sigma_q T^1(\mathbf{S},C^2 D^{2\ *}_{.q})\ \ ||\eta'\Lambda'N'S'J'\rangle$$

$$= \langle \eta\Lambda N||\ \Sigma_q\ C^2 D^{2\ *}_{.q}\ \ ||\eta'\Lambda'N'\rangle\ \times$$

$$\langle S||\ T^1(\mathbf{S})\ ||S'\rangle\ [5(2J+1)(2J'+1)]^{1/2}\ \times$$

$$\begin{Bmatrix} N & N' & 2 \\ S & S & 1 \\ J & J' & 1 \end{Bmatrix}$$

From equation (6.12),

$$\langle \eta\Lambda N || \ \Sigma_q \ C^2 D^{2\ *}_{.q} \ || \eta'\Lambda'N' \rangle \quad = \quad \Sigma_q \ \langle \eta\Lambda | \ C^2_q(\omega_{ij}) \ | \eta'\Lambda' \rangle \quad (-1)^{N-\Lambda}$$

$$[(2N+1)(2N''+1)]^{1/2} \begin{pmatrix} N & 2 & N' \\ -\Lambda & q & \Lambda' \end{pmatrix} .$$

As before, I am considering the case where $\Lambda = \Lambda' = 0 \Rightarrow q = 0$.

After absorbing the $\langle \eta\Lambda | \ C^2_q(\omega_{ij}) \ | \eta'\Lambda' \rangle$ term into a new constant, c, the matrix elements of the dipole-dipole interaction become

$$\langle \eta\Lambda NSJIF | \ cT^1(\mathbf{I}) \cdot T^1(\mathbf{S}, C^2) \ \ | \eta'\Lambda'N'S'J'I'F' \rangle \quad =$$

$$c[I(I+1)(2I+1)S(S+1)(2S+1)(2N+1)(2N'+1)(2J+1)(2J'+1)]^{1/2}$$

$$\times \ (-1)^{J'+I+F+N} \begin{pmatrix} N & 2 & N' \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} J & J' & 1 \\ I & I & F \end{Bmatrix} \tag{6.25}$$

$$\begin{Bmatrix} N & N' & 2 \\ S & S & 1 \\ J & J' & 1 \end{Bmatrix} .$$

with the restrictions $J' = J, J\pm1$; $N' = N, N\pm2$. Even the diagonal elements do not simplify much, so nothing is gained by working through the algebra.

6.4 Qualitative Description of Expected Energy Levels

With the diagonal elements of the Hamiltonian, it is possible to construct a qualitative description of the rotational energy levels. As in any case b molecule, the pattern forming quantum number is N, so the primary order of the levels is determined by BN(N+1). Each N level, with its eight spin components, is split by the spin-spin and spin-rotation interactions. Table 6-I shows the primary effects of the spin-spin interaction. The levels J =

N±7/2 will be grouped together, as will J = N+5/2 and N-5/2, J = N +3/2 and N-3/2, and J = N+1/2 and N-1/2. The pairs of lines will be separated due to the spin-spin and spin-rotation interactions. The separations between the |J-N| = 7/2, 5/2, 3/2 and 1/2 level paris, for a given N , i.e. Energy(7/2) - Energy(5/2) etc., should have the approximate ratios of 6:4:2. The qualitative energy level pattern expected for a simpler system, a $^4\Sigma^-$ state, is shown in Figure 6-0.

The hyperfine splitting terms have been evaluated for N = 30, 40 and 50, and are listed in Table 6-IV. Note that the coefficients do not change significantly with N. The table clearly shows that the J=N±7/2 components will be split the most. The hyperfine splitting will be discussed in more detail during the analysis section.

6.5 Parity and Naming of the Levels

6.5.1 Parity

The first issue is the absolute parity of the Σ state. The absolute electronic parity in a Hund's case b molecule, derived in Chapter 4, is parity = $(-1)^{L+\Sigma l-\Lambda}$. I am working in the separated atom picture, so I have to calculate the parity for both the Eu^{2+} and the O^{2-} ions. The O^{2-} ion without any electrons outside closed shells forms a ^1S state which will have plus parity. The lowest state of the $Eu^{2+}(4f^7)$ configuration is an ^8S atomic state, L = 0 and Σl = 7·3 = 21. The only molecular state these two atomic states can form is an $^8\Sigma$ state. The parity of the oxygen ion is plus, that of the Eu ion is minus, and since Λ = 0 the molecular state must be $^8\Sigma^-$. Σ^+ - Σ^- transitions are forbidden, so the upper state must also be an $^8\Sigma^-$ state.

For a Σ^- state, all rotational levels with even N are of minus parity while the odd N levels have plus total parity (see Chapter 5). Start with N = 6. According to the previous rule, all these levels have minus total parity. To determine their parity excluding the rotational parity, I will use the definition for the parity labels e and f given by Brown et. al. [2] The convention is:

levels with parity $+(-1)^{J-1/2}$ are e levels

levels with parity $-(-1)^{J-1/2}$ are f levels.

For the N = 6 level, one spin component is J = N+7/2 = 9.5. $(-1)^{9.5-0.5}$ = -1 which is the total parity of the level, so this component is an e level. For the J = N-7/2 = 2.5, $(-1)^{2.5-0.5}$ is +1 which is -(total parity) so this component is an f level. The rotationless parity of each level is tabulated in Table 6-III. In addition the absolute and rotationless parity for a $^4\Sigma^-$ state is given in Figure 6-I.

6.5.2 Name

By convention, the spin components for a case b molecule are called F_1 through F_{2S+1} where F_1 is the lowest energy level for a given J (not N) and F_{2S+1} is the highest energy. For constant J, the lowest energy level will be the one with the lowest N, since BN(N+1) is the largest energy term involving J or N in the Hamiltonian. Consequently, F_1 is the N = J-7/2 level while F_8 is the N = J+7/2 level. The names are tabulated in Table 6-III. Figure 6-I gives the names for a $^4\Sigma^-$ state with a positive spin-spin constant. Above N=3, the ordering of the F names in a $^4\Sigma$ state will remain the same; for a given N F_4 will be the lowest followed by F_1 and then F_3 and F_2.

There may be some confusion about the number of levels at low N. Since $J = N + S$, with $S = 7/2$ one might expect to have four levels for $N = 0$, J=7/2, 5/2, 3/2 and 1/2. However, the equation to use is $J = S \pm 0 = 7/2$ so that there is only one level for $N = 0$, $J = 7/2$.

6.6 Assignment of the Spectrum

Eu has two isotopes of atomic mass 150.9196 and 152.9209, with abundances of 48% and 52%, and with nuclear magnetic moments of +3.4 and +1.5 respectively. Consequently, the spectrum should show a total of 32 branches, 8 R and 8 P from each isotope.

As mentioned in the description of the spectra, 12 branches, 6 P and 6 R, have been found. The N numbering was determined by taking the lower state combination difference, $R(N''-1) - P(N''+1) = \Delta_2F = 4B''_v (N''+1/2)$, and dividing it by $4B''_v$, which is easily obtained by taking the difference between consecutive Δ_2F, $\Delta_2F(N+1) - \Delta_2F(N)$. Since the lowest N for which a $\Delta_2F(N)$ was observed was 22, the N numbering could be incorrect by one. In the six subbranches found, levels with the same N are easily identified by the similarity of the lower state combination differences. These combination differences are given in Table 6-VI.

Identifying the four strongest, narrowest P-R branches as the $J = N \pm 1/2$ spin components is fairly straightforward. The $J = N \pm 1/2$ states should have the smallest hyperfine splitting; consequently, all six hyperfine components fall within the Doppler envelope, making the lines strong and sharp relative to the other spin components in which the hyperfine splitting is larger than the Doppler width. Two lines come from each isotope.

Figure 6-1: $^1\Sigma^-$ Energy Levels

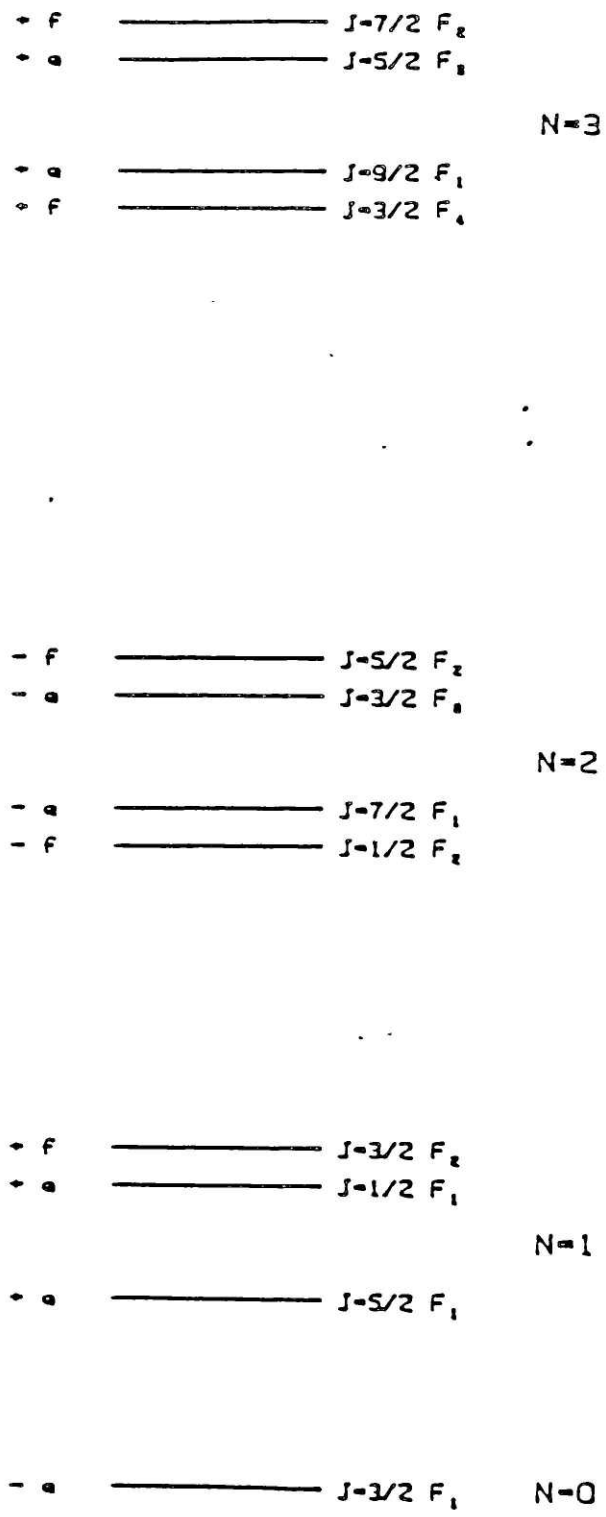


Table 6-III: Names and Parity of the Rotational Levels of an $^8\Sigma^-$ State

J	Name	Parity
N + 7/2	F_1	e
N - 7/2	F_8	f
N + 5/2	F_2	f
N - 5/2	F_7	e
N + 3/2	F_3	e
N - 3/2	F_6	f
N + 1/2	F_4	f
N - 1/2	F_5	e

Figure 6-2 shows the typical appearance of the four $J = N \pm 1/2$ lines. Notice that the lines labeled 2 and 4 show a slight broadening beyond the Doppler width. These are identified as belonging to the 151 isotope which has a nuclear magnetic moment of more than twice that of the 153 isotope. Lines 1 and 3 are the $J = N \pm 1/2$ components from the 153 isotope. Finally each line in the two pairs must be identified as the e or f parity line. For a given N, spin subbands with greater hyperfine lie to the blue of $J = N \pm 1/2$ lines. According to the matrix elements in Table 6-I the J = N-1/2 e lines should be closest to these additional spin components, so this gives the final identification of the narrow lines. Line 1 is the J = N+1/2 f component and line 3 is the J = N-1/2 e component of the 153 isotope while lines 2 and 4 are the f and e lines of the 151 isotope.

In addition to the four strongest spin subbands, two additional subbands, with much larger hyperfine splitting, were found. As noted, these bands lie to the blue of the $J = N \pm 1/2$ lines. A non-linear least squares fit of the $J = N \pm 1/2$ lines using the $^8\Sigma$ Hamiltonian identified the two branches as F_7 J=N-5/2 and F_8 J=N-7/2. However the fit was not able to associate the branches with the 151 isotope versus the 153 isotope. To make this final assignment, the hyperfine splitting must be used.

Figure 6-2: Strong EuO Lines

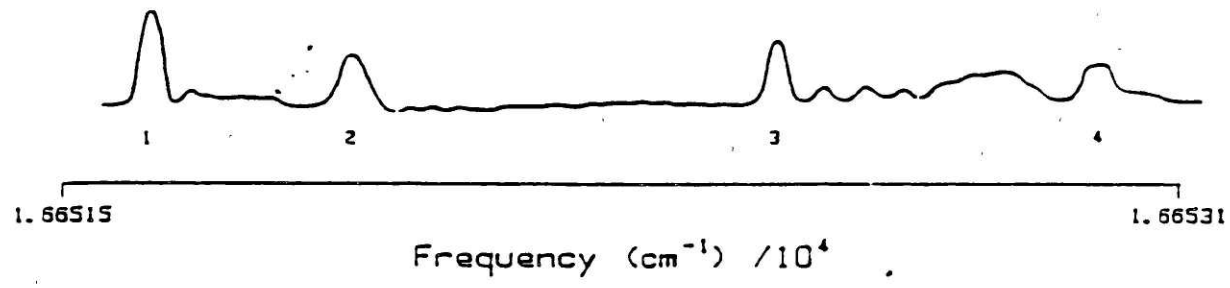


Table 6-IV shows the numerical formula for the separation of the two most widely split hyperfine components in an $^8\Sigma$ state. I can write the elements in the table in the following manner:

$$\text{separation} = b(K_{F=J+2.5}(N,J) - K_{F=J-2.5}(N,J)) + c(M_{F=J+2.5}(N,J) - M_{F=J-2.5}(N,J))$$

$$= b(\Delta K) - c(\Delta M) \qquad (6.26)$$

where K and M are coefficients determined by the formulas for the hyperfine energies given in equations (6.24) and (6.25). I will abbreviate these constants as $K_{2.5}$, $K_{-2.5}$, $M_{2.5}$ and $M_{-2.5}$. In the spectrum I do not directly observe the hyperfine energy splitting; rather I observe the difference between the splitting in the upper and lower states. As shown by Table 6-IV, K and M are relatively independent of N and J. Therefore even though the upper and lower states branches differ by one in N and J for P and R branches, for the hyperfine splittings I can treat all transitions as Q branch transitions. The hyperfine splitting observed in the spectrum corresponding to the seperation of

the frequency of two transitions, F'=J'+5/2 to F"=J"+5/2 and F'=J'-5/2 to F"=J"-5/2, will be given by

$$\text{Hf splitting} = [b'(K_{2.5} - K_{-2.5}) + c'(M_{2.5} - M_{-2.5})] - [b"(K_{2.5} - K_{-2.5}) + c"(M_{2.5} - M_{-2.5})].$$

Since the numerical coefficients K and M have approximately the same value for both upper and lower states (Q branch approximation) I can write

$$\text{Hf splitting} = \Delta b(K_{2.5} - K_{-2.5}) + \Delta(M_{2.5} - M_{-2.5})$$

$$= (\Delta b)(\Delta K) - (\Delta c)(\Delta M). \qquad (6.27)$$

The values of ΔK and ΔM are given by Table 6-IV.

With formula (6.27) I can assign the two remaining spin subbands. The average splitting of the J=N-7/2 lines is 0.260 cm^{-1}, while the average for the J=N-5/2 is 0.188 cm^{-1}. Using these values with the coefficients in Table 6-IV for N=40, there are two simultaneous equations to solve:

1. $0.260 = 17.27\Delta b + 1.63\Delta c$
2. $0.188 = 11.88\Delta b + 1.25\Delta c$.

The solution is $\Delta b = 0.0084$ cm^{-1} and $\Delta c = 0.070$ cm^{-1}. Putting these values into the formulas for the splitting of the narrow F_4 and F_5 components gives

$$\text{Hf splitting}(F_4, \text{calculated}) = 0.033 \text{ } cm^{-1}$$

Table 6-IV: Hyperfine Splittings of an $^8\Sigma$ State

Hyperfine splitting for EuO

N= 30

J	Name	Splitting
30.5	F_4, J=N+1/2, f	3.75b
29.5	F_5, J=N-1/2, e	-1.21b - 0.46c
31.5	F_3, J=N+3/2, e	8.56b + 0.48c
28.5	F_6, J=N-3/2, f	-6.34b - 0.88c
32.5	F_2, J=N+5/2, f	13.22b + 1.00c
27.5	F_7, J=N-5/2, e	-11.66b - 1.28c
33.5	F_1, J=N+7/2, e	17.76b + 1.54c
26.5	F_8, J=N-7/2, f	-17.18b - 1.65c

N= 40

J	Name	Splitting
40.5	F_4, J=N+1/2, f	3.45b + 0.05c
39.5	F_5, J=N-1/2, e	-1.53b - 0.40c
41.5	F_3, J=N+3/2, e	8.30b + 0.53c
38.5	F_6, J=N-3/2, f	-6.64b - 0.83c
42.5	F_2, J=N+5/2, f	13.05b + 1.03c
37.5	F_7, J=N-5/2, e	-11.88b - 1.25c
43.5	F_1, J=N+7/2, e	17.70b + 1.56c
36.5	F_8, J=N-7/2, f	-17.27b - 1.63c

N= 50

J	Name	Splitting
50.5	F_4, J=N+1/2, f	3.26b + 0.09c
49.5	F_5, J=N-1/2, e	-1.73b - 0.38c
51.5	F_3, J=N+3/2, e	8.15b + 0.56c
48.5	F_6, J=N-3/2, f	-6.81b - 0.80c
52.5	F_2, J=N+5/2, f	12.95b + 1.05c
47.5	F_7, J=N-5/2, e	-12.01b - 1.23c
53.5	F_1, J=N+7/2, e	17.66b + 1.57c
46.5	F_8, J=N-7/2, f	-17.32b - 1.63c

Hf splitting(F_5,calculated) = 0.041 cm^{-1}.

The experimental values for the splittings of the 151 isotope are

Hf splitting(F_4,experimental) $\sim$ 0.03 cm^{-1}

Hf splitting(F_5,experimental) $\sim$ 0.04 cm^{-1}.

The splittings of the 153 isotope will be smaller by (1.5/3.5), the ratio of the magnetic moments of the the two isotopes, much narrower than the Doppler width. Therefore, the lines of the 153 isotope should be too narrow to be grouped with the observed F_7 and F_8 subbranches. The hyperfine data indicates that the J=N-7/2 and J=N-5/2 branches belong to the 151 isotope. As I've mentioned special techniques were used to observe these branches. They were chosen because of their large hyperfine splitting. Since the higher spin 153 isotope subbands should have less splitting, I did not happen to choose them. Therefore I observed the higher spin subbands of only the 151 isotope.

6.7 The Fit

The non-linear least squares fitter used in fitting the data was based on the fitter of Marquardt [11]. Appendix B contains the Hamiltonian subroutine used in the fit. The program was based on the the Hamiltonian derived earlier for an $^8\Sigma$ state. The Hamiltonian used did not include the hyperfine interactions, since this would significantly raise the level of difficulty of the fitting procedure by greatly increasing the size of the basis sets needed for fitting. To remove the hyperfine splitting the frequencies of the six hyperfine

Table 6-V: EuO Transition Frequencies

151 Isotope

N	F4, J=N+1/2, f R(N)	F4, J=N+1/2, f P(N)	F5, J=N-1/2, e R(N)	F5, J=N-1/2, e P(N)
11		16718.698		
12		16716.656		
13		16714.939		
14		16712.949		
15		16710.839		
16		16708.616		
17		16706.315		
18		16703.917		
19		16701.412		16702.433
20		16698.806		16699.838
21		16696.111		16697.231
22	16718.088	16693.304		16694.472
23	16716.277	16690.432		
24	16714.293	16687.418		16688.566
25	16712.375	16684.323		16685.448
26	16710.273	16681.138		16682.222
27	16708.065	16677.815	16709.051	
28	16705.559	16674.412	16706.737	16675.659
29	16703.335	16670.908	16704.342	16672.102
30	16700.856	16667.330	16701.861	16668.472
31	16698.203	16663.612	16699.235	16664.791
32	16695.538	16659.811	16696.563	16660.928
33	16692.714	16656.028	16693.743	16657.007
34	16689.813	16651.912	16690.833	16652.981
35	16686.867	16647.788	16687.798	16648.855
36	16683.608	16643.617	16684.650	16644.635
37	16680.509	16639.388	16681.399	16640.316
38		16634.785	16678.040	16635.877
39	16673.598		16674.695	16631.322
40	16670.224		16671.135	16627.018
41	16666.665	16620.999	16667.476	16622.030
42		16616.231	16663.697	16617.178
43		16611.360	16659.992	16612.219
44		16606.515	16655.932	16607.175
45	16650.809		16651.888	16602.069
46			16647.741	16596.799
47	16642.547	16590.354	16643.450	16591.475
48	16638.072		16639.104	16586.028
49	16633.643	16579.539	16634.645	16580.452
50	16629.134		16630.119	16574.875
51		16568.044	16625.053	16569.161
52	16619.623	16562.246	16620.566	16563.149
53	16614.802	16556.144	16615 749	16556.808
54	16609 865	16550.175	16610.827	16551.014
55	16604.807	16544 053	16605.772	16544.924
56	16599.579	16537 842	16600.577	16538.702
57	16594 159	16531 500	16595.282	16532.375
58	16588 723	16524 985	16589.758	16525.898
59	16583.227			

F7, J=N-5/2, e

N			P Lines			
36	16646.681	16646.702	16646.729		16646.808	
37	16644.620	16644.648	16644.683	16644.717	16644.753	16644.786
38	16641.185	16641.201	16641.229	16641.257	16641.286	16641.318
39	16637.087	16637.104	16637.131	16637.162	16637.197	16637.234
40	16632.654	16632.683	16632.717	16632.751	16632.789	16632.828
41	16628.038	16628.068	16628.109	16628.146	16628.183	16628.221
42	16623.295	16623.332	16623.369	16623.408	16623.447	16623.484
43	16618.431	16618.463	16618.500	16618.533	16618.572	16618.627
44	16613.484	16613.527	16613.562	16613.602	16613.636	16613.671
45	16608.359	16608.399	16608.439	16608.475	16608.511	16608.546
46	16603.205	16603.249	16603.287	16603.325	16603.362	16603.395
47	16597.940	16597.992	16598.032	16598.070	16598.105	16598.124
48	16592.504	16592.589	16592.621	16592.655	16592.691	16592.727
49	16587.094	16587.129	16587.165	16587.202	16587.237	16587.273
50	16581.580	16581.622	16581.657	16581.688	16581.714	16581.738
51	16575.498	16575.541	16575.586	16575.628	16575.670	16575.711
52	16569.651	16569.701	16569.747	16569.791	16569.837	16569.877
53	16563.643	16563.666	16563.703	16563.739	16563.769	16563.811
54	16557.701	16557.744	16557.785	16557.821	16557.862	16557.886
55	16551.505	16551.526	16551.561	16551.598	16551.629	16551.666
56	16545.487	16545.516	16545.549	16545.581	16545.660	16545.680
57	16539.103	16539.138	16539.173	16539.210	16539.247	16539.284
58	16532.722	16532.763	16532.804	16532.839	16532.875	16532.906
			R Lines			
34	16692.993	16693.020	16693.042			
35	16692.144	16692.174	16692.203	16692.237	16692.271	16692.308
36	16690.119	16690.135	16690.159	16690.185	16690.212	16690.244
37	16687.308	16687.327	16687.353	16687.386	16687.416	16687.452
38	16684.170	16684.200	16684.233	16684.269	16684.305	16684.304
39	16680.852	16680.887	16680.923	16680.959	16680.996	16681.036
40	16677.399	16677.437	16677.473	16677.510	16677.549	16677.587
41	16673.828	16673.860	16673.899	16673.930	16673.966	16674.025
42	16670.177	16670.218	16670.257	16670.293	16670.331	16670.363
43	16666.354	16666.389	16666.427	16666.466	16666.501	16666.535
44	16662.489	16662.532	16662.572	16662.611	16662.646	16662.681
45	16658.503	16658.555	16658.595	16658.634	16658.667	16658.695
46	16654.421	16654.453	16654.484	16654.515	16654.549	16654.587
47	16650.241	16650.278	16650.311	16650.347	16650.384	16650.418
48	16646.017	16646.059	16646.089	16646.129	16646.154	16646.173
49	16642.539	16642.584	16642.628	16642.671	16642.712	16642.753
50	16637.985	16638.035	16638.076	16638.126	16638.165	16638.208
51	16631.942	16631.971	16632.000	16632.035	16632.069	16632.107
52	16627.293	16627.334	16627.237	16627.409	16627.445	16627.484
53	16622.359	16622.397	16622.431	16622.466	16622.500	16622.537
54	16617.553		16617.649	16617.682	16617.715	16617.747
55	16612.510	16612.584	16612.618	16612.654	16612.692	16612.731
56	16607.452	16607.488	16607.541	16607.569	16607.652	16607.633
57	16602.311	16602.351	16602.386	16602.424	16602.461	16602.497

F8, J = N-7/2, f

N						
			R Lines			
37	16694.012	16694.071	16694.127	16694.183	16694.241	16694.290
38	16690.731	16690.786	16690.833	16690.887	16690.942	16691.000
39					16687.501	16687.549
40			16683.930	16683.970		
41						
42	16676.732	16676.777	16676.823	16676.867	16676.916	16676.960
43	16672.887	16672.937	16672.992	16673.048	16673.096	16673.148
44	16669.000	16669.069	16669.111	16669.164	16669.215	16669.252
45		16665.123	16665.179	16665.274	16665.278	16665.313
			P Lines			
38					16648.463	16648.516
39	16643.771	16643.831	16643.889	16643.944	16643.996	16644.049
40	16639.190	16639.251	16639.311	16639.346	16639.419	16639.470
41			16634.560	16634.620		
42	16629.637			16629.787	16629.832	16629.871
43	16625.255	16625.301				
44	16620.007	16620.062	16620.106	16620.152		16620.240

153 Isotope

N	F4, J=N+1/2, f R(N)	F4, J=N+1/2, f P(N)	F5, J=N-1/2, e R(N)	F5, J=N-1/2, e P(N)
17		16705.949		16706.883
18		16703.559		16704.530
19		16701.080		16702.062
20		16698.492		16699.480
21		16695.775		
22	16717.787	16693.002		16693.876
23	16715.981	16690.189		
24	16714.079	16687.155		16688.170
25	16712.076	16684.050		16685.030
26	16709.980	16680.845		16681.829
27	16707.800	16677.552		16678.500
28	16705.479	16674.156	16706.438	16675.083
29	16703.086	16670.668	16704.006	16671.577
30	16700.543	16667.054	16701.520	16667.945
31	16697.933	16663.336	16698.900	16664.229
32	16695.220	16659.524	16696.183	16660.444
33	16692.401	16655.617	16693.362	16656.531
34	16689.476	16651.614	16690.444	16652.514
35	16686.451	16647.497	16687.402	16648.388
36	16683.291	16643.268	16684.269	16644.175
37	16680.206	16638.938	16680.995	16639.835
38	16676.916	16634.487	16677.555	16635.510
39	16673.475	16630.087	16674.287	16630.838
40	16669.954	16625.511	16670.926	16626.373
41	16666.278	16620.787	16667.340	16621.784
42	16662.532	16615.977	16663.490	16616.879
43	16658.695	16611.111	16659.523	
44	16654.654	16605.972	16655.554	16606.868
45	16650.687		16651.615	16601.662
46	16646.580	16595.510	16647.552	16596.330
47	16642.413	16590.250	16643.391	16591.111
48	16637.567	16584.844	16638.946	16585.791
49	16633.325	16579.383	16634.241	
50	16628.807	16573.418	16629.722	
51	16624.135	16567.749	16625.053	16568.580
52	16619.294	16561.932	16620.231	16562.770
53	16614.436	16555.976	16615.400	16556.711
54	16609.535	16549.862	16610.448	16550.705
55	16604.501	16543.702	16605.382	16544.584
56	16599.208	16537.525	16600.108	16538.348
57	16593.924	16531.205	16594.799	16531.993
58	16588.377	16524.638	16589.338	16525.439
59		16518.068		16518.845
60		16511.179		16512.027

Table 6-VI: Combination Differences for EuO Subbands

151 Isotope

N	f Parity, J=N+1/2				e Parity, J=N-1/2			
	Lower	Lower/4(N+1/2)	Upper	Upper/4(N+1/2)	Lower	Lower/4(N+1/2)	Upper	Upper/4(N+1/2)
22			24.784	0.27538				
23	30.670	0.32628	25.845	0.27495				
24	31.954	0.32606	26.875	0.27423				
25	33.155	0.32505	28.052	0.27502				
26	34.560	0.32604	29.135	0.27486				
27	35.861	0.32601	30.250	0.27500				
28	37.157	0.32594	31.147	0.27322	36.949	0.32411	31.078	0.27261
29	38.229	0.32397	32.427	0.27481	38.265	0.32428	32.240	0.27322
30	39.723	0.32560	33.526	0.27480	39.551	0.32419	33.389	0.27368
31	41.045	0.32575	34.591	0.27453	40.933	0.32487	34.444	0.27337
32	42.175	0.32442	35.727	0.27482	42.228	0.32483	35.635	0.27412
33	43.626	0.32557	36.686	0.27378	43.582	0.32524	36.736	0.27415
34	44.926	0.32555	37.901	0.27464	44.888	0.32528	37.852	0.27429
35	46.196	0.32532	39.079	0.27520	46.198	0.32534	38.943	0.27425
36	47.479	0.32520	39.991	0.27391	47.482	0.32522	40.015	0.27408
37	48.823	0.32549	41.121	0.27414	48.773	0.32515	41.083	0.27389
38					50.077	0.32518	42.163	0.27379
39					51.022	0.32292	43.373	0.27451
40	52.599	0.32469			52.665	0.32509	44.117	0.27233
41	53.993	0.32526	45.666	0.27510	53.957	0.32504	45.446	0.27377
42	55.305	0.32532			55.257	0.32504	46.519	0.27364
43					56.522	0.32484	47.773	0.27456
44					57.923	0.32541	48.757	0.27392
45					59.133	0.32491	49.819	0.27373
46	60.455	0.32503			60.413	0.32480	50.942	0.27388
47			52.193	0.27470	61.713	0.32481	51.975	0.27355
48	63.008	0.32478			62.998	0.32473	53.076	0.27359
49			54.104	0.27325	64.229	0.32439	54.193	0.27370
50	65.599	0.32475			65.484	0.32418	55.244	0.27349
51	66.888	0.32470			66.970	0.32510	55.892	0.27132
52			57.377	0.27322	68.245	0.32498	57.417	0.27341
53	69.448	0.32452	58.658	0.27410	69.552	0.32501	58.941	0.27543
54	70.749	0.32454	59.690	0.27381	70.825	0.32489	59.813	0.27437
55	72.023	0.32443	60.754	0.27367	72.125	0.32489	60.848	0.27409
56	73.307	0.32437	61.737	0.27317	73.397	0.32477	61.875	0.27378
57	74.594	0.32432	62.659	0.27243	74.679	0.32469	62.907	0.27351
58			63.738	0.27238			63.860	0.27291

e Parity, J=N-5/2

N	Lower	Lower/4(N+1/2)	Upper	Upper/4(N+1/2)
35	46.305	0.32609		
36	47.523	0.32550	43.437	0.29751
37	48.865	0.32577	42.677	0.28451
38	50.219	0.32610	42.940	0.27883
39	51.516	0.32605	43.783	0.27711
40	52.814	0.32601	44.752	0.27625
41	54.103	0.32592	45.797	0.27589
42	55.398	0.32587	46.880	0.27576
43	56.692	0.32582	47.916	0.27538
44	57.992	0.32580	49.007	0.27532
45	59.285	0.32574	50.146	0.27553
46	60.567	0.32563	51.204	0.27529
47	61.888	0.32573	52.298	0.27525
48	63.146	0.32549	53.479	0.27566
49	64.436	0.32543		
50				
51			56.420	0.27388
52	68.298	0.32523	57.624	0.27440
53	69.594	0.32521	58.721	0.27440
54	70.862	0.32506	59.856	0.27457
55	72.066	0.32462	61.035	0.27493
56	73.427	0.32490	61.959	0.27415
57	74.729	0.32491		

f Parity, J=N-7/2

N	Lower	Lower/4(N+1/2)	Upper	Upper/4(N+1/2)
37	48.729	0.32486		
38	50.258	0.32635	42.500	0.27597
39	51.541	0.32621	43.540	0.27557
40	52.843	0.32619	44.611	0.27538
41	54.176	0.32636		
42			47.078	0.27693
43	56.713	0.32594	47.680	0.27402
44			49.008	0.27533
45				
46	60.615	0.32589		

153 Isotope

N	f Parity, J=N+1/2 Lower	Lower/4(N+1/2)	Upper	Upper/4(N+1/2)	e Parity, J=N-1/2 Lower	Lower/4(N+1/2)	Upper	Upper/4(N+1/2)
22			24.785	0.27539				
23	30.632	0.32587	25.792	0.27438				
24	31.931	0.32583	26.924	0.27473				
25	33.234	0.32582	28.026	0.27476				
26	34.524	0.32570	29.135	0.27486				
27	35.824	0.32567	30.248	0.27498				
28	37.132	0.32572	31.323	0.27476				
29	38.425	0.32564	32.418	0.27473			31.355	0.27504
30	39.750	0.32582	33.489	0.27450	38.493	0.32621	32.429	0.27482
31	41.019	0.32555	34.597	0.27458	39.777	0.32604	33.575	0.27520
32	42.316	0.32551	35.696	0.27458	41.076	0.32600	34.671	0.27517
33	43.606	0.32542	36.784	0.27451	42.369	0.32592	35.739	0.27492
34	44.904	0.32539	37.862	0.27436	43.669	0.32589	36.831	0.27486
35	46.208	0.32541	38.954	0.27432	44.974	0.32590	37.930	0.27486
36	47.513	0.32543	40.023	0.27413	46.269	0.32584	39.014	0.27475
37	48.804	0.32536	41.268	0.27512	47.567	0.32580	40.094	0.27462
38	50.119	0.32545	42.429	0.27551	48.759	0.32506	41.160	0.27440
39	51.405	0.32535	43.388	0.27461	50.157	0.32569	42.045	0.27302
40	52.688	0.32523	44.443	0.27434	51.182	0.32394	43.449	0.27499
41	53.977	0.32516	45.491	0.27404	52.503	0.32409	44.553	0.27502
42	55.167	0.32451	46.555	0.27385	54.047	0.32558	45.556	0.27443
43	56.560	0.32506	47.584	0.27347			46.611	0.27418
44			48.682	0.27349	56.622	0.32541		
45	59.144	0.32497			57.861	0.32506	48.686	0.27352
46	60.437	0.32493	51.070	0.27457	59.224	0.32541	49.953	0.27447
47	61.736	0.32493	52.163	0.27454	60.504	0.32529	51.222	0.27539
48	63.030	0.32490	52.723	0.27177	61.761	0.32506	52.280	0.27516
49	64.149	0.32398	53.942	0.27243			53.155	0.27399
50	65.576	0.32463	55.389	0.27420	65.661	0.32505		
51	66.875	0.32464	56.386	0.27372	66.952	0.32501	56.473	0.27414
52	68.159	0.32457	57.362	0.27315	68.342	0.32544	57.461	0.27362
53	69.432	0.32445	58.460	0.27318	69.526	0.32489	58.689	0.27425
54	70.734	0.32447	59.673	0.27373	70.816	0.32484	59.743	0.27405
55	72.010	0.32437	60.799	0.27387	72.100	0.32477	60.798	0.27386
56	73.296	0.32432	61.683	0.27293	73.389	0.32473	61.760	0.27327
57	74.570	0.32422	62.719	0.27269	74.669	0.32465	62.806	0.27307
58	75.856	0.32417	63.739	0.27239	75.954	0.32459	63.899	0.27307
59	77.198	0.32436			77.311	0.32484		

components were averaged together with each component receiving equal weigth. It is possible to ignore the hyperfine interactions for several reasons. The lines with measurable hyperfine have been identified only for relatively high N, where the separation of the lines due to the BN(N+1) term has reduced the size of the interactions between spin components. Also the upper state is obviously perturbed and the strength of the perturbation, as shown by the scatter of the upper state combination differences listed in Table 6-VI, will mask the smaller hyperfine interactions until a complete deperturbation is done.

The constants obtained from the fit are listed in Table 6-VII.

Table 6-VII: EuO Constants

	151 Isotope	
Name	Value(cm^{-1})	Standard Deviation(cm^{-1})
T_v	16735.79	0.02
B"	0.32582	0.0001
D"	2.0×10^{-7}	$0\ 2 \times 10^{-7}$
λ"	0.31	0.01
γ"	-0.014	0.005
B'	0.27540	0.0001
D'	$3\ 4 \times 10^{-7}$	0.2×10^{-7}
λ'	-0 64	0.01
γ'	-0.024	0.005

	153 Isotope	
Name	Value(cm^{-1})	Standard Deviation(cm^{-1})
T_v	16735.79	0 02
B"	0 32624	0.00009
D"	2.46×10^{-7}	$0\ 2 \times 10^{-7}$
λ"	0 42	0 4
γ"	-0.020	0.005
B'	0.2758	0.0001
D'	$3\ 73 \times 10^{-7}$	$0\ 2 \times 10^{-7}$
λ'	-0 582	0 04
γ'	-0 030	0 005

Table 6-VIII is the output from the fit. If an experimental line is listed with an uncertainty greater than 1.0, that line was statistically excluded from the fit. I kept the line in the data set (deweighted) in order to compare the calculated and experimental location. The columns in the fit may be decoded as follows. Eigenvectors labelled 4, 3, 2 or 1 correspond respectively to J = N ±1/2, ±3/2, ±5/2 or ±7/2. The levels with a block number of 4 or 2 are f parity levels while those with a block number of 3 or 1 are e parity levels.

6.8 Discussion

6.8.1 Constants

Considering that the upper state is perturbed, the fit is quite good. One test to see if perturbation effects are being included in the value of the centrifugal distortion constant is to check the fitted value against a theoretical value of $D = 4B^3/\omega^2$ [9] based on the harmonic nonrigid rotator model. The calculated values for the lower state, using an ω_e of 680 cm^{-1} are given in Table 6-IX. The value for the 153 isotope is in rather close agreement. Even the 151 fitted number is acceptably close, although there is a reason why the results for the 151 isotope may not be as good. The fit shows that the J = N+3/2 component of both isotopes lies almost directly on top of the J = N-1/2 component of the 151 isotope. This could cause the J = N - 1/2 component to be incorrectly measured. Since this component contributed over 1/3 of the lines used in the fit, a systematic measurement bias could easily affect the fitted D value.

The value of the upper state vibrational constant is not known, so the

Table 6-VIII: EuO Fit to Experimental Data

151 Isotope

J'	Eigenvector	Block	J"	Eigenvector	Block	Expt.	Calc.	Expt-Calc	Expt. Uncert.
27.5	4	3	26.5	4	1	16709.05100	16709.06856	-0.01756	0.00600
28.5	4	3	27.5	4	1	16706.73700	16706.75825	-0.02125	0.00600
29.5	4	3	28.5	4	1	16704.34200	16704.34762	-0.00562	0.00600
30.5	4	3	29.5	4	1	16701.86100	16701.83634	0.02466	0.00600
31.5	4	3	30.5	4	1	16699.23500	16699.22413	0.01087	0.00600
32.5	4	3	31.5	4	1	16696.56300	16696.51072	0.05228	0.00600
33.5	4	3	32.5	4	1	16693.74300	16693.69584	0.04716	0.00600
34.5	4	3	33.5	4	1	16690.83300	16690.77925	0.05375	0.00600
35.5	4	3	34.5	4	1	16687.79800	16687.76073	0.03727	0.00600
36.5	4	3	35.5	4	1	16684.65000	16684.64005	0.00995	0.00600
37.5	4	3	36.5	4	1	16681.39900	16681.41701	-0.01801	0.00600
38.5	4	3	37.5	4	1	16678.04000	16678.09138	-0.05138	0.00600
39.5	4	3	38.5	4	1	16674.69500	16674.66298	0.03202	0.00600
40.5	4	3	39.5	4	1	16671.13500	16671.13160	0.00340	0.00600
41.5	4	3	40.5	4	1	16667.47600	16667.49705	-0.02105	0.00600
42.5	4	3	41.5	4	1	16663.69700	16663.75914	-0.06214	0.00600
43.5	4	3	42.5	4	1	16659.99200	16659.91768	0.07432	0.00600
44.5	4	3	43.5	4	1	16655.93200	16655.97247	-0.04047	0.00600
45.5	4	3	44.5	4	1	16651.88800	16651.92333	-0.03533	0.00600
46.5	4	3	45.5	4	1	16647.74100	16647.77008	-0.02908	0.00600
47.5	4	3	46.5	4	1	16643.45000	16643.51251	-0.06251	0.00600
48.5	4	3	47.5	4	1	16639.10400	16639.15045	-0.04645	0.00600
49.5	4	3	48.5	4	1	16634.64500	16634.68370	-0.03870	0.00600
50.5	4	3	49.5	4	1	16630.11900	16630.11208	0.00692	0.00600
51.5	4	3	50.5	4	1	16625.05300	16625.43538	-0.38238	100.00500
52.5	4	3	51.5	4	1	16620.56600	16620.65342	-0.08742	0.00600
53.5	4	3	52.5	4	1	16615.74900	16615.76600	-0.01700	0.00600
54.5	4	3	53.5	4	1	16610.82700	16610.77293	0.05407	0.00600
55.5	4	3	54.5	4	1	16605.77200	16605.67400	0.09800	0.00600
56.5	4	3	55.5	4	1	16600.57700	16600.46902	0.10798	0.00600
57.5	4	3	56.5	4	1	16595.28200	16595.15778	0.12422	0.00600
58.5	4	3	57.5	4	1	16589.75800	16589.74008	0.01792	0.00600
17.5	4	3	18.5	4	1	16702.43300	16702.63377	-0.20077	100.00500
18.5	4	3	19.5	4	1	16699.83800	16699.99738	-0.15938	100.00500
19.5	4	3	20.5	4	1	16697.23100	16697.26729	-0.03629	0.00600
20.5	4	3	21.5	4	1	16694.47200	16694.44234	0.02966	0.00600
21.5	4	3	22.5	4	1	16600.00000	16691.52157	-91.52157	100.00500
22.5	4	3	23.5	4	1	16688.56600	16688.50415	0.06185	0.00600
23.5	4	3	24.5	4	1	16685.44800	16685.38938	0.05862	0.00600
24.5	4	3	25.5	4	1	16682.22200	16682.17665	0.04535	0.00600
25.5	4	3	26.5	4	1	16600.00000	16678.86545	-78.86546	100.00500
26.5	4	3	27.5	4	1	16675.65900	16675.45531	0.20369	100.00599
27.5	4	3	28.5	4	1	16672.10200	16671.94581	0.15619	100.00599
28.5	4	3	29.5	4	1	16668.47200	16668.33658	0.13542	0.00600
29.5	4	3	30.5	4	1	16664.79100	16664.62730	0.16370	0.00600
30.5	4	3	31.5	4	1	16660.92800	16660.81766	0.11034	0.00600
31.5	4	3	32.5	4	1	16657.00700	16656.90737	0.09963	0.00600
32.5	4	3	33.5	4	1	16652.98100	16652.89618	0.08482	0.00600
33.5	4	3	34.5	4	1	16648.85500	16648.78384	0.07116	0.00600
34.5	4	3	35.5	4	1	16644.63500	16644.57012	0.06488	0.00600
35.5	4	3	36.5	4	1	16640.31600	16640.25481	0.06119	0.00600

36.5	4	3	37.5	4	1	16635.87700	16635.83768	0.03932	0.00600
37.5	4	3	38.5	4	1	16631.32200	16631.31855	0.00345	0.00600
38.5	4	3	39.5	4	1	16627.01800	16626.69721	0.32079	100.00500
39.5	4	3	40.5	4	1	16622.03000	16621.97347	0.05653	0.00600
40.5	4	3	41.5	4	1	16617.17800	16617.14715	0.03085	0.00600
41.5	4	3	42.5	4	1	16612.21900	16612.21806	0.00094	0.00600
42.5	4	3	43.5	4	1	16607.17500	16607.18602	-0.01102	0.00600
43.5	4	3	44.5	4	1	16602.06900	16602.05084	0.01816	0.00600
44.5	4	3	45.5	4	1	16596.79900	16596.81236	-0.01336	0.00600
45.5	4	3	46.5	4	1	16591.47500	16591.47039	0.00461	0.00600
46.5	4	3	47.5	4	1	16586.02800	16586.02475	0.00325	0.00600
47.5	4	3	48.5	4	1	16580.45200	16580.47527	-0.02327	0.00600
48.5	4	3	49.5	4	1	16574.87500	16574.82177	0.05323	0.00600
49.5	4	3	50.5	4	1	16569.16100	16569.06406	0.09694	0.00600
50.5	4	3	51.5	4	1	16563.14900	16563.20196	-0.05296	0.00600
51.5	4	3	52.5	4	1	16556.80800	16557.23530	-0.42730	100.00500
52.5	4	3	53.5	4	1	16551.01400	16551.16389	-0.14989	100.00500
53.5	4	3	54.5	4	1	16544.92400	16544.98755	-0.06355	100.00500
54.5	4	3	55.5	4	1	16538.70200	16538.70608	-0.00408	100.00500
55.5	4	3	56.5	4	1	16532.37500	16532.31930	0.05570	0.00600
56.5	4	3	57.5	4	1	16525.89800	16525.82702	0.07098	0.00600
23.5	4	4	22.5	4	2	16718.08800	16718.01944	0.06856	0.00600
24.5	4	4	23.5	4	2	16716.27700	16716.22347	0.05353	0.00600
25.5	4	4	24.5	4	2	16714.29300	16714.32774	-0.03474	0.00600
26.5	4	4	25.5	4	2	16712.37500	16712.33184	0.04316	0.00600
27.5	4	4	26.5	4	2	16710.27300	16710.23538	0.03762	0.00600
28.5	4	4	27.5	4	2	16708.06500	16708.03803	0.02697	0.00600
29.5	4	4	28.5	4	2	16705.55900	16705.73949	-0.18049	100.00599
30.5	4	4	29.5	4	2	16703.33500	16703.33952	-0.00452	0.00600
31.5	4	4	30.5	4	2	16700.85600	16700.83785	0.01815	0.00600
32.5	4	4	31.5	4	2	16698.20300	16698.23427	-0.03127	0.00600
33.5	4	4	32.5	4	2	16695.53800	16695.52858	0.00942	0.00600
34.5	4	4	33.5	4	2	16692.71400	16692.72057	-0.00657	0.00600
35.5	4	4	34.5	4	2	16689.81300	16689.81006	0.00294	0.00600
36.5	4	4	35.5	4	2	16686.86700	16686.79688	0.07012	0.00600
37.5	4	4	36.5	4	2	16683.60800	16683.68084	-0.07284	0.00600
38.5	4	4	37.5	4	2	16680.50900	16680.46178	0.04722	0.00600
39.5	4	4	38.5	4	2	16600.00000	16677.13954	-77.13954	100.00500
40.5	4	4	39.5	4	2	16673.59800	16673.71395	-0.11595	0.00600
41.5	4	4	40.5	4	2	16670.22400	16670.18484	0.03916	0.00600
42.5	4	4	41.5	4	2	16666.66500	16666.55205	0.11295	0.00600
43.5	4	4	42.5	4	2	16663.09400	16662.81543	0.27857	100.00500
44.5	4	4	43.5	4	2	16600.00000	16658.97480	-58.97480	100.00500
45.5	4	4	44.5	4	2	16600.00000	16655.03001	-55.03001	100.00500
46.5	4	4	45.5	4	2	16650.80900	16650.98089	-0.17189	100.00599
47.5	4	4	46.5	4	2	16600.00000	16646.82727	-46.82727	100.00500
48.5	4	4	47.5	4	2	16642.54700	16642.56899	-0.02199	0.00600
49.5	4	4	48.5	4	2	16638.07200	16638.20587	-0.13387	0.00600

50.5	4	4	49.5	4	2	16633.64300	16633.73775	-0.09475	0.00600
51.5	4	4	50.5	4	2	16629.13400	16629.16445	-0.03045	0.00600
52.5	4	4	51.5	4	2	16600.00000	16624.48579	-24.48579	100.00500
53.5	4	4	52.5	4	2	16619.62300	16619.70160	-0.07860	0.00600
54.5	4	4	53.5	4	2	16614.80200	16614.81169	-0.00969	0.00600
55.5	4	4	54.5	4	2	16609.86500	16609.81587	0.04913	0.00600
56.5	4	4	55.5	4	2	16604.80700	16604.71397	0.09303	0.00600
57.5	4	4	56.5	4	2	16599.57900	16599.50579	0.07321	0.00600
58.5	4	4	57.5	4	2	16594.15900	16594.19114	-0.03214	0.00600
59.5	4	4	58.5	4	2	16588.72300	16588.76983	-0.04683	0.00600
60.5	4	4	59.5	4	2	16583.22700	16583.24165	-0.01465	0.00600
10.5	4	4	11.5	4	2	16718.69800	16719.09314	-0.39514	100.00599
11.5	4	4	12.5	4	2	16716.65600	16717.17540	-0.51940	100.00500
12.5	4	4	13.5	4	2	16714.93900	16715.18214	-0.24314	100.00599
13.5	4	4	14.5	4	2	16712.94900	16713.10952	-0.16052	0.00600
14.5	4	4	15.5	4	2	16710.83900	16710.95394	-0.11494	0.00600
15.5	4	4	16.5	4	2	16708.61600	16708.71222	-0.09622	0.00600
16.5	4	4	17.5	4	2	16706.31500	16706.38165	-0.06665	0.00600
17.5	4	4	18.5	4	2	16703.91700	16703.95996	-0.04296	0.00600
18.5	4	4	19.5	4	2	16701.41200	16701.44528	-0.03328	0.00600
19.5	4	4	20.5	4	2	16698.80600	16698.83609	-0.03009	0.00600
20.5	4	4	21.5	4	2	16696.11100	16696.13113	-0.02013	0.00600
21.5	4	4	22.5	4	2	16693.30400	16693.32936	-0.02536	0.00600
22.5	4	4	23.5	4	2	16690.43200	16690.42993	0.00207	0.00600
23.5	4	4	24.5	4	2	16687.41800	16687.43214	-0.01414	0.00600
24.5	4	4	25.5	4	2	16684.32300	16684.33538	-0.01238	0.00600
25.5	4	4	26.5	4	2	16681.13800	16681.13915	-0.00115	0.00600
26.5	4	4	27.5	4	2	16677.81500	16677.84302	-0.02802	0.00600
27.5	4	4	28.5	4	2	16674.41200	16674.44663	-0.03463	0.00600
28.5	4	4	29.5	4	2	16670.90800	16670.94964	-0.04164	0.00600
29.5	4	4	30.5	4	2	16667.33000	16667.35177	-0.02177	0.00600
30.5	4	4	31.5	4	2	16663.61200	16663.65276	-0.04076	0.00600
31.5	4	4	32.5	4	2	16659.81100	16659.85238	-0.04138	0.00600
32.5	4	4	33.5	4	2	16656.02800	16655.95041	0.07759	0.00600
33.5	4	4	34.5	4	2	16651.91200	16651.94666	-0.03466	0.00600
34.5	4	4	35.5	4	2	16647.78800	16647.84093	-0.05293	0.00600
35.5	4	4	36.5	4	2	16643.61700	16643.63306	-0.01606	0.00600
36.5	4	4	37.5	4	2	16639.38800	16639.32286	0.06514	0.00600
37.5	4	4	38.5	4	2	16634.78500	16634.91017	-0.12517	0.00600
38.5	4	4	39.5	4	2	16630.69200	16630.39485	0.29715	100.00500
39.5	4	4	40.5	4	2	16626.14200	16625.77671	0.36529	100.00500
40.5	4	4	41.5	4	2	16620.99900	16621.05562	-0.05662	0.00600
41.5	4	4	42.5	4	2	16616.23100	16616.23142	-0.00042	0.00600
42.5	4	4	43.5	4	2	16611.36000	16611.30396	0.05604	0.00600
43.5	4	4	44.5	4	2	16606.51500	16606.27308	0.24192	100.00500
44.5	4	4	45.5	4	2	16600.00000	16601.13862	-1.13862	100.00500
45.5	4	4	46.5	4	2	16600.00000	16595.90045	4.09955	100.00500

46.5	4	4	47.5	4	2	16590.35400	16590.55839	-0.20439	100.00500
47.5	4	4	48.5	4	2	16600.00000	16585.11229	14.88770	100.00500
48.5	4	4	49.5	4	2	16579.53900	16579.56200	-0.02300	0.00600
49.5	4	4	50.5	4	2	16600.00000	16573.90736	26.09264	100.00500
50.5	4	4	51.5	4	2	16568.04400	16568.14820	-0.10420	0.00600
51.5	4	4	52.5	4	2	16562.24600	16562.28435	-0.03835	0.00600
52.5	4	4	53.5	4	2	16556.14400	16556.31566	-0.17166	0.00600
53.5	4	4	54.5	4	2	16550.17500	16550.24194	-0.06694	0.00600
54.5	4	4	55.5	4	2	16544.05300	16544.06304	-0.01004	0.00600
55.5	4	4	56.5	4	2	16537.84200	16537.77877	0.06323	0.00600
56.5	4	4	57.5	4	2	16531.50000	16531.38896	0.11104	0.00600
57.5	4	4	58.5	4	2	16524.98500	16524.89343	0.09157	0.00600
32.5	2	3	31.5	2	1	16693.05000	16697.62531	-4.57531	100.02000
33.5	2	3	32.5	2	1	16692.22600	16694.61888	-2.39288	100.02000
34.5	2	3	33.5	2	1	16690.18200	16691.51049	-1.32849	100.02000
35.5	2	3	34.5	2	1	16687.38000	16688.29991	-0.91991	100.01500
36.5	2	3	35.5	2	1	16684.25700	16684.98695	-0.72995	100.01500
37.5	2	3	36.5	2	1	16680.94400	16681.57140	-0.62740	100.01500
38.5	2	3	37.5	2	1	16677.49300	16678.05305	-0.56005	100.01500
39.5	2	3	38.5	2	1	16673.92700	16674.43171	-0.50471	0.02000
40.5	2	3	39.5	2	1	16670.27000	16670.70717	-0.43717	0.02000
41.5	2	3	40.5	2	1	16666.44500	16666.87925	-0.43425	0.02000
42.5	2	3	41.5	2	1	16662.58500	16662.94775	-0.36275	0.02000
43.5	2	3	42.5	2	1	16658.59900	16658.91247	-0.31347	0.02000
44.5	2	3	43.5	2	1	16654.50400	16654.77323	-0.26923	0.02000
45.5	2	3	44.5	2	1	16650.33000	16650.52982	-0.19982	0.02000
46.5	2	3	45.5	2	1	16646.09500	16646.18206	-0.08706	0.02000
47.5	2	3	46.5	2	1	16642.64600	16641.72976	0.91624	100.02000
48.5	2	3	47.5	2	1	16638.09700	16637.17270	0.92430	100.02000
49.5	2	3	48.5	2	1	16632.02500	16632.51071	-0.48571	0.02000
50.5	2	3	49.5	2	1	16627.38800	16627.74358	-0.35558	0.02000
51.5	2	3	50.5	2	1	16622.44800	16622.87111	-0.42311	0.02000
52.5	2	3	51.5	2	1	16617.65000	16617.89310	-0.24310	0.02000
53.5	2	3	52.5	2	1	16612.62100	16612.80935	-0.18835	0.02000
54.5	2	3	53.5	2	1	16607.54300	16607.61965	-0.07665	0.02000
55.5	2	3	54.5	2	1	16602.40400	16602.32380	0.08020	0.02000
32.5	2	3	33.5	2	1	16646.74500	16651.35518	-4.61018	100.01500
33.5	2	3	34.5	2	1	16644.70300	16647.05224	-2.34924	100.01500
34.5	2	3	35.5	2	1	16641.31700	16642.64765	-1.33065	100.01500
35.5	2	3	36.5	2	1	16637.16100	16638.14123	-0.98023	100.01500
36.5	2	3	37.5	2	1	16632.74100	16633.53277	-0.79177	100.02000
37.5	2	3	38.5	2	1	16628.13000	16628.82208	-0.69208	0.02000
38.5	2	3	39.5	2	1	16623.39000	16624.00897	-0.61897	0.02000
39.5	2	3	40.5	2	1	16618.52900	16619.09325	-0.56425	0.02000
40.5	2	3	41.5	2	1	16613.57800	16614.07474	-0.49674	0.02000
41.5	2	3	42.5	2	1	16608.45300	16608.95326	-0.50026	0.02000
42.5	2	3	43.5	2	1	16603.30000	16603.72862	-0.42862	0.02000

43.5	2	3	44.5	2	1	16598.03200	16598.40064	-0.36864	0.02000
44.5	2	3	45.5	2	1	16592.61600	16592.96913	-0.35313	0.02000
45.5	2	3	46.5	2	1	16587.18400	16587.43392	-0.24992	0.02000
46.5	2	3	47.5	2	1	16581.65900	16581.79482	-0.13582	0.02000
47.5	2	3	48.5	2	1	16575.60500	16576.05165	-0.44665	0.02000
48.5	2	3	49.5	2	1	16569.76400	16570.20422	-0.44022	0.02000
49.5	2	3	50.5	2	1	16563.72700	16564.25234	-0.52534	0.02000
50.5	2	3	51.5	2	1	16557.79400	16558.19584	-0.40184	0.02000
51.5	2	3	52.5	2	1	16551.58600	16552.03452	-0.44852	0.02000
52.5	2	3	53.5	2	1	16545.58400	16545.76819	-0.18419	0.02000
53.5	2	3	54.5	2	1	16539.19400	16539.39665	-0.20265	0.02000
54.5	2	3	55.5	2	1	16532.81400	16532.91972	-0.10572	0.02000
55.5	2	3	56.5	2	1	16526.39200	16526.33719	0.05481	100.02000
33.5	1	4	32.5	1	2	16697.10000	16697.06478	0.03522	0.10000
34.5	1	4	33.5	1	2	16694.15000	16693.87601	0.27399	0.02000
35.5	1	4	34.5	1	2	16690.87100	16690.58426	0.28674	0.02000
36.5	1	4	35.5	1	2	16687.43200	16687.18936	0.24264	0.02000
37.5	1	4	36.5	1	2	16683.94100	16683.69116	0.24984	0.02000
38.5	1	4	37.5	1	2	16680.85000	16680.08948	0.76052	100.02000
39.5	1	4	38.5	1	2	16676.84300	16676.38417	0.45883	0.02000
40.5	1	4	39.5	1	2	16673.04400	16672.57506	0.46894	0.02000
41.5	1	4	40.5	1	2	16669.13800	16668.66198	0.47602	0.02000
42.5	1	4	41.5	1	2	16665.22700	16664.64476	0.58224	0.02000
43.5	1	4	42.5	1	2	16600.00000	16660.52323	-60.52323	100.02000
44.5	1	4	43.5	1	2	16658.72500	16656.29722	2.42778	100.02000
33.5	1	4	34.5	1	2	16648.37100	16648.17693	0.19407	0.02000
34.5	1	4	35.5	1	2	16643.89200	16643.69214	0.19986	0.02000
35.5	1	4	36.5	1	2	16639.33000	16639.10474	0.22526	0.02000
36.5	1	4	37.5	1	2	16634.58900	16634.41456	0.17444	0.02000
37.5	1	4	38.5	1	2	16629.76500	16629.62147	0.14353	0.02000
38.5	1	4	39.5	1	2	16625.36400	16624.72530	0.63870	0.02000
39.5	1	4	40.5	1	2	16620.13000	16619.72590	0.40410	0.02000
40.5	1	4	41.5	1	2	16600.00000	16614.62313	-14.62313	100.02000
41.5	1	4	42.5	1	2	16600.00000	16609.41681	-9.41681	100.02000
42.5	1	4	43.5	1	2	16604.61200	16604.10679	0.50521	0.02000
43.5	1	4	44.5	1	2	16600.00000	16598.69291	1.30709	100.02000
44.5	1	4	45.5	1	2	16695.51000	16593.17501	102.33499	100.02000
45.5	1	4	46.5	1	2	16691.25000	16587.55292	103.69707	100.02000

153 Isotope

J'	Eigenvector	Block	J"	Eigenvector	Block	EXPT	CALC	EXPT-CALC	STD DEV
28.5	4	3	27.5	4	1	16706.43800	16706.42197	0.01603	0.00600
29.5	4	3	28.5	4	1	16704.00600	16704.01240	-0.00640	0.00600
30.5	4	3	29.5	4	1	16701.52000	16701.50179	0.01821	0.00600
31.5	4	3	30.5	4	1	16698.90000	16698.88991	0.01009	0.00600
32.5	4	3	31.5	4	1	16696.18300	16696.17652	0.00648	0.00600
33.5	4	3	32.5	4	1	16693.36200	16693.36140	0.00060	0.00600
34.5	4	3	33.5	4	1	16690.44400	16690.44434	-0.00034	0.00600
35.5	4	3	34.5	4	1	16687.40200	16687.42513	-0.02313	0.00600
36.5	4	3	35.5	4	1	16684.26900	16684.30359	-0.03459	0.00600
37.5	4	3	36.5	4	1	16680.99500	16681.07954	-0.08454	0.00600
38.5	4	3	37.5	4	1	16677.55500	16677.75278	-0.19778	0.00600
39.5	4	3	38.5	4	1	16674.28700	16674.32314	-0.03614	0.00600
40.5	4	3	39.5	4	1	16670.92600	16670.79046	0.13554	0.00600
41.5	4	3	40.5	4	1	16667.34000	16667.15455	0.18545	0.00600
42.5	4	3	41.5	4	1	16663.49000	16663.41525	0.07475	0.00600
43.5	4	3	42.5	4	1	16659.52300	16659.57239	-0.04939	0.00600
44.5	4	3	43.5	4	1	16655.55400	16655.62580	-0.07180	0.00600
45.5	4	3	44.5	4	1	16651.61500	16651.57531	0.03969	0.00600
46.5	4	3	45.5	4	1	16647.55200	16647.42076	0.13124	0.00600
47.5	4	3	46.5	4	1	16643.39100	16643.16198	0.22902	100.00599
48.5	4	3	47.5	4	1	16638.94600	16638.79880	0.14720	0.00600
49.5	4	3	48.5	4	1	16634.24100	16634.33105	-0.09005	0.00600
50.5	4	3	49.5	4	1	16629.72200	16629.75856	-0.03656	0.00600
51.5	4	3	50.5	4	1	16625.05300	16625.08116	-0.02816	0.00600
52.5	4	3	51.5	4	1	16620.23100	16620.29867	-0.06767	0.00600
53.5	4	3	52.5	4	1	16615.40000	16615.41092	-0.01092	0.00600
54.5	4	3	53.5	4	1	16610.44800	16610.41774	0.03026	0.00600
55.5	4	3	54.5	4	1	16605.38200	16605.31894	0.06306	0.00600
56.5	4	3	55.5	4	1	16600.10800	16600.11434	-0.00634	0.00600
57.5	4	3	56.5	4	1	16594.79900	16594.80377	-0.00477	0.00600
58.5	4	3	57.5	4	1	16589.33800	16589.38704	-0.04904	0.00600
15.5	4	3	16.5	4	1	16706.88300	16707.17258	-0.28958	100.00599
16.5	4	3	17.5	4	1	16704.53000	16704.73080	-0.20080	100.00599
17.5	4	3	18.5	4	1	16702.06200	16702.19667	-0.13467	100.00599
18.5	4	3	19.5	4	1	16699.48000	16699.56871	-0.08871	100.00599
19.5	4	3	20.5	4	1	16600.00000	16696.84573	-96.84573	100.00599
20.5	4	3	21.5	4	1	16693.87600	16694.02675	-0.15075	0.00600
21.5	4	3	22.5	4	1	16600.00000	16691.11092	-91.11092	100.00599
22.5	4	3	23.5	4	1	16688.17000	16688.09756	0.07244	0.00600
23.5	4	3	24.5	4	1	16685.03000	16684.98607	0.04393	0.00600
24.5	4	3	25.5	4	1	16681.82900	16681.77594	0.05306	0.00600
25.5	4	3	26.5	4	1	16678.50000	16678.46671	0.03329	0.00600
26.5	4	3	27.5	4	1	16675.08300	16675.05801	0.02499	0.00600
27.5	4	3	28.5	4	1	16671.57700	16671.54947	0.02753	0.00600
28.5	4	3	29.5	4	1	16667.94500	16667.94079	0.00421	0.00600
29.5	4	3	30.5	4	1	16664.22900	16664.23169	-0.00269	0.00600

30.5	4	3	31.5	4	1	16660.44400	16660.42191	0.02209	0.00600
31.5	4	3	32.5	4	1	16656.53100	16656.51121	0.01979	0.00600
32.5	4	3	33.5	4	1	16652.51400	16652.49936	0.01464	0.00600
33.5	4	3	34.5	4	1	16648.38800	16648.38617	0.00183	0.00600
34.5	4	3	35.5	4	1	16644.17500	16644.17143	0.00357	0.00600
35.5	4	3	36.5	4	1	16639.83500	16639.85496	-0.01996	0.00600
36.5	4	3	37.5	4	1	16635.51000	16635.43658	0.07342	0.00600
37.5	4	3	38.5	4	1	16630.83800	16630.91612	-0.07812	0.00600
38.5	4	3	39.5	4	1	16626.37300	16626.29340	0.07960	0.00600
39.5	4	3	40.5	4	1	16621.78400	16621.56826	0.21574	100.00599
40.5	4	3	41.5	4	1	16616.87900	16616.74054	0.13846	0.00600
41.5	4	3	42.5	4	1	16600.00000	16611.81008	-11.81008	100.00599
42.5	4	3	43.5	4	1	16606.86800	16606.77673	0.09127	0.00600
43.5	4	3	44.5	4	1	16601.66200	16601.64032	0.02168	0.00600
44.5	4	3	45.5	4	1	16596.33000	16596.40071	-0.07071	0.00600
45.5	4	3	46.5	4	1	16591.11100	16591.05773	0.05327	0.00600
46.5	4	3	47.5	4	1	16585.79100	16585.61123	0.17977	100.00599
47.5	4	3	48.5	4	1	16600.00000	16580.06105	19.93895	100.00599
48.5	4	3	49.5	4	1	16600.00000	16574.40704	25.59296	100.00599
49.5	4	3	50.5	4	1	16568.58000	16568.64905	-0.06905	0.00600
50.5	4	3	51.5	4	1	16562.77000	16562.78690	-0.01690	0.00600
51.5	4	3	52.5	4	1	16556.71100	16556.82045	-0.10945	100.00599
52.5	4	3	53.5	4	1	16550.70500	16550.74952	-0.04452	0.00600
53.5	4	3	54.5	4	1	16544.58400	16544.57397	0.01003	0.00600
54.5	4	3	55.5	4	1	16538.34800	16538.29362	0.05438	0.00600
55.5	4	3	56.5	4	1	16531.99300	16531.90830	0.08470	0.00600
56.5	4	3	57.5	4	1	16525.43900	16525.41785	0.02115	0.00600
57.5	4	3	58.5	4	1	16518.84500	16518.82210	0.02290	0.00600
58.5	4	3	59.5	4	1	16512.02700	16512.12087	-0.09387	0.00600
23.5	4	4	22.5	4	2	16717.78700	16717.74642	0.04058	0.00600
24.5	4	4	23.5	4	2	16715.98100	16715.95235	0.02865	0.00600
25.5	4	4	24.5	4	2	16714.07900	16714.05779	0.02121	0.00600
26.5	4	4	25.5	4	2	16712.07600	16712.06241	0.01359	0.00600
27.5	4	4	26.5	4	2	16709.98000	16709.96590	0.01410	0.00600
28.5	4	4	27.5	4	2	16707.80000	16707.76801	0.03199	0.00600
29.5	4	4	28.5	4	2	16705.47900	16705.46851	0.01049	0.00600
30.5	4	4	29.5	4	2	16703.08600	16703.06719	0.01881	0.00600
31.5	4	4	30.5	4	2	16700.54300	16700.56385	-0.02085	0.00600
32.5	4	4	31.5	4	2	16697.93300	16697.95833	-0.02533	0.00600
33.5	4	4	32.5	4	2	16695.22000	16695.25044	-0.03044	0.00600
34.5	4	4	33.5	4	2	16692.40100	16692.44003	-0.03903	0.00600
35.5	4	4	34.5	4	2	16689.47600	16689.52694	-0.05094	0.00600
36.5	4	4	35.5	4	2	16686.45100	16686.51104	-0.06004	0.00600
37.5	4	4	36.5	4	2	16683.29100	16683.39215	-0.10115	100.00599
38.5	4	4	37.5	4	2	16680.20600	16680.17016	0.03584	0.00600
39.5	4	4	38.5	4	2	16676.91600	16676.84491	0.07109	0.00600
40.5	4	4	39.5	4	2	16673.47500	16673.41626	0.05874	0.00600

41.5	4	4	40.5	4	2	16669.95400	16669.88408	0.06992	0.00600
42.5	4	4	41.5	4	2	16666.27800	16666.24821	0.02979	0.00600
43.5	4	4	42.5	4	2	16662.53200	16662.50852	0.02348	0.00600
44.5	4	4	43.5	4	2	16658.69500	16658.66487	0.03013	0.00600
45.5	4	4	44.5	4	2	16654.65400	16654.71711	-0.06311	0.00600
46.5	4	4	45.5	4	2	16650.68700	16650.66510	0.02190	0.00600
47.5	4	4	46.5	4	2	16646.58000	16646.50869	0.07131	0.00600
48.5	4	4	47.5	4	2	16642.41300	16642.24772	0.16528	100.00599
49.5	4	4	48.5	4	2	16637.56700	16637.88206	-0.31506	100.00599
50.5	4	4	49.5	4	2	16633.32500	16633.41155	-0.08655	0.00600
51.5	4	4	50.5	4	2	16628.80700	16628.83602	-0.02902	0.00600
52.5	4	4	51.5	4	2	16624.13500	16624.15533	-0.02033	0.00600
53.5	4	4	52.5	4	2	16619.29400	16619.36931	-0.07531	0.00600
54.5	4	4	53.5	4	2	16614.43600	16614.47781	-0.04181	0.00600
55.5	4	4	54.5	4	2	16609.53500	16609.48065	0.05435	0.00600
56.5	4	4	55.5	4	2	16604.50100	16604.37766	0.12334	0.00600
57.5	4	4	56.5	4	2	16599.20800	16599.16869	0.03931	0.00600
58.5	4	4	57.5	4	2	16593.92400	16593.85355	0.07045	0.00600
59.5	4	4	58.5	4	2	16588.37700	16588.43207	-0.05507	0.00600
16.5	4	4	17.5	4	2	16705.94900	16706.05483	-0.10583	100.00599
17.5	4	4	18.5	4	2	16703.55900	16703.64295	-0.08395	100.00599
18.5	4	4	19.5	4	2	16701.08000	16701.13615	-0.05615	100.00599
19.5	4	4	20.5	4	2	16698.49200	16698.53316	-0.04116	0.00600
20.5	4	4	21.5	4	2	16695.77500	16695.83294	-0.05794	0.00600
21.5	4	4	22.5	4	2	16693.00200	16693.03466	-0.03266	0.00600
22.5	4	4	23.5	4	2	16690.18900	16690.13765	0.05135	0.00600
23.5	4	4	24.5	4	2	16687.15500	16687.14134	0.01366	0.00600
24.5	4	4	25.5	4	2	16684.05000	16684.04527	0.00473	0.00600
25.5	4	4	26.5	4	2	16680.84500	16680.84905	-0.00405	0.00600
26.5	4	4	27.5	4	2	16677.55200	16677.55233	-0.00033	0.00600
27.5	4	4	28.5	4	2	16674.15600	16674.15485	0.00115	0.00600
28.5	4	4	29.5	4	2	16670.66800	16670.65634	0.01166	0.00600
29.5	4	4	30.5	4	2	16667.05400	16667.05657	-0.00257	0.00600
30.5	4	4	31.5	4	2	16663.33600	16663.35536	-0.01936	0.00600
31.5	4	4	32.5	4	2	16659.52400	16659.55251	-0.02851	0.00600
32.5	4	4	33.5	4	2	16655.61700	16655.64786	-0.03086	0.00600
33.5	4	4	34.5	4	2	16651.61400	16651.64124	-0.02724	0.00600
34.5	4	4	35.5	4	2	16647.49700	16647.53251	-0.03551	0.00600
35.5	4	4	36.5	4	2	16643.26800	16643.32152	-0.05352	0.00600
36.5	4	4	37.5	4	2	16638.93800	16639.00813	-0.07013	0.00600
37.5	4	4	38.5	4	2	16634.48700	16634.59221	-0.10521	0.00600
38.5	4	4	39.5	4	2	16630.08700	16630.07362	0.01338	0.00600
39.5	4	4	40.5	4	2	16625.51100	16625.45224	0.05876	0.00600
40.5	4	4	41.5	4	2	16620.78700	16620.72794	0.05906	0.00600
41.5	4	4	42.5	4	2	16615.97700	16615.90057	0.07643	0.00600
42.5	4	4	43.5	4	2	16611.11100	16610.97003	0.14097	100.00599
43.5	4	4	44.5	4	2	16605.97200	16605.93616	0.03584	0.00600

44.5	4	4	45.5	4	2	16600.00000	16600.79885	-0.79885	100.00599
45.5	4	4	46.5	4	2	16595.51000	16595.55795	-0.04795	0.00600
46.5	4	4	47.5	4	2	16590.25000	16590.21335	0.03665	0.00600
47.5	4	4	48.5	4	2	16584.84400	16584.76489	0.07911	0.00600
48.5	4	4	49.5	4	2	16579.38300	16579.21244	0.17056	100.00599
49.5	4	4	50.5	4	2	16573.41800	16573.55587	-0.13787	0.00600
50.5	4	4	51.5	4	2	16567.74900	16567.79503	-0.04603	0.00600
51.5	4	4	52.5	4	2	16561.93200	16561.92977	0.00223	0.00600
52.5	4	4	53.5	4	2	16555.97600	16555.95996	0.01604	0.00600
53.5	4	4	54.5	4	2	16549.86200	16549.88544	-0.02344	0.00600
54.5	4	4	55.5	4	2	16543.70200	16543.70606	-0.00406	0.00600
55.5	4	4	56.5	4	2	16537.52500	16537.42168	0.10332	0.00600
56.5	4	4	57.5	4	2	16531.20500	16531.03212	0.17288	100.00599
57.5	4	4	58.5	4	2	16524.63800	16524.53724	0.10076	0.00600
58.5	4	4	59.5	4	2	16518.06800	16517.93687	0.13113	100.00599
59.5	4	4	60.5	4	2	16511.17900	16511.23085	-0.05185	0.00600

Table 6-IX: Calculated vs. Experimental Centrifugal Distortion Constants

Isotope	Experimental	Calculated
151	1.88×10^{-7}	3.00×10^{-7}
153	2.48×10^{-7}	3.00×10^{-7}

calculation in the previous paragraph cannot be done to check the value of D'. However I can use the B' and D' values to estimate an ω' value. The closeness of the D values for the two isotopes provides some reassurance that they are not drastically affected by the perturbation. The calculation gives an ω_e' of about 500 cm^{-1}. This is similar to upper state ω_e values found in the spectrum of YbO.

A disturbing result of the fit is that the B value for the unperturbed lower state is smaller in the 151 isotope than in the 153, the opposite of what it should be. One apparent solution would be to switch the assignment of the two isotopes. However the assignment was made by noting that the width of one of the J = N+1/2 components was slightly larger than the other. The hyperfine should be a more reliable label since the B'' values could include some effects of perturbations in the upper state as well as slightly shifted values for the J = N-1/2 lines of the 151 isotope mentioned above.

As I demonstrated in Section 6.3, the spin-spin constant can be accounted for by a second order spin-orbit perturbation. Second order spin-orbit perturbations can also mimic the spin rotation interaction. In the ion, there is a ^6P level $\sim$28,000 cm^{-1} above the ground ^8S state; both come from the f^7 configuration. The combination of the ^6P state on the Eu^{2+} ion with the $^1\Sigma$ state on the O^{2-} will give the molecular states $^6\Pi$ and $^6\Sigma$. Using the angular momentum techniques already discussed, a Hamiltonian was written which included the second order spin-orbit perturbation between the ground $^8\Sigma$ state

and the $^6\Pi$ and $^6\Sigma$ states. Second order perturbation predicts a constant with a value of $(\varsigma)^2/\Delta E$. Using the ionic value of 1180 cm^{-1} for ς and 28,000 cm^{-1} forΔE [14], the spin-orbit constant from the fit should be about -50 cm^{-1}. When both the spin-spin and spin rotation constants of the lower state are held to zero and the spin-orbit interaction is fit with a single constant, the fit is worse by a factor of 1.4 and the fitted constant is -6 cm^{-1}. If the data is then fit with the spin-orbit perturbation constant held at -6 cm^{-1} and the spin-spin and spin rotation constants of the lower state allowed to vary, the fitted constants are very small. These fits indicate that the apparent spin-spin and spin rotation constants could come from a second order spin-orbit perturbation. The difference between the predicted -50 cm^{-1} and the observed -6 cm^{-1} could be due to several causes. The spin-spin interaction may well account for part of the spin-spin constant such that the observed constant is a combination of the spin-spin and spin-orbit interactions. The other possibility is that the simple picture of a pure upper state may not be correct. If the ligand field matrix elements between configurations were taken into account, the predicted constant could match the observed constant more closely.

A possible configuration for the upper state is the charge transfer state $Eu^+(4f^7\sigma)O^-(\pi^4\sigma)$. The Eu^+ $4f^7\sigma$ configuration will produce $^9\Sigma$ and $^7\Sigma$ states, while the O^- $\pi^4\sigma$ will give a $^1\Sigma$ state. These two will combine to give one $^{10}\Sigma$, two $^8\Sigma$ states and a $^6\Sigma$ state. The states will be fairly close in energy and can perturb each other. The O^- will also have a $\pi^3\sigma^2$ configuration, which produces $^{10}\Pi$, $^8\Pi$ and $^6\Pi$ states. The difference in energy between these states and those of the $\pi^4\sigma$ configuration is the difference in energy of a π-hole and a σ hole on the O^-. Because the π orbital will keep the positively

charged hole farther away from the Eu^+ than the σ orbital, the $\pi^3\sigma^2$ will lie lower in energy. The π and σ orbitals on the O^- should be quite large, thus the difference in energy between the π and σ holes due to +1 charge of the Eu^+ ion should be significant.

The spin-spin constants support this description of the upper state. I've shown that the spin-spin constant of the lower state probably results in part from a spin-orbit interaction with a state 28,000 cm^{-1} higher in energy. Therefore the small magnitude of the spin-spin constant in the upper state, on the order of the constant of the lower state, is surprising since the density of states should be quite high for the upper state. However the small magnitude can be explained if the Σ upper state is due to the charge transfer configuration. The π hole should have significantly lower energy than the σ hole so the $^{10,8,6}\Pi$ states will not interact strongly with the $^{10,8,6}\Sigma$ states because of the large energy difference. Furthermore the states from $Eu^+(4f^7\sigma)O^-(\pi^4\sigma)$ will not interact strongly with the Π states from another possible upper state configuration, $Eu^{2+}(4f^6\pi)O^{2-}(\pi^4\sigma^2)$, since the configurations differ by two electrons and spin-orbit is a one electron operator. Thus the charge transfer Σ states should be well isolated.

6.8.2 Agreement With Ligand Field Prediction

The Ligand Field Model predicted an isolated $^+\Sigma^-$ ground state coming from the $Eu^+(4f^7)$ configuration. The lower state of the transition is an $^8\Sigma$ state. However I cannot state that its parent configuration is the $4f^7$ because I was unable to determine the hyperfine splitting of the lower state. A very small splitting would confirm the $4f^7$ configuration prediction. The lower state

shows no signs of a perturbation except for the non-zero spin-spin constant. As I have discussed, this could be due to charge transfer states.

6.9 Conclusion

The transtion is $8\Sigma^- - 8\Sigma^-$. The lower state is not strongly perturbed but its non-zero spin-spin constant suggests that it is perturbed weakly by distant Π states. Unfortunately it is not possible to determine the hyperfine splitting of the lower state so the $4f^{14}$ assignment cannot be proved by a lack of hyperfine splitting. However the vibrational constant of 688 cm^{-1} suggests that the lower state comes from the $4f^7$ configuration. The upper $^8\Sigma^-$ state is perturbed to the extent that lines are missing from their predicted positions. The small B value, along with the evidence for a small vibrational constant, suggest that the upper state comes from the charge transfer configuration $Eu^+(4f^7\sigma)O^-(\pi^4\sigma)$. The reasons why an ω_e value of about 500 cm^{-1} indicate a charge transfer state have been discussed in Chapter 4. The fairly small spin-spin constant also supports this conclusion.

There are additional experiments and calculations which should be done on EuO. Some of the other branches of the 151 isotope should be found. This should be fairly straightforward, although time consuming, using the predicted line positions from the fit. A complete ligand field calculation which includes the ligand field configuration interactions should be done on EuO. This could provide valuable clues to the identity of the upper state and to its perturber. It may also explain the discrepancy between the fitted and calculated spin-orbit constant.

References

[1] P. W. Atkins.
Molecular Quantum Mechanics.
Oxford University Press, New York, 1983.

[2] J. M. Brown, J. T. Hougen, K. P. Huber, J. W. C. Johns, I. Kopp, H Lefebvre-Brion, A. J. Merer, D. A. Ramsay, J. Rostas and R. N. Zare.
The Labeling of Parity Doublet Levels in Linear Molecules.
J. Mol. Spec. 55:500-503, 1975.

[3] J. M. Brown and B. J. Howard.
An Approach to the Anomalous Commutation Relations of Rotational Angular Momenta in Molecules.
Mol. Phys. 31(5):1517, 1976.

[4] A. Carrington, D. H. Levy and T. A. Miller.
Electron Resonance of Gaseous Diatomic Molecules.
Adv. Chem. Phys. 18:149, 1970.

[5] A. Carrington, D. H. Levy and T. A. Miller.
Electron Resonance of Gaseous Diatomic Molecules.
Adv. Chem. Phys. 18:196, 1970.
eq. 3.64.

[6] A. R. Edmonds.
Angular Momentum in Quantum Mechanics.
Princeton University Press, Princeton, New Jersey, 1974.

[7] A. R. Edmonds.
Angular Momentum in Quantum Mechanics.
Princeton University Press, Princeton, New Jersey, 1974.
page 60; eq. 4.2.7.

[8] S. D. Gabelnick, G. T. Reedy and M. G. Chasanov.
Infrared Spectra and Structrue of some Matrix-Isolated Lanthanide and Actinide Oxides.
J. Chem. Phys. 60:1167, 1974.

[9] G. Herzberg.
Molecular Spectra and Molecular Structrue.
Van Nostrand Reinhold Company, New York, 1950.
page 103.

[10] J. T. Hougen.
The Calculation of Rotational Energy Levels and Rotational Line Intensities in Diatomic Molecules.
NBS Monograph 115, National Bureau of Standards, June, 1970.

[11] D. W. Marquardt.
An Algorithm for Least-Squares Estimation of Nonlinear Parameters.
J. Soc. Ind. Appl. Math 11(2):431-441, 1963.

[12] B. L. Silver.
Irreducible Tensor Methods.
Academic Press, New York, 1976.
page 71.

[13] B. L. Silver.
Irreducible Tensor Methods.
Academic Press, New York, 1976.
page 92.

[14] J. Sugar and N. Spector.
Spectrum and Energy Levels of Doubly Ionized Europium (Eu III).
Journal of the Optical Society of America 64:1484, 1974.

Chapter 7
Ytterbium Oxide

7.1 Introduction

The electronic spectrum of Ytterbium Oxide provides an excellent testing ground for the predictions of Ligand Field Theory. The 4f orbitals of YbO differ significantly in the number of electrons from those of CeO, where the ligand field model has been shown to work, and yet the model for the two molecules is almost identical. Thus YbO should indicate whether the ligand field approach is indeed a good global model for the rare-earth oxides, or if its success with CeO and PrO was simply a fortuitous accident. As was discussed in Chapter 5, YbO should have a $Yb^{2+}(4f^{14})\ O^{2-}(\pi^4\sigma^2)$ ground state configuration with $Yb^{2+}(4f^{13}6s)\ O^{2-}(\pi^4\sigma^2)$ somewhat higher in energy, so there is a chance of finding the relative energies of the two low lying configurations. Futhermore, the 10 lowest lying states predicted to originate from these configurations are enough to make a distintive pattern yet sufficiently small in number that it was possible to obtain and assign an adequate sample of the low lying states in a reasonable amount of time.

Two types of spectroscopy were used, single mode dye laser excitation scans with wavelength selective detection and fluorescence scans taken with a monochromator while exciting with the single mode dye laser. A full rotational analysis was done on five of the eleven excitation bands, but the other six were found to be too complicated due to the five isotopes of

observable abundance coupled with perturbations in both the upper and lower states. Analyses of several other bands were abandoned early in the data collection process as being too obviously pathalogical. A full rotational analysis produces only one number of interest, the Ω value, or name, of the lower state in the transition. This name then provides a key to the fluorescence spectrum. By using the rules for the relative intensity of the branches in fluorescence, $I_Q > I_R > I_P$ if $\Delta\Omega = +1$, $I_R = I_P >> I_Q$ if $\Delta\Omega = 0$, $I_Q > I_P > I_R$ if $\Delta\Omega = -1$, and the Ω value of the state involved in the laser-excited transition, each fluorescence spectrum corresponded to a piece of the energy level diagram, complete with separations between and names of the states. If the relative intensities of the branches in the fluorescence spectra were changed by a J dependent perturbation, the relative intensity rule could not be used, so that other methods of Ω assignment were necessary. These methods were based on the fact that $\Omega = 0$ states have only one parity, so that certain perturbations and transitions had to involve $\Omega = 0$ states. In all cases the spectra had to be interpreted with care because the large number of overlapping transitions could lead to fluorescence from several upper states for a given excitation, thus resulting in false levels in an energy level diagram. Spurious linkages were detected by recording several spectra from the same upper level while exciting from different lower levels, for example exciting the R(10), Q(11) and P(12) lines, and then discarding fluorescence lines which did not appear in all the spectra. In this way, a perturbation facilitated linkage, based on extensive mixing in a single J' level, could be relied upon.

7.2 YbO Hamiltonian

In a Hund's case c molecule the useful quantum numbers are:

$\mathbf{J_a} = \mathbf{L} + \mathbf{S}$, defined in the molecule fixed coordinate system;

$\mathbf{J} = \mathbf{J_a} + \mathbf{R}$, the total angular momentum excluding nuclear spin;

Ω = projection of both $\mathbf{J_a}$ and $\mathbf{J}$ onto the internuclear axis;

$\mathbf{I}$ = nuclear spin angular momentum, defined in the lab;

$\mathbf{F} = \mathbf{J} + \mathbf{I}$, the total angular momentum.

In addition, for YbO, the following angular momenta prove to be useful.

$\mathbf{J_f} = \Sigma\, \mathbf{j_f}$, the sum of the electronic spin and orbital angular momenta of the f electrons.

$\mathbf{j_s}$ = total angular momentum of the single s electron = 1/2

7.2.1 Rotational Hamiltonian

The rotational Hamiltonian is $B\mathbf{R}^2$.

$$\mathbf{R} = \mathbf{J} - \mathbf{J_a}$$

$$B\mathbf{R}^2 = B[\mathbf{J}^2 + \mathbf{J_a}^2 - 2\mathbf{J}\cdot\mathbf{J_a}]$$

$$\langle J_a' J' \Omega' |\, B[\mathbf{J}^2 + \mathbf{J_a}^2]\, | J_a J \Omega \rangle = \delta_{J_a' J_a} \delta_{J' J} \delta_{\Omega' \Omega}\, [BJ(J+1) + BJ_a(J_a+1)]$$

$$\mathbf{J}\cdot\mathbf{J}_a = \Sigma_p \; T^1_p(\mathbf{J}) \; T^1_{-p}(\mathbf{J}_a)$$

$$= \Sigma_{p,q} \; T^1_p(\mathbf{J}) \; D^{1\,*}_{-pq} \; T^1_q(\mathbf{J}_a)$$

where p refers to the projections of the angular momenta on a space fixed axis and q to the molecule fixed axis. The rotation operator is necessary because the angular momenta that $\mathbf{J}_a$ operates on are defined in molecular coordinates. The matrix element of $\mathbf{J}\cdot\mathbf{J}_a$ becomes

$$\langle J_a' J'\Omega' M' |\; \mathbf{J}\cdot\mathbf{J}_a \;| J_a J\Omega M\rangle =$$

$$\Sigma_{p,q} \langle J'\Omega' M' |\; T^1_p(\mathbf{J}) \; D^{1\,*}_{-pq} \;| J\Omega M\rangle \langle J_a'\Omega' |\; T^1_q(\mathbf{J}_a) \;| J_a\Omega\rangle. \qquad (7.1)$$

The matrix element of $\mathbf{J}_a$ can be evaluated immediately with the Wigner-Eckart theorem

$$\langle J_a'\Omega' |\; T^1_q(\mathbf{J}_a) \;| J_a\Omega\rangle =$$

$$(-1)^{J_a'-\Omega'} \; [J_a(J_a+1)(2J_a+1)]^{1/2} \quad \begin{pmatrix} J_a' & 1 & J_a \\ -\Omega' & q & \Omega \end{pmatrix} \qquad (7.2)$$

The matrix element involving the rotation operator is given by [2]

$$\Sigma_q \langle J'\Omega' M' |\; T^1_p(\mathbf{J}) \; D^{1\,*}_{-pq} \;| J\Omega M\rangle = (-1)^{J'-\Omega'} \begin{pmatrix} J & 1 & J \\ -\Omega' & q & \Omega \end{pmatrix}$$

$$[J(J+1)(2J+1)]^{1/2} \; \delta_{JJ'} \; \delta_{MM'}. \qquad (7.3)$$

With equations (7.2) and (7.3) equation (7.1) becomes

$$\langle J_a' J'\Omega' M' |\; \mathbf{J}\cdot\mathbf{J}_a \;| J_a J\Omega M\rangle = \Sigma_q \; (-1)^{J+J_a'-2\Omega'} \begin{pmatrix} J & 1 & J \\ -\Omega' & q & \Omega \end{pmatrix}$$

$$\begin{pmatrix} J_a & 1 & J_a \\ -\Omega' & q & \Omega \end{pmatrix} \quad [J(J+1)(2J+1)]^{1/2} \quad [J_a(J_a+1)(2J_a+1)]^{1/2} \quad \delta_{JJ'}\, \delta_{MM'}. \quad (7.4)$$

The 3-j symbols must be evaluated for the possible values of q.

$\underline{q = 0}$

$$\begin{pmatrix} J & 1 & J \\ -\Omega & 0 & \Omega \end{pmatrix} = (-1)^{J-\Omega}\, \Omega/[J(J+1)(2J+1)]^{1/2}$$

$$\begin{pmatrix} J_a & 1 & J_a \\ -\Omega & 0 & \Omega \end{pmatrix} \quad (-1)^{J_a-\Omega}\, \Omega/[J_a(J_a+1)(2J_a+1)]^{1/2}$$

These values for the 3-j symbols in equation (7.4) give

$$<J_a J\Omega M|\ 2B\mathbf{J}\cdot\mathbf{J}_a\ |J_a J\Omega M> = 2B\Omega^2.$$

This term, as well as $BJ_a(J_a+1)$, has no dependence on J. For a given state, these terms will be constants, and will be included in the electronic energy of the state. For all practical purposes it will be dropped from the Hamiltonian.

Next evaluating the off-diagonal elements: $\underline{q = 1}$

$$\begin{pmatrix} J & 1 & J \\ -\Omega-1 & 1 & \Omega \end{pmatrix} = (-1)^{J-\Omega}\, ([(J-\Omega)(J+\Omega+1)]/[2J(J+1)(2J+1)])^{1/2}$$

$$\begin{pmatrix} J_a & 1 & J_a \\ -\Omega-1 & 1 & \Omega \end{pmatrix} = (-1)^{J_a-\Omega}\, ([(J_a-\Omega)(J_a+\Omega+1)]/[2J_a(J_a+1)(2J_a+1)])^{1/2}.$$

Putting these values into equation (7.4) yields

$$<J_a J\Omega+1\ M|\ 2B\mathbf{J}\cdot\mathbf{J}_a\ |J_a J\Omega M> =$$

$$B[(J-\Omega)(J+\Omega+1)(J_a-\Omega)(J_a+\Omega+1)]^{1/2}.$$

7.2.2 Centrifugal Distortion

As shown in the derivation of the EuO Hamiltonian, the centrifugal distortion interaction can be fairly complicated if perturbing states are nearby. However I will use the zero order approximation of

$$H_{cd} = DR^4.$$

The diagonal element which depends on J is $-DJ^4$, or $-D[J(J+1)]^2$.

7.2.3 Hyperfine Hamiltonian

The hyperfine Hamiltonian is given by [3]

$$H_{hf} = aT^1(\mathbf{I})\cdot T^1(\mathbf{L}) + bT^I(\mathbf{I})\cdot T^1(\mathbf{S}) + cT^1(\mathbf{I})\cdot T^1(\mathbf{S},C^2)$$

where $T^1(\mathbf{I})\cdot T^1(\mathbf{S},C^2)$ is the second rank tensor formed from $T^1(\mathbf{S})$ and C^2.

Since **I**, **L**, **S** and C^2 are all defined in molecular coordinates, I do not have to worry about mixing laboratory and molecular coordinate systems. In fact I will show that all the matrix elements have the same form. Let $T^1(P)$ represent any of the operators dotted into $T^1(\mathbf{I})$. Then [2]

$$\langle J_a'J'\Omega'I'F'|\ T^1(\mathbf{I})\cdot T^1(\mathbf{P})\ |J_aJ\Omega IF\rangle = (-1)^{I'+J+F} \begin{Bmatrix} I & J & F \\ J' & I' & 1 \end{Bmatrix}$$

$$\langle I'||T^1(\mathbf{I})||I\rangle \quad \langle J_a'J'\Omega'||T^1(\mathbf{P})||J_aJ\Omega\rangle$$

and

$$\langle J_a' J'\Omega' || T^1(\mathbf{P}) || J_a J\Omega \rangle = \langle J_a' \Omega' || T^1(\mathbf{P}) || J_a \Omega \rangle .$$

That is really as far as I can go with $\langle J_a'\Omega' || T^1(\mathbf{P}) || J_a\Omega \rangle$ without knowing more about **L** or **S**. Of course, in a case c molecule, I do not usually know **L** or **S**, only their vector sum $\mathbf{J_a}$. The important point is that this reduced matrix element is independent of J and F, so for a given state it will be just another constant. This will be the case for $T^1(\mathbf{P}) = T^1(\mathbf{L})$, $T^1(\mathbf{S})$ or $T^1(\mathbf{S},C^2)$. Therefore all I can say about the hyperfine is

$$\langle |H_{hf}| \rangle = k\ (-1)^{I+J+F} \begin{Bmatrix} I & J & F \\ J' & I & 1 \end{Bmatrix} [I(I+1)(2I+1)]^{1/2}$$

where k is a constant. Now to evaluate the 6-j symbol.

<u>J' = J</u>

$$\begin{Bmatrix} I & J & F \\ J & I & 1 \end{Bmatrix} = (-1)^{F+I+J+1}\ 2[I(I+1) + J(J+1) - F(F+1)] \div$$

$$4[I(I+1)(2I+1)J(J+1)(2J+1)]^{1/2}$$

Thus the diagonal hyperfine interaction is given by

$$\langle J_a J\Omega IF | H_{hf} | J_a J\Omega IF \rangle = k\ [F(F+1) - I(I+1) + J(J+1)] \div$$

$$2[J(J+1)(2J+1)]^{1/2}. \qquad (7.5)$$

<u>J' = J+1</u>

$$\begin{Bmatrix} I & J & F \\ J+1 & I & 1 \end{Bmatrix} = (-1)^{F+I+J+1}\ [(F+I+J+2)(-F+I+J+1)(F-I+J+1)(F+I-$$

$$J-1)(F+I-J)]^{1/2} \div 2[(J+1)(2J+1)(2J+3)I(I+1)(2I+1)]^{1/2}$$

This gives an off-diagonal matrix element of

$$<J_a J+1\ \Omega I F|H_{hf}|J_a J \Omega I F> = -k/2\ [(F+I+J+2)(-F+I+J+1)(F-I+J+1)$$

$$(F+I-J-1)(F+I-J)]^{1/2} \div [(J+1)(2J+1)(2J+3)]^{1/2}. \quad (7.6)$$

7.3 5793, 6032 and 6067 Å Bands

7.3.1 Survey

Visible fluorescence was observed with each of the argon ion laser lines. The most intense fluorescence, which was excited at 5017 Å, consisted of three fluorescence bands, each with the same RP separation, whose relative lower state energies were ~0, 910 and 1010 cm^{-1} respectively. Similar separations were observed in resolved fluorescence excited at 4965 and 5145 Å. The 5145 Å spectrum showed a state 676 cm^{-1} above the lowest state, while the spectrum excited at 4880 Å showed the beginning of a progression with separations of 676 and 656 cm^{-1}. Each of the laser lines seemed to excite several transitions. Because of the complexity of the spectra, we were unable to detect a sufficient number of energy linkages to enable us to build a comprehensive energy diagram. For each transition observed, spectra were obtained at only one value of J which was not sufficient to enable us to assign Ω values to the states.

An excitation spectrum, taken in the rhodamine 6G region with the Coherent 599 dye laser operating broadband and with O_2 as oxidant, showed three intense bands at 5793, 6032 and 6067 Å. When N_2O was the oxidant, many more bands were observed but the three mentioned above were more

intense by about an order of magnitude. Resolved fluorescence spectra were taken with the laser operating single mode, exciting transitions in the 6032 and 6067 Å bands. In each case, long vibrational progressions, showing the same vibrational frequencies, were observed. The two bands, therefore, involve transitions to the same lower electronic state. The first two members of the progression excited at 6032 Å were at ~5793 and 6032 Å. The 5793 and 6032 Å bands are thus transitions from a common upper level to $v = 0$ and 1 of the same lower state. The 6067 Å band was the first member of its progression and is therefore a transition from a different upper state to $v = 0$ of the lower state. The three bands at 5793, 6032 and 6067 Å have thus been assiged as $(v_A,0)$, $(v_A,1)$ and $(v_B,0)$ respectively, where v_A and v_B are vibrational levels of two different electronic states. The vibrational spacings within the progressions are consistent with the lower state assignments. The observed frequencies of the fluorescence lines and the vibrational spacings are listed in Table 7-I. As the measurements were taken from rapid spectrometer scans and are uncalibrated, the accuracy of the vibrational spacings is probably ~5 cm^{-1}. However, this is sufficient to show that the vibrational frequency, ω_e, of the lower state is near 680 cm^{-1}. The vibrational spacings agree with those observed with the argon ion laser, indicating that the lower state of the three bands is probably the lowest energy electronic state observed in the argon laser spectra. The following features of the dye laser fluorescence spectra are worthy of note:

1. In all spectra, the most intense transitions are those to $v'' = 0$, 1, 5 and 6. On the basis of Franck-Condon arguments, this strongly suggests that both upper electronic states are vibrationally excited and have significantly smaller B values than the lower state.

2. The vibrational frequencies listed in Table 7-I decrease rapidly with

increasing vibration up to $v = 4$, after which the rate of decrease is much less rapid. This represents a sudden decrease in the anharmonicity, from ~ 8 cm^{-1} to ~ 3.5 cm^{-1} at $v = 4$, and suggests that the lower state is perturbed in this region.

3. The resolved fluorescence spectra are partially relaxed rotationally. The rotational structure of the bands in the near infrared is partially resolved and a rotational analysis of the $(v_A,6)$ band at 7470 A, using second differences, shows that B'$\sim$0.29 cm^{-1} and B"$\sim$0.33 cm^{-1}.

Table 7-I: Resolved Fluorescence from the 6032 and 6067 Å Bands

	6032A (J' - 8)		6067A (J' - 13)		(J' - 30)		
v"	ν	ΔG	ν	ΔG	ν	ΔG	ΔG_{ave}
0	17 254.4		16 474.0		16 455.0		
	17 242.5		16 455.0		16 411.8		
		683.0		679.8		679.2	680.7(2.0)
1	16 570.9		15 794.7		15 774.7		
	16 559.9		15 774.7		15 733.8		
				664.9		663.5	664.2(0.7)
2			15 130.1		15 109.6		
			15 109.6		15 072.0		
				651.5		649.7	650.6(0.9)
3	15 256.0		14 478.3		14 461.5		
	15 244.3		14 458.4		14 420.8		
		635.9		632.8		628.5	632.4(3.7)
4	14 619.1		13 846.6		13 832.2		
	14 609.4		13 824.6		13 793.1		
		627.7		627.0		625.8	626.8(1.0)
5	13 991.9		13 219.5		13 206.4		
	13 980.2		13 197.7		13 167.3		
		616.8		620.0		617.6	618.1(1.7)
6	13 375.1		12 598.9		12 587.8		
			12 578.3		12 550.7		
		611.3					611.3
7	12 763.8						
		604.7					604.7
8	12 159.1						

7.3.2 High Resolution

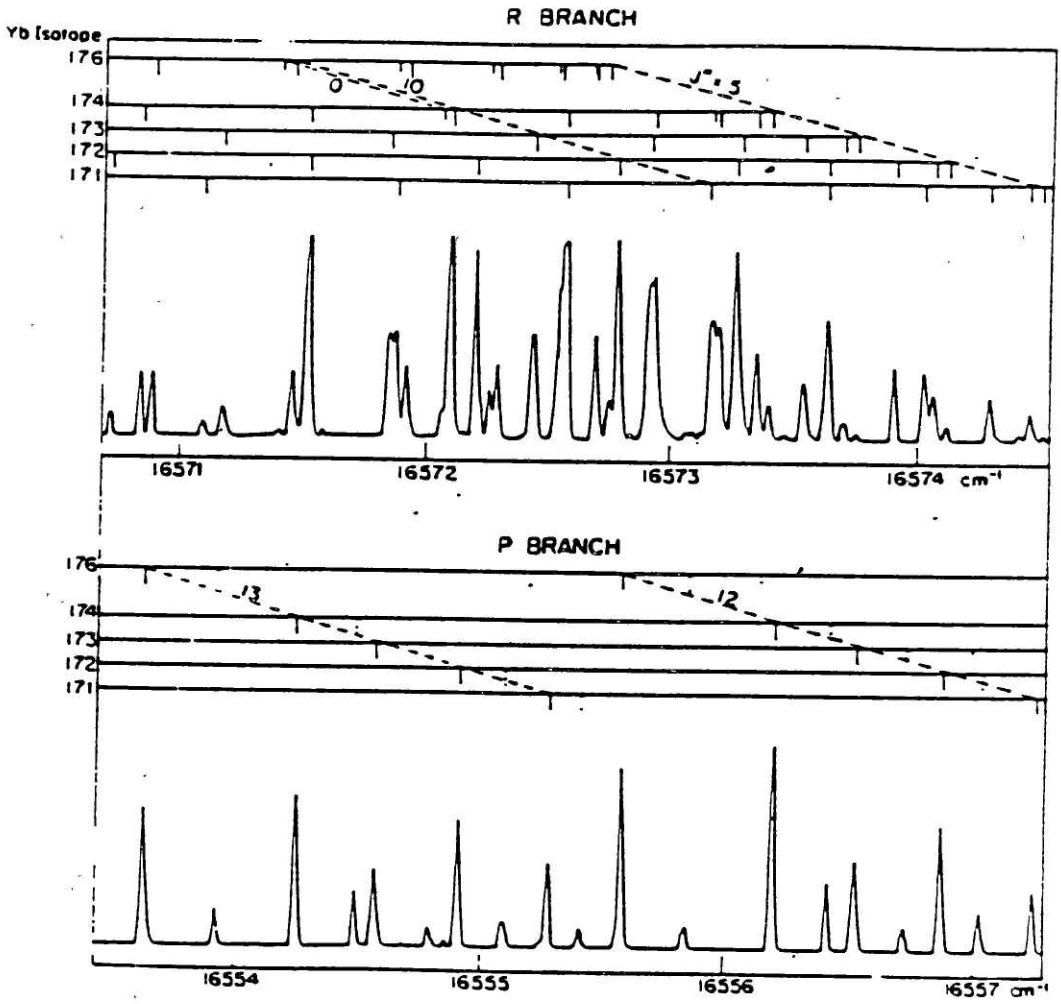
To assign the electronic states, obtain high quality rotational constants and confirm the vibrational assignments, high resolution excitation spectra of the 5793, 6032 and 6067 Å bands were obtained with the dye laser operating single mode. Wavelength selected detection was used in an attempt to reduce spectral congestion. The monochromator was usually set to detect fluorescence in a different band from that excited by the laser. This enabled us to use wide slits, thereby increasing detection band pass and intensity without having to worry about scattered laser light. This also eliminated overlapping bands which did not fluoresce at the monochromator wavelength. The monochromator was usually set on one of the strong fluorescence features in the 7500 Å region. In the excitation regions where the branches overlap, an unsuccessful attempt was made to obtain pure P-branch spectra by setting the monochromator to detect the corresponding R-branch lines. However, because of rotational relaxation to levels that fluoresced at the monochromator wavelength, all the R-branch lines in the laser scan region appeared on the spectra.

In spite of all the steps taken to simplify the spectra, they were still very congested, with no immediately recognizable patterns. Ytterbium has six stable isotopes of significant abundance, with atomic masses of 170, 171, 172, 173, 174 and 176 and abundances of 3.0, 14.3, 21.8, 16.1, 31.8 and 12.7%, respectively. Five of the isotopes, 171-176, were prominent in the spectra, while several weak lines were attributed to the 170 isotope. As a start to the analysis, the laser was set to excite selected P-branch lines and very carefully recorded resolved fluorescence was used to find the frequency of the

corresponding R line. By doing this for four or five P lines, it was usually possible to identify the corresponding R lines in the spectrum and, using this as a starting point, assign the whole spectrum. This technique was used for all three bands and all lines were assigned. A portion of one of the bands is shown in Figure 7-1.The three bands, all of which were similar in structure, contained only R and P branches. The P(1) line was observed in each band, the R(0) line in the 5793 and 6032 Å bands. Identical combination differences showed that the 5793 and 6067 Å bands have the same lower states. All three transitions can thus be assigned, with certainty, as $\Omega' = 0$, $\Omega'' = 0$. A great deal of care had to be taken to detect the low-J R lines as they were overlapped by higher-J lines in the returning limb of the R branch. To detect R(0) and remove the overlapping lines, the monochromator was set to detect the P(2) line using very narrow slits ($\sim$20 μm) so that the higher-J P lines did not fall within the detection band pass. All the observed lines are listed in Table 7-II.It was found that the upper state combination differences, $\Delta_2 F(J)$, of the 5793 and 6032 Å bands were in good agreement, as were those for the lower state of the 5793 and 6067 Å bands, confirming the assignments made from the resolved fluorescence spectra. Combination differences, showing the agreement for one of the isotopes, are shown in Table 7-III.

For each isotopic species, energies and rotational constants were calculated from a simultaneous least squares fit to all three bands. As measurements were not made beyond J = 24, the centrifugal distortion constants, although required to fit the lines adequately, are not well determined, and should be regarded as effective fitting parameters. The constants for all isotopes are listed in Table 7-IV and the ratio of the B values (B_e for the lower state, B_v for the others) for the different isotopes is

Figure 7-1: Excitation Spectrum of the 6032 Å Band



Portions of the excitation spectrum of the 6032-Å band of YbO. The top portion shows the bandhead region of the R branch and includes the first lines for the ^{176}YbO isotope. The diagonal dashed lines connect lines of equal J for the five isotopes. The bottom portion is part of the P branch clearly showing isotopic structure. Lines marked by vertical dashed lines are overlapping high J lines of the R branch.

Table 7-II: Observed Lines in the 5793, 5032 and 6067 Å Bands

YbO 5793 Band

J	171 R(J)	171 P(J)	172 R(J)	172 P(J)	173 R(J)	173 P(J)	174 R(J)	174 P(J)	176 R(J)	176 P(J)
0			17256.212		17255.584		17255.170		17254.191	
1	17257.250	17255.489	17256.685	17254.927	17256.164	17254.398	17255.641		17254.570	
2	17257.601	17254.670	17257.041	17254.110	17256.613	17253.687	17255.997	17253.069	17256.022	17252.094
3	17257.840	17253.742	17257.281	17253.179	17256.767	17252.650	17256.239	17252.137	17256.261	17251.164
4	17257.970	17252.695	17257.409	17252.137		17251.506	17256.376	17251.095	17256.398	17250.122
5	17257.970	17251.532	17257.409	17250.971	17256.889	17250.449		17249.938		17248.965
6	17257.865	17250.243	17257.310	17249.695	17256.785	17249.173	17256.284	17248.664		17247.696
7	17257.643	17248.856	17257.091	17248.302	17256.573	17247.782	17256.064	17247.277	17255.096	17246.313
8	17257.310	17247.346	17256.757	17246.797	17256.239	17246.281	17255.739	17245.777	17254.768	17244.814
9	17256.856	17245.722	17256.306	17245.174	17255.792	17244.660	17255.292	17244.162	17254.327	17243.204
10	17256.284	17243.985	17255.739	17243.440	17255.231	17242.933	17254.733	17242.433	17263.771	17241.480
11	17255.607	17242.131	17255.067	17241.591	17254.569	17241.082	17254.061	17240.590	17263.106	
12	17254.811	17240.166	17254.275	17239.632	17253.770		17253.275		17252.324	
13	17253.900		17253.369		17252.859		17252.376		17251.431	
14	17252.877		17252.352		17251.853		17251.364		17250.425	
15	17251.742		17251.218		17250.722		17250.243		17249.303	
16	17250.492		17249.973		17249.482		17248.998		17248.066	
17	17249.129		17248.613		17248.124		17247.647		17246.719	
18	17247.647		17247.141		17246.656		17246.180		17245.256	
19	17246.062		17245.554		17246.073		17244.600		17243.684	
20	17244.360		17243.867		17243.380		17242.912		17242.000	
21	17242.545		17242.046		17241.562		17241.102		17240.202	
22			17240.122							

YbO 5032 Band

J	171 R(J)	171 P(J)	172 R(J)	172 P(J)	173 R(J)	173 P(J)	174 R(J)	174 P(J)	176 R(J)	176 P(J)
0									16571.399	
1						16571.138		16570.786	16571.879	16570.128
2						16570.340		16569.987	16572.256	16569.327
3	16574.290	16570.183	16573.902	16569.797		16569.426		16569.081	16572.616	16568.420
4	16574.452	16569.176	16574.062	16568.783	16573.700	16568.420	16673.362		16672.693	16667.415
5	16574.502	16568.061	16574.114	16567.685	16573.749	16567.302	16673.404	16568.965	16672.746	16668.337
6	16574.462	16566.834	16574.062	16566.441	16573.700	16566.082	16673.352	16566.738	16672.699	16666.087
7	16574.290	16565.497	16573.902	16565.111	16573.548	16564.766	16673.196	16564.410	16672.546	
8	16574.022	16564.058	16573.640	16563.672	16573.282	16563.319	16572.937	16562.977	16672.287	16662.340
9	16573.640	16562.516	16573.266	16562.132	16572.916	16561.778	16672.578	16561.441	16671.922	16660.900
10	16673.165	16560.862	16572.785	16560.478	16572.442	16560.130	16672.101	16559.794	16671.463	16659.162
11	16572.578	16559.101	16572.204	16558.726	16571.856	16558.380	16571.521	16558.046	16670.884	16657.415
12	16571.887	16557.232	16571.514	16556.864	16571.171	16556.524	16570.837	16556.194	16670.199	16656.568
13	16571.085	16555.265	16570.716	16654.899	16570.377	16564.562	16670.050	16654.237		16663.515
14	16570.183	16553.190	16569.820		16569.484		16569.165		16668.528	
15			16568.812		16568.478		16568.154		16667.536	
16	16668.061		16567.704		16567.373		16567.061			
17	16566.834		16566.488		16566.167		16566.849		16665.239	
18	16565.519		16565.176		16564.852				16663.932	
19	16564.022		16563.763		16563.433		16663.120			
20	16562.565		16562.230		16561.913		16661.602		16561.005	
21	16560.925		16560.594		16560.281		16659.974		16659.384	
22	16659.190		16558.861		16558.555		16658.249		16667.662	
23	16557.344		16557.020		16556.717		16656.417			
24			16555.076		16554.778		16654.480		16663.912	

YbO 6067 Band

J	171 R(J)	171 P(J)	172 R(J)	172 P(J)	173 R(J)	173 P(J)	174 R(J)	174 P(J)	176 R(J)	176 P(J)
0										
1			16476.625				16476.069			16473.786
2	16477.291		16476.984			16472.840			16475.933	16472.975
3	16477.545	16473.416	16477.242	16473.113	16477.116		16476.698	16472.666	16476.191	16472.057
4	16477.685	16472.378	16477.388	16472.080		16470.663	16476.848	16471.538	16476.334	16471.026
5	16477.720	16471.233	16477.425	16470.935	16477.078		16476.879	16470.397	16476.372	16469.885
6	16477.646	16469.975	16477.350	16469.680		16469.409	16476.810	16469.143	16476.299	16468.538
7	16477.456		16477.160	16468.318		16468.043	16476.625	16467.780	16476.117	16467.276
8	16477.160	16467.126		16466.836	16476.696	16466.570	16476.334	16466.309	16475.826	16465.908
9	16476.747		16476.459	16465.248	16476.191	16464.984	16475.933	16464.726	16475.423	16464.228
10	16476.227	16463.834		16463.551	16475.681	16463.290	16475.423	16463.033	16474.918	16462.540
11	16475.598	16462.025	16475.321	16461.745	16475.081	16461.487	16474.801	16461.233	16474.299	16460.743
12	16474.863		16474.585		16474.328		16474.071		16473.576	
13	16474.021		16473.745		16473.490		16473.234		16472.740	
14	16473.068		16472.794		16472.537		16472.287		16471.798	
15	16472.008		16471.738		16471.486		16471.233		16470.745	
16	16470.841		16470.673		16470.323		16470.073		16469.584	
17	16469.568		16469.302		16469.053		16468.803		16468.318	
18			16467.921		16467.673		16467.426		16466.946	
19	16466.703		16466.439				16466.947		16465.463	
20	16465.113		16464.849		16464.604		16464.367		16463.876	
21	16463.421		16463.157		16462.911		16462.666		16462.184	
22	16461.626		16461.362		16461.116		16460.869			

compared to $\rho^2 = (\mu/\mu^i)$, the reduced mass ratio. The agreement is good for

Table 7-III: Comparison of Combination Differences for ^{174}YbO Bands with Levels in Common

	$\Delta_2F'(J)(cm^{-1})$		$\Delta_2F''(J)(cm^{-1})$	
J	5793Å	6032Å	5793Å	6067Å
2	2.928		3.504	3.503
3	4.102		4.902	
4	5.280		6.301	6.301
5		6.449	7.711	7.705
6	7.620	7.614	9.098	9.099
7	8.787	8.785	10.507	10.501
8	9.962	9.960	11.902	11.899
9	11.130	11.137	13.306	13.301
10	12.300	12.307	14.702	14.700
11	13.471	13.475		

the lower states but not as good for the upper states, suggesting that the upper states are weakly perturbed and that the strength of the perturbations varies slightly from isotope to isotope.

It is important to note that YbO belongs to a class of molecules which has an enormous number of electronic states at 2-3 eV and that perturbations will destroy the structural significance of the usual molecular constants. Negative D values, departures from expected isotope ratios for B values, and misleading isotope shifts are all expected to occur. Since most perturbations will be of the heterogeneous ($-BJ\cdot J_a$) type and must vanish at $J = 0$, isotopic information derived from band origins rather than B_v or D_v constants should be more reliable.

Inspection of Table 7-III shows that both upper states have similar rotational constants. Extrapolation of the lower state constants to $v = 6$ gives $B_6 \sim 0.325$ cm^{-1}, consistent with the value of $B_6 = 0.33$ cm^{-1} obtained

Molecular Constants (in cm^{-1}) for the 5793, 6032 and 6067 Å Bands

	Yb Isotope 171	172	173	174	176
Lower State					
B_0	0.35069(5)[b]	0.35047(4)	0.35036(5)	0.35022(4)	0.34987(7)
$10^7 D_0$	5.4(18)	3.1(19)	5.1(22)	7.8(17)	8.2(22)
B_1	0.34636(5)	0.34617(4)	0.34606(5)	0.34594(4)	0.34557(5)
$10^7 D_1$	4.4(18)	2.6(18)	4.9(22)	7.9(17)	7.8(22)
$\Delta G_{1/2}$	683.599(1)	683.432(1)	683.270(1)	683.107(1)	682.792(1)
B_e	0.35286(7)	0.35262(6)	0.35251(7)	0.35236(6)	0.35202(7)
$10^5 \alpha_e$	433(7)	430(6)	430(7)	428(6)	430(7)
ρ^2	1.00148	1.00098	1.00049	1	0.99904
$B_e/B_e(174)$	1.00142(26)	1.00074(24)	1.00043(26)	1	0.99904(26)
ρ	1.00074	1.00049	1.00024	1	0.99952
$\Delta G_{1/2}/\ \Delta G_{1/2}(174)$	1.00072(2)	1.00048(2)	1.00024(2)	1	0.99954(2)
	ω_e = 699 ± 5[a]; $\omega_e x_e$ = 8.0 ± 1.2 (v<4), 3.5 ± 0.7(v>4)				
Upper State A(v = 4)					
T	17256.191(1)	17255.627(1)	17255.098(1)	17254.584(1)	17253.607(2)
B	0.29291(4)	0.29289(4)	0.29295(5)	0.29296(4)	0.29281(5)
$10^7 D$	-1.9(15)	-2.6(16)	0.1(19)	3.4(14)	3.9(19)
B/B(174)	0.99983(26)	0.99976(20)	0.99997(22)	1	0.99949(22)
		ω_e = 547 ± 20			
Upper State B(v=2)					
T	16475.854(1)	16475.551(1)	16475.274(1)	16475.002(1)	16474.488(2)
B	0.29492(4)	0.29489(4)	0.29493(5)	0.29493(4)	0.29475(5)
$10^7 D$	-9.5(15)	-9.5(16)	-6.3(20)	-2.5(15)	-0.7(18)
B/B(174)	0.99997(20)	0.99986(20)	1.00000(22)	1	0.99939(22)
		ω_e = 585 ± 20			

a. Calculated from resolved fluorescence spectra.
b. Numbers in parenthesis are one standard deviation in the last one or two digits.

from the medium resolution analysis of the relaxed fluorescence. The vibrational frequencies, $\Delta G''_{1/2}$, of the lower state were determined for each isotope from the band origins and are listed in Table 7-III. Isotope ratios of vibrational frequencies are in good agreement with theory.

Although the presence of six isotopes greatly added to the complexity of the spectrum and hindered assignment of the lines, it enabled us to determine tentative vibrational numberings and to estimate the vibrational constants of

the upper states without the necessity of observing additional vibrational levels. The 174 isotope was chosen as the standard and the vibrational isotope shifts $\Delta T = T_i - T_{174}$ were calculated for each isotope, using the term energies in Table 7-III. Standard treatment of the isotope effect [4] yields the following expression for the isotope shifts:

$$\frac{\Delta T}{(\rho - 1)} = \omega_e'(v'+1/2) - (1/2)\omega_e''. \qquad (7.7)$$

The second term is required as the term energies are measured relative to $v = 0$ of the lower state. There are also anharmonicity terms but these are small and cannot be determined accurately without good estimates for $\omega_e x_e'$ and $\omega_e x_e''$. For the lower state of each transition, as $\Delta G_{1/2}$ is known accurately and a value of $\omega_e x_e'' \sim 8$ cm^{-1} for the lower levels has been determined from resolved fluorescence, ω_e'' could be calculated ($\omega_e'' = \Delta G_{1/2} + 2\omega_e x_e''$) with an uncertainty of about 5 cm^{-1}.

Application of equation (7.7) then enables us to determine the product, $\omega_e'(v' + 1/2)$. Examination of the various values of ω_e' obtained by trying different values of v' generally shows only one that is physically reasonable. For all three bands, as B" is considerably larger than B', it is almost certain that $\omega_e'' > \omega_e'$. As an example, for state A, we find that if $v' = 3$ $\omega_e' \sim 700$ cm^{-1}, $v' = 4$ $\omega_e' \sim 550$ cm^{-1}, $v' = 5$ $\omega_e' = 450$ cm^{-1}. Of these, 700 cm^{-1} is certainly too large and 450 cm^{-1} probably too small. The combination, $v' = 4$ $\omega_e' \sim 550$ cm^{-1} is the most reasonable of the three, though $v' = 5$ cannot entirely be ruled out. Thus we find that, for the 5793 and 6032 Å bands, $v_A = 4$ and for the 6067 Å band, $v_B = 2$. Extrapolation then enables us to determine the approximate ground vibrational level for each

state. The upper state vibrational constants are included in Table 7-III. The values of ω_e' are not very accurate as there are large uncertainties caused by a combination of uncertainties of 0.002 cm^{-1} in ΔT and 5 cm^{-1} in ω_e'' and by our neglect of terms in both $\omega_e x_e'$ and $\omega_e x_e''$. It must also be stressed that the assignment of v' is based on assigning the first member of the vibrational progressions as $v'' = 0$. This is not necessarily correct but the same vibrational spacing between the first two members of the progression was observed from four different excited vibrational levels, strongly suggesting that the assignment is correct. Any error in assigning v'' would cause an identical error in v', leaving v'-v'' the same, and would require small changes in ω_e'' and ω_e'.

7.4 5929 Å Band

This band was observed when using N_2O as oxidant for the Yb metal. As with all bands save the three already discussed, the fluorescence appeared green by eye.

7.4.1 High Resolution

High resolution excitation spectra were taken with a single mode dye laser while monitoring fluorescence at 4968 Å with the monochromator. The spectrum was red degraded with the Q branch the strongest in intensity, followed by the P branch, and the R the weakest, indicating that the transition is a $\Delta\Omega = -1$. Of the six isotopes, two have nuclear spin, 173 has a spin of 5/2 and 171 has a spin of 1/2. I have picked out what could be the branches due to the two hyperfine components of the 171 isotope, but the

spectrum is far too congested to be sure of the assignment. Consequently I have included the lines in the transition list, but will not list constants derived from a fit. None of the six components of the 173 isotope were observed. Since the hyperfine splitting is greater than the Doppler width, the percent abundance of each component of the 173 isotope is reduced to less than 3%, thus reducing the signal to below the noise. The molecular constants are shown in Table 7-V. The D values are affected by the lack of high-J lines and by a perturbation. They should be considered as fitting parameters more than realistic centrifugal distortion constants. A line list for the transition is given in Table 7-IV. As tabulated, the first lines in each branch, R(3), Q(3) and P(3), have been observed proving, that the transition is $\Omega' = 2 \rightarrow \Omega'' = 3$, which agrees with the $\Delta\Omega = -1$ assignment suggested by the intensity information.

A perturbation was observed to turn on at J$\sim$22. Because the spectrum was so cluttered, I was unable to follow the band into the perturbed region, and therefore could not tell whether the perturbation was in the upper or lower state. A section of the spectrum with assignments is shown in Figure 7-2.

7.4.2 Fluorescence

In the following descriptions of the fluorescence spectra, the wavelengths of the band origins, used as names for the fluorescence bands, are estimated in the following manner. The separation between a fluorescence band and the laser frequency is taken from a low resolution monochromator scan. This value was then added to the band origin of the laser transition. Because the scans are low resolution and uncalibrated, the separations are probably good to

Table 7-IV
YbO 5929 Å Band

171a Isotope

J	R(J)	Q(J)	P(J)
4		16866.203	16863.480
5	16870.163	16866.069	16862.659
6		16865.897	
7			16860.917
8	16871.587	16865.460	16860.002
9		16865.192	16859.057
10	16872.409	16864.906	16858.083
11		16864.578	16857.078
12	16873.109	16864.235	16856.059
13		16863.866	16854.988
14			
15	16873.937	16863.026	16852.798
16		16862.560	16851.652
17	16874.352	16862.068	16850.476
18		16861.548	16849.260
19		16861.003	
20	16874.767	16860.435	
21		16859.823	
22		16859.187	
23	16874.906		
24	16874.906	16857.836	
25		16857.121	
26		16856.369	
27	16874.724		

171b Isotope

J	R(J)	Q(J)	P(J)
2		16866.506	
3			
4			
5	16870.243	16866.157	16862.737
6	16870.738	16865.971	16861.880
7	16871.215	16865.760	
8	16871.660	16865.517	16860.060
9		16865.247	16859.112
10	16872.455	16864.949	16858.126
11		16864.627	16857.121
12	16873.109		
13	16873.381	16863.866	16855.016
14	16873.627	16863.429	16853.895
15			
16			
17			16850.388
18			
19	16874.352		
20	16874.377	16860.155	

172 Isotope

J	R(J)	Q(J)	P(J)
3	16868.366	16865.638	16863.589
4	16868.927	16865.517	16862.793
5	16869.460	16865.371	16861.960
6		16865.192	16861.100
7	16870.441	16864.986	16860.212
8	16870.890	16864.750	16859.294
9	16871.303	16864.485	16858.349
10	16871.695	16864.190	16857.370
11	16872.057	16863.866	16856.369
12	16872.390	16863.519	16855.338
13	16872.698	16863.144	16854.275
14	16872.969	16862.735	16853.180
15	16873.217	16862.303	16852.068
16	16873.441	16861.833	16850.923
17	16873.627	16861.346	16849.746
18	16873.797	16860.830	16848.551
19	16873.937		
20	16874.047	16859.712	
21	16874.136	16859.112	
22	16874.192		

174 Isotope

J	R(J)	Q(J)	P(J)
2	16866.431		16863.026
3	16867.033	16864.304	16862.268
4	16867.598	16864.190	16861.461
5	16868.133	16864.041	16860.633
6	16868.640	16863.866	16859.776
7	16869.112	16863.662	16858.893
8	16869.565	16863.429	16857.976
9	16869.980	16863.166	16857.032
10	16870.373	16862.875	16856.059
11	16870.738	16862.560	16855.061
12	16871.067	16862.212	16854.032
13	16871.382	16861.833	16852.971
14	16871.660	16861.433	16851.884
15	16871.913	16861.003	16850.775
16	16872.140	16860.545	16849.637
17	16872.337	16860.060	16848.471
18	16872.511	16859.549	
19	16872.650	16859.009	
20	16872.760	16858.445	
21	16872.857	16857.836	
22	16872.921	16857.224	
23	16872.969	16856.577	

176 Isotope

J	R(J)	Q(J)	P(J)
3		16863.144	16861.100
4	16866.431	16863.026	16860.298
5	16866.969	16862.875	16859.477
6	16867.478	16862.709	16858.621
7	16867.958	16862.505	16857.735
8	16868.409	16862.268	16856.828
9	16868.831	16862.017	16855.887
10	16869.224	16861.727	16854.917
11	16869.589		16853.921
12	16869.929	16861.071	16852.892
13	16870.243	16860.698	16851.843
14	16870.526	16860.298	
15	16870.782	16859.879	16849.657
16	16871.011	16859.425	16848.524
17	16871.215	16858.948	
18	16871.382	16858.445	
19	16871.542	16857.907	
20	16871.659	16857.350	
21	16871.747	16856.760	
22	16871.825	16856.143	
23			
24		16854.857	

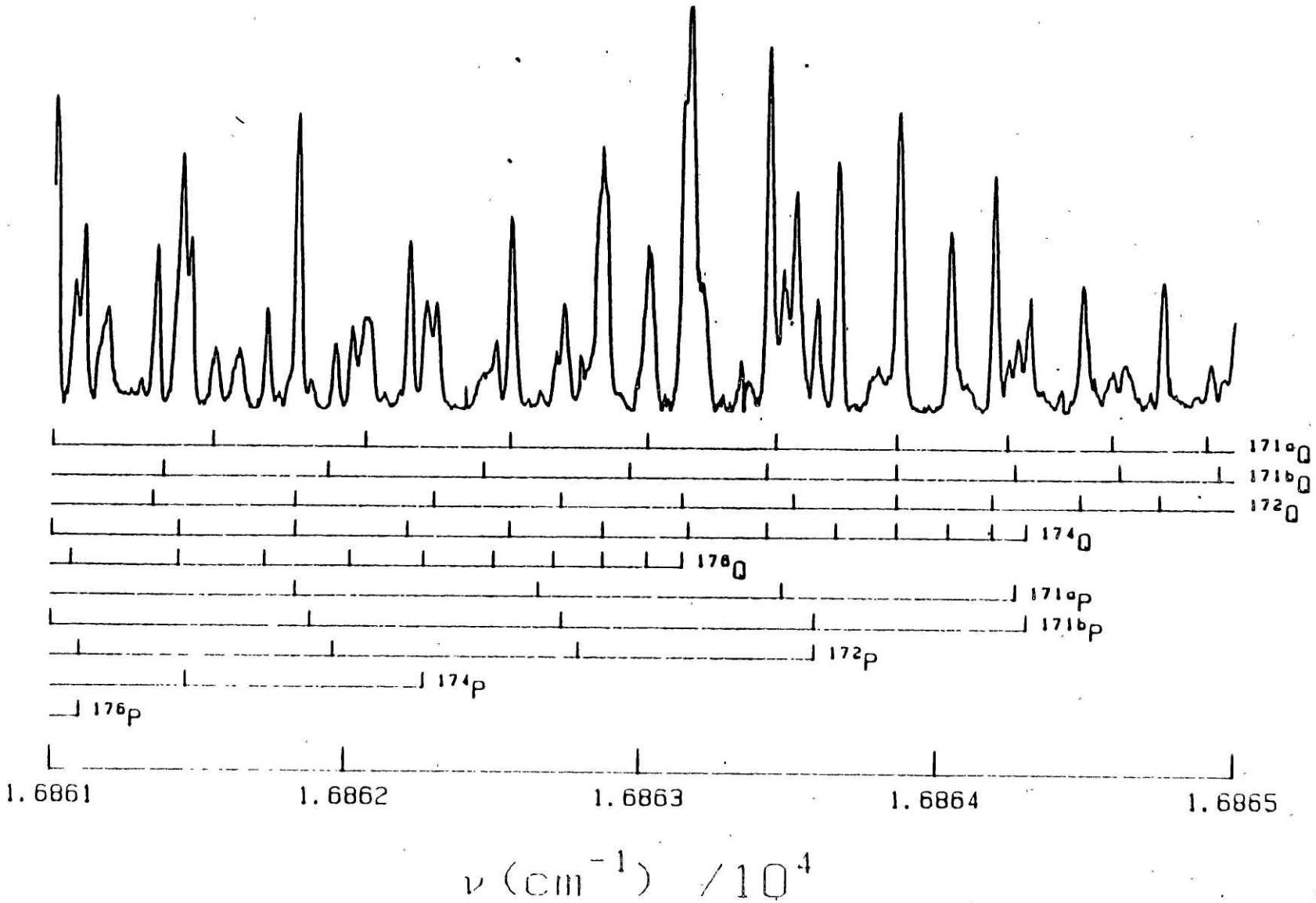
Table 7-V: Constants from the 5929 Å Band

Constant	172 Isotope Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16866 155	0.005
B"	0.35583	0 00002
D"	4.7×10^{-7}	0.3×10^{-7}
B'	0.34102	0.00003
D'	-4.7×10^{-7}	0.3×10^{-7}

Constant	174 Isotope Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16864.820	0.007
B"	0.35529	0.00003
D"	0.2×10^{-7}	$0\ 3 \times 10^{-7}$
B'	0.34066	0.00003
D'	-5.6×10^{-7}	0.3×10^{-7}

Constant	176 Isotope Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16863.654	0.007
B"	0.35513	0.00005
D"	5.5×10^{-7}	0.5×10^{-7}
B'	0.34067	0.00003
D'	-0.4×10^{-7}	$0\ 5 \times 10^{-7}$

Figure 7-2: YbO 5929 Å Band



about 2 Å or 10 cm^{-1}.

I will discuss two different kinds of relative intensities. The most important is the relative intensities of the branches, which helps determine $\Delta\Omega$. The other is an arbitrary number that I have assigned to each fluorescence band which describes the intensity of the band with respect to all others, where an intensity of 10 is arbitrarily assigned to the strongest band.

4940 Å Fluorescence

The intensity varies greatly with J. At low J, J=7, the transition could not be found, while at J=16 its intensity is about 1/6 that of the 4963 Å band. The growth of intensity with J indicates that this band is gaining intensity through a heterogeneous perturbation. The Ω-doubling in the $\Omega''=1$ 4963 Å band suggests that the 4940 Å band is an $\Omega''=0$ band which gains transition strength to the $\Omega'=2$ upper state from the $\Omega''=1$ 4968 Å band. Further support of this comes from the fact that $I_Q > I_R > I_P$, which makes sense if the band borrows its intensity from an $\Omega''=1$ state through an $\Omega''=1 \sim \Omega''=0$ perturbation.

4968 Å Fluorescence

The relative intensity of this band is 10. $I_Q > I_R > I_P$, which suggests that this is a transition into an $\Omega''=1$ state from the $\Omega'=2$ upper state. In support of this assignment, the lower state rotational levels are obviously Ω-doubled. Ω-doubling occurs because a state with single parity for each J, an $\Omega=0$ state, interacts with only one of the parity components of an $\Omega>0$ state, thereby splitting the two otherwise degenerate parity components. All of the doubling must reside in the lower level. Ω-doubling usually occurs in second order for an $\Omega=1$ state and at higher orders of perturbation theory for $\Omega>1$.

Since the Ω-splitting is so large at relatively low J, the level is very probably an Ω=1 perturbed by an Ω = 0. The splittings of the two parity components vs. J are given in Table

Table 7-VI: Splitting of Ω-Doubled Lines in the 4968 Å Band

J	Splitting(cm^{-1})
6	1.3
7	1.6
8	2.3
11	3.9
12	4.6
13	5.5
17	8.7
18	9.7
19	10.5

5178 A Fluorescence

The relative intensity of this band is 0.2. The band is Ω-doubled, thus labeling it as an Ω"=1 band. It is 830 cm^{-1} from the 4968 Å band. 830 cm^{-1} is the standard vibrational frequency for an $f^N s$ configuration, so the 5178 Å band is probably a transition into the next higher ground state vibrational level than the 4968 Å band.

5424 Å Fluorescence

This is an Ω"=1 ($I_Q > I_R > I_P$), non-Ω-doubled band with relative intensity of 4. Because the separation from the 5178 Å band is 862 cm^{-1} and because of the lack of Ω-doubling, this is not an additional vibrational level of the 4968, 5178 Å progression but a different Ω"=1 level.

7.4.3 Summary of 5929 Å Band

Table 7-VIIsummarizes the information garnered from the 5929 Å band.

Table 7-VII: Bands Observed from the 5929 Å Laser Excitation

Fluorescence Wavelength(A)	Name (Ω, v)	Separation from Laser(cm^{-1})
4940	0. v_a	3377
4968	1. v_b	3263
5178	1. v_b+1	2447
5424	1'. v_c	1572
5929	3. v_d	0

Because the band shows the signs of a strong perturbation, extra lines and the disruption of the smooth pattern based on an energy level pattern of BJ(J+1), I do not trust isotopic information. Therefore, I did not use the isotopic information to estimate ω_e' as in the previous bands.

7.5 5925 Å Band

7.5.1 High Resolution

Dye laser excitation scans were taken with the monochromator detecting the fluorescence at 5648 Å. The spectrum was red degraded with $I_Q > I_R >> I_P$. The first lines were R(3) and Q(4), identifying the transition as $\Omega'=4 - \Omega''=3$. The P branch was so weak that only a few lines were found. The extreme weakness of the P branch can be explained by a perturbation. Given an unperturbed state Ψ_0 and a perturbing state Ψ_1 first order perturbation theory gives the wavefunction as [1]

$$\Psi = \Psi_0 + \Psi_1[<1|H_{pert}|0>/(E_0-E_p)]$$

where $<1|H_{pert}|0>$ is the matrix element of the perturbation between the two states. The important point is that a perturbed state can be approximated as a linear combination of two wavefunctions, assuming only one perturbing state. Next the intensities of the various branches must be examined.

Hougen [5] explains how to calculate rotational line intensities. The matrix element of interest is

$$<J_a'\Omega';J'M'\Omega'|\mu_p|J_a\Omega;JM\Omega>, \tag{7.8}$$

where μ_p is the electric dipole moment operator quantized in the laboratory frame. The intensity of the transition will be proportional to the square of the matrix element. Using the direction cosine matrix elements, the lab fixed components of μ can be written in terms of the molecule fixed components,

$$\mu_p = \Sigma_q \alpha_{pq} \mu_{q'}$$

where μ_q are the elements of μ in molecule fixed coordinates and α_{pq} are the direction cosines for the transformation between space and molecule fixed coordinates. Putting this expression into equation (7.8) gives

$$<J_a'\Omega';J'M'\Omega'|\Sigma_q \alpha_{pq}\mu_q|J_a\Omega;JM\Omega>$$

$$= \Sigma_q <J_a'\Omega'|\mu_q|J_a\Omega> \quad <J'M'\Omega'|\alpha_{pq}|JM\Omega>.$$

The two matrix elements can be separated because the μ operates on the electrons while the direction cosines operate on the Euler angles of the molecule. The matrix elements involving μ_q are treated as parameters which will be constant for a given transition. Hougen divides the matrix elements of

the direction cosines into the product of three factors. The important factor for the intensity perturbations of the bands analyzed is summarized in Table 7-VIII.

Table 7-VIII: Direction Cosine Matrix Element for Relative Intensity

R Branch (J'=J+1)

$\Delta\Omega$	Factor
0	$2[(J+\Omega+1)(J-\Omega+1)]^{1/2}$
±1	$\mp[(J\pm\Omega+1)(J\pm\Omega+2)]^{1/2}$

Q Branch (J'=J)

$\Delta\Omega$	Factor
0	2Ω
±1	$[(J\mp\Omega)(J\pm\Omega+1)]^{1/2}$

P Branch (J'=J-1)

$\Delta\Omega$	Factor
0	$2[(J+\Omega)(J-\Omega)]^{1/2}$
±1	$\pm[(J\mp\Omega)(J\mp\Omega-1)]^{1/2}$

Suppose the following three conditions are satisfied:

1. The upper state is perturbed such that the wavefunction is given by $\Psi = \Psi_0 - k\Psi_1$ where k is a positive constant.
2. Both Ψ_0 and Ψ_1 have transition intensity to the lower state Ψ_L.
3. The Ω value of Ψ_0 is greater by one than the lower state Ψ_L while Ω of Ψ_1 equals that of the lower state.

The intensity of a transition excited with z polarized light will be proportional to the square of $<\Psi_0+k\Psi_1|\mu_Z|\Psi_L>$. Thus the R branch intensity will be proportional to

$$[[(J+\Omega+1)(J+\Omega+2)]^{1/2} + 2k[(J+\Omega+1)(J-\Omega+1)]^{1/2}]^2$$

while the intensity of the P branch will be proportional to

$$[[(J-\Omega)(J-\Omega-1)]^{1/2} - 2k[(J+\Omega)(J-\Omega)]^{1/2}]^2.$$

As k becomes larger the intensity of the P branch will decrease while the intensity of the R branch will increase, exactly the effect which has been observed.

A typical piece of the spectrum is shown in Figure 7-3 which demonstrates the clumping effect arising from the small isotopic splitting. A strong perturbation makes it impossible to track the band above J~21. Only one of the two hyperfine components of the rotational lines for the 171 isotope can be seen because the other hyperfine component apparently lies under the 172 lines. Futhermore, no 173 lines were observed, so the hyperfine splitting must be significant in this band.

A list of constants derived from a fit is given in Table 7-IX. All the observed lines are listed in Table 7-X.

The question naturally arises as to whether the $\Omega''=3$ state in this band is the same as the lower state of the 5929 Å transition. Table 7-XI compares the two sets of combination differences for the lower states of the 174 isotopes. The similarity of the combination differences indicates that the lower states in the 5925 and 5929 Å bands are the same state.

7.5.2 Fluorescence

5648 Å Fluorescence

This band lies 827 cm^{-1} to the blue of the laser excited transition. The intensity pattern is $I_Q > I_R > I_P$ thus labeling the lower state as $\Omega''=3$ since $\Omega=4$ from the laser transition. This, along with the characteristic

Figure 7-3: YbO 5925 Å Band

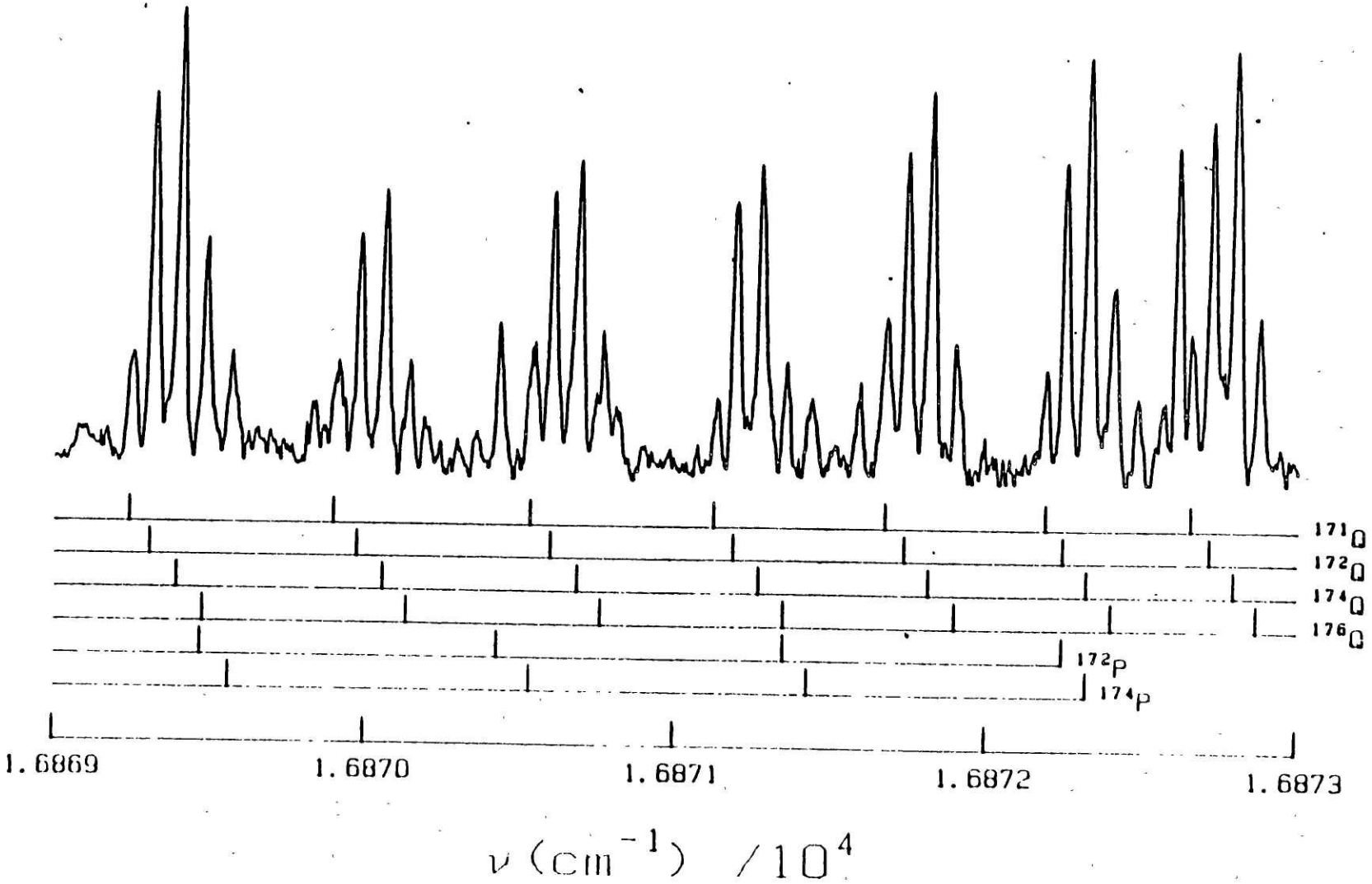


Table 7-IX: Constants from the 5925 Å Band

171 Isotope		
Constant	Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16875.182	0.002
B"	0.3574	0 0001
D"	4.4×10^{-6}	0.2×10^{-6}
B'	0.3386	0.0001
D'	$6\ 7 \times 10^{-6}$	0.1×10^{-6}

172 Isotope		
Constant	Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16875.314	0.002
B"	0.3544	0.0001
D"	-8.3×10^{-6}	0.1×10^{-6}
B'	0.3347	0.0001
D'	-8.5×10^{-6}	0.1×10^{-6}

174 Isotope		
Constant	Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16875.372	0.001
B"	0.35549	0.00005
D"	-1.1×10^{-6}	0.1×10^{-6}
B'	0 33599	0.00005
D'	-0.92×10^{-6}	0.1×10^{-6}

176 Isotope		
Constant	Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16875.437	0.001
B"	0.35595	0.00008
D"	2.2×10^{-6}	0.1×10^{-6}
B'	0.33657	0.00008
D'	2.6×10^{-6}	0.1×10^{-6}

Table 7-X: 5925 Å Line List

171 Isotope

J	R(J)	Q(J)	P(J)
1		16875.154	
2			
3			
4			
5		16874.625	
6	16879.122		
7	16879.526		16869.394
8	16879.892	16873.824	
9		16873.474	
10	16880.489	16873.088	
11	16880.734	16872.654	16865.235
12	16880.930	16872.185	
13		16871.673	
14		16871.124	
15		16870.533	
16		16869.899	

172 Isotope

J	R(J)	Q(J)	P(J)
3	16877.756		
4	16878.280	16874.914	16872.245
5		16874.713	
6	16879.186		
7	16879.591	16874.205	
8	16879.952	16873.892	
9		16873.539	
10	16880.551	16873.148	
11	16880.800	16872.717	16865.311
12		16872.245	
13		16871.736	
14		16871.187	
15		16870.600	
16		16869.975	
17		16869.309	
18		16868.602	
19		16867.857	
20		16867.067	

174 Isotope

J	R(J)	Q(J)	P(J)
1/2 STATES, F,E, R,P			
1	16876.678		
2			
3	16877.828		
4	16878.345	16874.979	
5	16878.819	16874.789	16871.428
6	16879.260	16874.553	
7	16879.658	16874.280	
8	16880.019	16873.966	
9	16880.341	16873.614	16867.561
10	16880.620	16873.222	16866.504
11	16880.864	16872.794	16865.400
12	16881.065	16872.322	16864.254
13	16881.237	16871.815	16863.075
14	16881.369	16871.269	
15	16881.446	16870.685	
16	16881.490	16870.056	
17		16869.394	
18		16868.689	
19		16867.945	
20	16881.305	16867.156	

176 Isotope

J	R(J)	Q(J)	P(J)
2			16873.966
3	16877.900	16875.208	
4		16875.049	
5	16878.885	16874.860	
6	16879.329	16874.625	
7	16879.724		
8	16880.095	16874.044	16868.661
9		16873.691	
10	16880.689	16873.299	
11	16880.930	16872.869	
12		16872.403	16864.342
13		16871.899	
14	16881.421	16871.350	
15	16881.490	16870.761	
16		16870.134	
17		16869.479	
18		16868.775	
19		16868.029	
20		16867.244	
21		16866.408	

Table 1-XI: Comparison of Combination Differences for Lower States of the 5929 and 5925 Å Bands

	5929 Å Band			5925 Å Band		
J	R(J-1) - P(J+1)	R(J) - Q(J+1)	Q(J) - P(J+1)	R(J-1) - P(J+1)	R(J) - Q(J+1)	Q(J) - P(J+1)
2		2.127				
3	6.400	2.843	2.843		2.849	
4	7.822	3.557	3.557		3.556	3.551
5	.9.240	4.267	4.265		4.266	
6	10.664	4.978	4.973		4.980	
7	12.080	5.683	5.686		5.692	
8	13.506	6.399	6.397	13.515	6.405	6.405
9	14.919	7.105	7.107	14.941	7.119	7.110
10	16.341	7.813	7.814	16.366	7.826	7.822
11	17.767	8.528	8.528	17.789	8.542	8.540
12	19.183	9.234	9.241		9.250	9.247
13	20.607	9.949	9.949		9.968	
14	22.023	10.657	10.658		10.684	
15	23.442	11.368	11.366		11.390	
16		12.080	12.074		12.096	
17		12.788				
18		13.502				
19		14.205				
20		14.924				
21		15.633				
22		16.344				

vibrational separation, identify this $\Omega''=3$ state as the next lower vibrational level of the same lower electronic state that is responsible for the 5925 Å band.

7.5.3 Summary of 5925 Å Band

Table 7-XIIsummarizes the 5925 Å band.

Table 7-XII: Bands Observed from the 5925 Å Laser Excitation

Fluorescence Wavelength(A)	Name (Ω, v)	Separation from Laser(cm^{-1})
5648	$3, v$	827
5925	$3, v+1$	0

This band is not very interesting since the fluorescence spectrum contains transitions to only one state and therefore provides no new information for the energy level diagram.

7.6 6021 Å Band

7.6.1 High Resolution

An excitation spectrum was obtained with a single mode dye laser while monitoring the fluorescence at 4829 Å. This red-degraded band has P(3), Q(3) and R(3) as first lines along with an intensity pattern of $I_Q > I_P > I_R$. This establishes that the transition is $\Omega'=2$ - $\Omega''=3$. A strong perturbation prevents analysis above J=17. Lines assignable to the 173 isotope are missing, presumably the hyperfine splitting has reduced their intensity to below the noise, and the 171 isotope lines are split into two components. As before I will give the lines of the 171 isotope for informational purposes, but I do not

feel confident enough in the lines to do a fit. Table 7-XIIIlists the constants derived from a fit to the data and Table 7-XIV lists all assigned lines. As for the previous excitation analyses, the D values are not physically reasonable.

Table 7-XIII: Molecular Constants from the 6021 Å Band

	172 Isotope	
Constant	Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16607.908	0 001
B"	0.35396	0 00007
D"	-0.1×10^{-6}	$0\ 1 \times 10^{-6}$
B'	0.33103	0.00006
D'	2.8×10^{-6}	0.1×10^{-6}

	174 Isotope	
Constant	Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16607.287	0.0007
B"	0.35351	0.00004
D"	-6.2×10^{-7}	0.8×10^{-7}
B'	0.33072	0.00004
D'	45.0×10^{-7}	0.7×10^{-7}

	176 Isotope	
Constant	Value(cm^{-1})	Uncertainty(cm^{-1})
T_v	16606.706	0.001
B"	0.3542	0.0001
D"	3.1×10^{-6}	0.3×10^{-6}
B'	0.3306	0.0001
D'	5.0×10^{-6}	0.2×10^{-6}

7.6.2 Fluorescence

4803 Å Fluorescence

The relative intensity of this band is 2. The intensities of the branches provide no information on $\Delta\Omega$. The band gains significant intensity with increasing J.

4829 Å Fluorescence

Table 7-XIV
YbO 6021 Å Band

171a Isotope

J	R(J)	Q(J)	P(J)
1	16609.443	0.000	0.000
2	0.000	16608.029	0.000
3	16610.537	16607.890	16605.909
4	16611.025	16607.720	16605.068
5	16611.451	16607.499	16604.186
6	0.000	16607.220	16603.257
7	16612.196	16606.910	16602.274
8	16612.488	16606.542	0.000
9	16612.745	0.000	0.000
10	0.000	16605.707	0.000
11	16613.104	0.000	0.000
12	16613.213	0.000	16596.761
13	16613.279	16604.090	16595.541

171b Isotope

J	R(J)	Q(J)	P(J)
2	16610.166	0.000	16606.832
3	0.000	16608.029	16606.030
4	0.000	16607.825	0.000
5	16611.562	16607.583	0.000
6	0.000	16607.303	0.000
7	0.000	0.000	16602.326
8	16612.528	0.000	0.000
9	0.000	0.000	0.000
10	16612.923	0.000	16599.045
11	16613.063	16605.137	16597.864
12	0.000	16604.568	16596.645
13	0.000	16603.940	0.000
14	16613.145	16603.295	0.000

172 Isotope

J	R(J)	Q(J)	P(J)
2	16609.761	0.000	0.000
3	0.000	16607.630	16605.645
4	16610.760	16607.446	16604.798
5	16611.195	16607.220	16603.906
6	16611.562	16606.941	16602.968
7	16611.901	16606.624	16601.985
8	16612.196	16606.243	16600.951
9	16612.431	16605.821	16599.873
10	16612.616	16605.350	16598.742
11	0.000	16604.826	16597.563
12	16612.845	16604.251	16596.330

13	0.000	0.000	0.000
14	16612.845	16602.968	0.000
15	0.000	0.000	0.000
16	16612.650	0.000	0.000

174 Isotope

J	R(J)	Q(J)	P(J)
2	0.000	16607.158	16605.821
3	0.000	16607.018	16605.033
4	0.000	16606.832	16604.186
5	0.000	16606.606	16603.295
6	16610.940	16606.321	16602.354
7	16611.277	16605.994	16601.372
8	16611.562	16605.616	16600.333
9	16611.794	0.000	16599.253
10	16611.969	16604.715	16598.119
11	16612.094	16604.186	16596.935
12	16612.170	16603.608	16595.699
13	0.000	16602.968	16594.416
14	16612.130	16602.274	16593.066
15	16612.031	16601.519	0.000
16	16611.861	16600.713	0.000
17	0.000	16599.839	0.000
18	16611.341	16598.892	0.000

176 Isotope

J	R(J)	Q(J)	P(J)
1	0.000	16606.668	16605.994
2	0.000	0.000	0.000
3	16609.061	16606.415	16604.433
4	0.000	16606.243	0.000
5	16609.965	16605.994	16602.694
6	16610.333	16605.707	16601.754
7	16610.666	16605.382	16600.764
8	16610.940	16605.003	16599.729
9	16611.159	16604.568	16598.643
10	16611.341	16604.090	16597.507

This Ω-doubled band has a relative intensity pattern of $I_Q > I_R > I_P$. It is the strongest of the fluorescence bands with a relative intensity of 10. There can be no question that this is an Ω"=1 band. The Ω-doubling vs. J is listed in Table 7-XV.

Table 7-XV: Splitting of Ω-Doubled 4829 Å Band

J	Splitting(cm^{-1})
5	0.9
6	1.4
7	1.6
8	1.8
9	2.6
10	3.4
11	3.7

5027 Å Fluorescence

This band has a relative fluorescence intensity of 5. Ω-doubling along with an intensity pattern of $I_Q > I_R > I_P$ label the state as Ω"=1. Ω-doubling vs. J is listed in Table 7-XVI.

Table 7-XVI: Splitting of Ω-Doubled 5027 Å Band

J	Splitting(cm^{-1})
6	1.5
7	1.5
10	3.9
11	4.0
12	4.4
13	5.1
14	6.0
15	6.6

5244 Å Fluorescence

The relative intensity of the band is 1, $I_Q > I_R > I_P$, and the band is Ω-doubled, all the classic signs of an Ω"=1 state. Table 7-XVII lists the

Table 7-XVII: Splitting of Ω-Doubled 5244 Å Band

J	Splitting(cm^{-1})
10	3.4
11	3.6
12	4.0

Ω-doubling.

5254 Å Fluorescence

The relative intensity of this band is 2. $I_Q > I_R > I_P$, but there is no Ω-doubling, so this is a different Ω"=1 state from the previous one.

5733 Å Fluorescence

The relative intensity of this band was 0.2, with $I_Q > I_P > I_R$, suggesting that the lower state is an Ω"=3. The intensity pattern, along with a separation of 837 cm^{-1} from the laser transitions, labels this as the next lower vibrational level of the laser transition.

6339 Å Fluorescence

This weak band, relative intensity of 0.8, has $I_Q > I_P > I_R$, and is located 832 cm^{-1} away from the laser excited transition. It is an Ω"=3 state with vibrational quantum one higher than the Ω"=3 level in the laser excited transition.

7.6.3 Summary of 6021 Å Band

Table 7-XVIIIsummarizes the 6021 Å band.

Note that if the 4803 Å Ω"=0 state is assumed to be the same as the 4940 Å Ω"=0 state coming from the 5929 Å laser excitation, a perfect match is acheived between the partial energy level diagrams. The agreement of the

Table 7-XVIII: Bands Observed from the 6021 Å Laser Excitation

Fluorescence Wavelength(A)	Name (Ω, v)	Separation from Laser(cm^{-1})
4803	0, v_a	4214
4829	1, v_b	4106
5027	1, v_b+1	3285
5244	1, v_b+2	2462
5254	1', v_c	2426
5733	3, v_d	837
6021	3, v_d+1	0
6339	3, v_d+2	-832

Ω" values for the same levels observed from two different upper states leaves little doubt that the assignments are correct.

7.7 4518 Å Band

The molecule was excited with a single mode dye laser using Stilbene 420 dye. In spite of a large number of high resolution scans, I was unable to rotationally analyze the transition. However the intensity information from the fluorescence data was sufficient to identify most of the observed lower states.

7.7.1 Fluorescence

4518, 4663 and 4811 Å Band Fluorescence

The 4518 Å band is the laser excitation band. The other two bands, with separations of 690 and 660 cm^{-1}, identify the lower level in the laser excited transition as the $v" = 0$ Σ^+ state which is the lower state in the 5793, 6032 and 6067 Å bands while the other two fluorescence bands involve transitions into $v" = 1$ and 2. When a line in the P or R branch is excited, the fluorescence back to the laser excited band has two branches, while a Q

branch excitation produces only Q branch fluorescence. This identifies the upper state as $\Omega'=1$. The relative intensities the 4518, 4663 and 4811 Å bands were 5, 0.6 and 0.06.

4712 Å Fluorescence

This band was the strongest in fluorescence with a relative intensity of 10. When a Q branch was excited with the laser, P and R branches were observed in the 4712 Å fluorescence while a P or R branch excitation produces only Q branch fluorescence. The laser transition has already been identified as $\Omega'=1$ - $\Omega''=0^+$. Therefore, all of the levels of the ground state have e-parity. P or R branch excitations go to the e-parity component of the $\Omega'=1$ state which then fluoresces by way of a Q branch to an f-parity level of the lower state in the 4712 Å band. Q branch excitation populates the f-parity component of the $\Omega' = 1$ state which fluoresces in P and R branches to the f-parity component of the lower state of the 4712 Å band. Since fluorescence is seen only to f-parity states, the lower state of the 4712 Å fluorescence band must be an $\Omega''=0^-$.

4734 and 4927 Å Fluorescence

These two bands have relative intensities of 5 and 0.06. The 830 cm^{-1} separation labels the two as consecutive vibrational levels of states from the $f^{13}s$ configuration. No matter which branch was excited, a P and R branch was always observed in the fluorescence of the two bands. Only an $\Omega'' = 1$ lower state would give this behavior, since for $\Delta\Omega = 0$ the Q branch can be very weak. The relative intensities of the P and R branches are greatly affected by a perturbation. For some excitations, the intensity of the P branch is about 8 times greater than that of the R branch, while in others the

intensities are approximately equal. As I have already discussed, interference effects arising from perturbations by states differing in Ω can cause the relative intensities of the P, Q and R branches to change. A single $\Omega' = 2$ or $\Omega' = 0$ state perturbing the upper $\Omega' = 1$ state could cause the P branch to be significantly larger than the R branch, but the Q branch intensity would be increased by an amount equivalent to the increase of the P branch. Since the Q branch has no observed intensity, one perturbing state is not sufficient to explain the anomaly. However, the intensity observations can be explained if there are two states perturbing the upper $\Omega' = 1$ state, an $\Omega = 0$ and an $\Omega = 2$ state. If the perturbation and transition matrix elements have the same sign, the $\Omega = 0$ level must be at higher energy and the $\Omega = 2$ at lower energy. In this case the wavefunction of the upper state will be given by

$$\Psi_{upper} = \Psi_1 - a\Psi_0 + b\Psi_2$$

where Ψ_1 is the wavefunction for $\Omega = 1$ and a and b are positive numbers. From Table 7-VII the intensity of the R branch will be proportional to the square of

$$2[J(J+2)]^{1/2} - a[J(J+1)]^{1/2} - b[(J+2)(J+3)]^{1/2}$$

while the intensity of the P branch will be proportional to

$$2[(J+1)(J-1)]^{1/2} + a[(J-1)(J-2)]^{1/2} + b[J(J+1)]^{1/2}.$$

Finally, the Q branch intensity will be proportional to the square of

$$2 + a[(J-1)(J+2)]^{1/2} - b[J(J+1)]^{1/2}. \quad (7.9)$$

If a $\sim$ b, then the Q intensity will be very small.

There is one problem with this explanation. The intensity anomaly was observed with P,R branch excitation and with Q branch excitation. Thus the intensity perturbation seems to be independent of the parity of the upper state. Since an $\Omega = 0$ state consists exclusively of either e or f-parity levels, it can perturb only one parity of the $\Omega' = 1$ state. Two $\Omega = 0$ states of opposite parity are required to explain the parity independence. The $\Omega = 0^-$ state 100 cm^{-1} below the lower state $\Omega'' = 1$ state could be the additional $\Omega = 0$ state. Because this perturbing $\Omega = 0$ state interacts with the lower state and because the energy of the $\Omega = 0$ state lies below the $\Omega'' = 1$ state, the contribution to the intensity of the transition due to perturbation will have exactly the same sign as the perturbation by the $\Omega = 0$ in the upper state. Thus the intensity anomaly of the 4734 and 4927 Å bands could be due to an upper state perturbation from an $\Omega = 0^+$ at higher energy and an $\Omega = 2$ at lower energy than the $\Omega' = 1$ upper state.

5070, 5293 and 5531 Å Fluorescence

These bands have relative intensities of 1, 0.1 and 0.06. The separations of 821 and 818 cm^{-1} suggest that the lower states belong to the $f^{13}s$ configuration. P, Q and R branches are always observed in fluorescence, although the relative intensities of the branches are J dependent. Because the intensities are not consistent with the Honl-London factors for an unperturbed band, the $\Delta\Omega$ of the transition cannot be assigned from branch intensities. However, the lower state cannot be an $\Omega'' = 0$, since a transition into a 0 state would not have all three branches. Furthermore, Q branch is always the

most intense of the three, which tends to rule out an $\Omega'' = 1$. An $\Omega'' = 2$ lower state is the most likely possibility.

5143 Å Fluorescence

The relative intensity of this band is 0.3. It was too weak to obtain good resolved fluorescence spectra, but judging from the difficulty of using relative branch intensities in the other bands, monochromator scans would be of little use. The important point is that there is definitely a band here.

5715, 5994, 6299 and 6631 Å Fluorescence

These bands have relative intensities of 1, 0.1, 0.1 and 0.06. When a P or R branch is excited, two branches are observed in fluorescence, whereas a Q branch excitation produces a single line in fluorescence, labeling the lower level as an $\Omega'' = 0^+$ state.

7.7.2 Summary of 4518 Å Band

The 100 cm^{-1} separation of an $\Omega'' = 0$ from an $\Omega'' = 1$ state identifies these two states as the 0 and 1 states observed in both the 6021 and 5929 Å bands. Furthermore the 1780 cm^{-1} interval between the 0^- state and the unknown state at 5143 Å matches the splitting of 1788 cm^{-1} between the states labeled $\Omega'' = 0$ and 1' in the 5929 and 6021 Å bands perfectly, thus labeling the lower state of the 5143 Å transition. Therefore, with this band I now have the linkage between the f^{14} and $f^{13}s$ configurations. Table 7-XIX summarizes the 4518 Å band.

Table 7-XIX: Bands Observed from the 4518 Å Laser Excitation

Fluorescence Wavelength(A)	Name (Ω, v)	Separation from Laser(cm^{-1})
4518	$0^+, v_a=0$	0
4663	$0^+, v_a=1$	-690
4811	$0^+, v_a=2$	-1350
4712	$0^-, v_b=0$	-910
4734	$1, v_c=0$	-1010
4927	$1, v_c=1$	-1840
5070	$?, v_d$	-2410
5293	$?, v_d+1$	-3240
5531	$?, v_d+2$	-4054
5143	$1', v_e$	-2690
5715	$0^+, v_f$	-4637
5994	$0^+, v_f+1$	-5450
6299	$0^+, v_f+2$	-6258
6631	$0^+, v_f+3$	-7052

7.8 5955 Å Band

Very little work was done on this band, mainly because it was obvious that a rotational analysis would be difficult. However a long range monochromator scan yielded a beautiful vibrational progression of the f^{14} 0^+ state. There is also a shorter progression into the $\Omega = 1$ state which starts 1009 cm^{-1} above the 0^+ state. The fluorescence bands are listed in Table 7-XX.

7.9 5952 Å Band

7.9.1 High Resolution

It was not possible to rotationally analyze this transition. All the

Table 7-XX: Bands Observed from the 5955 Å Laser Excitation

Fluorescence Wavelength(A)	Name (Ω, v)	Separation from Laser(cm^{-1})
4066	$0^+, v_a=0$	7813
4302	$0^+, v_a=2$	6461
4425	$0^+, v_a=3$	5818
4552	$0^+, v_a=4$	5185
4686	$0^+, v_a=5$	4559
4825	$0^+, v_a=6$	3944
4974	$0^+, v_a=7$	3324
5124	$0^+, v_a=8$	2735
5274	$0^+, v_a=9$	2180
5449	$0^+, v_a=10$	1571
4240	$1, v_c=0$	6804
4394	$1, v_c=1$	5977
4740	$1, v_c=3$	4316

information comes from fluorescence scans.

7.9.2 Fluorescence

5352 Å Fluorescence

The relative intensity of this band is 1. Only P and R branches are observed, which indicates either that the transition involves parity selection or that $\Delta\Omega = 0$. A single parity would be excited only if one of the levels is an $\Omega'' = 0$ or if the lower state is Ω-doubled. Since P, Q and R branches are observed in the laser transition, this rules out an $\Omega'' = 0$ state. There is no evidence of Ω-doubling in the laser band fluorescence. Of course, the Ω-doubling could be large enough that the laser would excite only one parity but small enough that Ω-doubling could not be detected in a monochromator scan. However, the Ω-doubling would have to be far smaller than has been observed up to now in YbO. Therefore this transition is probably $\Delta\Omega = 0$.

5439 Å Fluorescence

The relative intensity of this band is 10. The intensities of the P, Q and R branches are clearly $I_Q > I_R > I_P$, establishing this transition as $\Delta\Omega = +1$.

5952 Å Fluorescence

This is the fluorescence of the laser excited excited transition. For Q line excitation at low J, $I_P > I_R$ so the transition is probably $\Delta\Omega = -1$.

7.9.3 Summary of 5952 Å Band

The absolute Ω values of the observed states can be determined if the lower state of the 5352 Å band is aligned with the lower state of the 5070 Å band. This locates the lower state in the 5439 Å band at the same energy as the lower state of both the 5143 Å transition excited through the 4518 Å band and the 5254 Å fluorescence band excited by the 6021 Å band. Thus the lower state of the 5952 Å laser excited transition is the same $\Omega'' = 3$ state that appeared in the 5143 Å fluorescence excited in the 6021 Å band. The omega values are summarized in Table 7-XXI.

Table 7-XXI: Bands Observed from the 5952 Å Laser Excitation

Fluorescence Wavelength(A)	Name (Ω, v)	Separation from Laser(cm^{-1})
5352	2. v_a=0	1881
5439	1'. v_b=0	1583
5952	3. v_c=0	0

7.10 Discussion

I have been noting the linkages with the description of the bands so that by now a lower state energy level diagram is fairly easy to construct. With the Ω'' values from the spectra and a qualitative knowledge of the expected ligand field level pattern, I can assign J_a values. The lowest state is an $\Omega'' = 0^+$ state which belongs to the f^{14} configuration, as indicated by its 683 cm^{-1} vibrational frequency. The 0^- state, which lies 910 cm^{-1} higher in energy, must be the $J_a = 4$, $\Omega = 0^-$ state (denoted as $(4,0^-)$), since this is the lowest lying 0^- state expected. The operator which causes Ω-doubling is $BJ \cdot J_a$, which has off-diagonal matrix elements in Ω but not in J_a. Therefore the $\Omega'' = 1$ state about 100 cm^{-1} higher in energy than the 0^- state and perturbed by the 0^- state must be the (4,1) state. The other $\Omega'' = 1$ state, called 1' in the description of the individual bands, must be (3,1). The $\Omega = 2$ state below (3,1) must be the (4,2) level since, according to the ligand field prediction, the (3,2) level lies above the (3,1) state. Finally the $\Omega = 3$ state could be the (3,3) or the (4,3) state. A ligand field fit suggests that this level is the (4,3) state.

All results pertaining to the energy level diagram are compiled in Table 7-XXIIfor the $v = 0$ levels of the various lower electronic states.

According to the ligand field fit, the $\Omega = 0^+$ state at 4637 cm^{-1} cannot belong to the f^{14} or $f^{13}s$ the configurations. Therefore it must be coming from another configuration, possibly $f^{13}d$.

The other states have been fit to the ligand field Hamiltonian derived in chapter 4 for a $f^{13}s$ configuration. There are six parameters to be fit

Table 7-XXII: Observed Lower Energy Levels of YbO

Name (J_a.Ω)	Energy Above Ground State(cm^{-1})	Fluorescence Bands(Å)
(0.0^+)	0	5793. 6032. 6067. 4518. 4066.
(4.0^-)	910	4940. 4803. 4712
(4.1)	1015	5424. 4829. 4734
(4.2)	2408	5070. 5352
(3.1)	2702	5424. 5254. 5143. 5439
(4.3)	4287	5929. 5733. 5952
(?.0^+)	4637	5715

1. $B_0^0(4f) - B_0^0(6s)$ for the separation between the f^{14} and $f^{13}s$ configurations
2. $G_3(4f,6s)$ the exchange term for the 4f and 6s orbitals
3. ς_{4f} the spin-orbit term for the 4f electrons
4. B_0^2, B_0^4 and B_0^6 the ligand field parameters.

I will not be able to determine the spin-orbit interaction because this term is the same for all the levels I have detected. Its value will be set to the free ion value of 2916 cm^{-1} [6]. I am then left with six levels, five differences which will be fit, and five constants. To force the fit to be more than an empirical fit, the ratios of the ligand field parameters will be locked to a theoretical estimate. A Hartree-Fock calculation to determine the radial behavior of the electrons produced the following values for the ligand field parameters [7]

$B_0^0(4f) = 129{,}045\ cm^{-1}$

$B_0^0(6s) = 118{,}920\ cm^{-1}$

$B_0^2(4f^{13}6s) = 7003\ cm^{-1}$

$B_0^4(4f^{13}6s) = 1068\ cm^{-1}$

$B_0^6(4f^{13}6s) = 382\ cm^{-1}$

The fit was done such that $B_0^4/B_0^2 = 1068/7003$ and $B_0^6/B_0^2 = 382/7003$. The exchange term, $G_3(4f,6s)$, was not allowed to vary in the fit since it is highly correlated to the ligand field parameters. The separation of the (4,2) and (3,1) levels in the molecule is 294 cm^{-1} quite close to the separation in the atom of 334 cm^{-1}, indicating that $G_3(4f,6s)$ is almost the same as in the atom. Consequently the exchange parameter in the molecule was set to the atomic exchange parameter = 341 cm^{-1}. This reduced the number of fitting parameters to two. The results of the fit are shown in Table 7-XXIII.

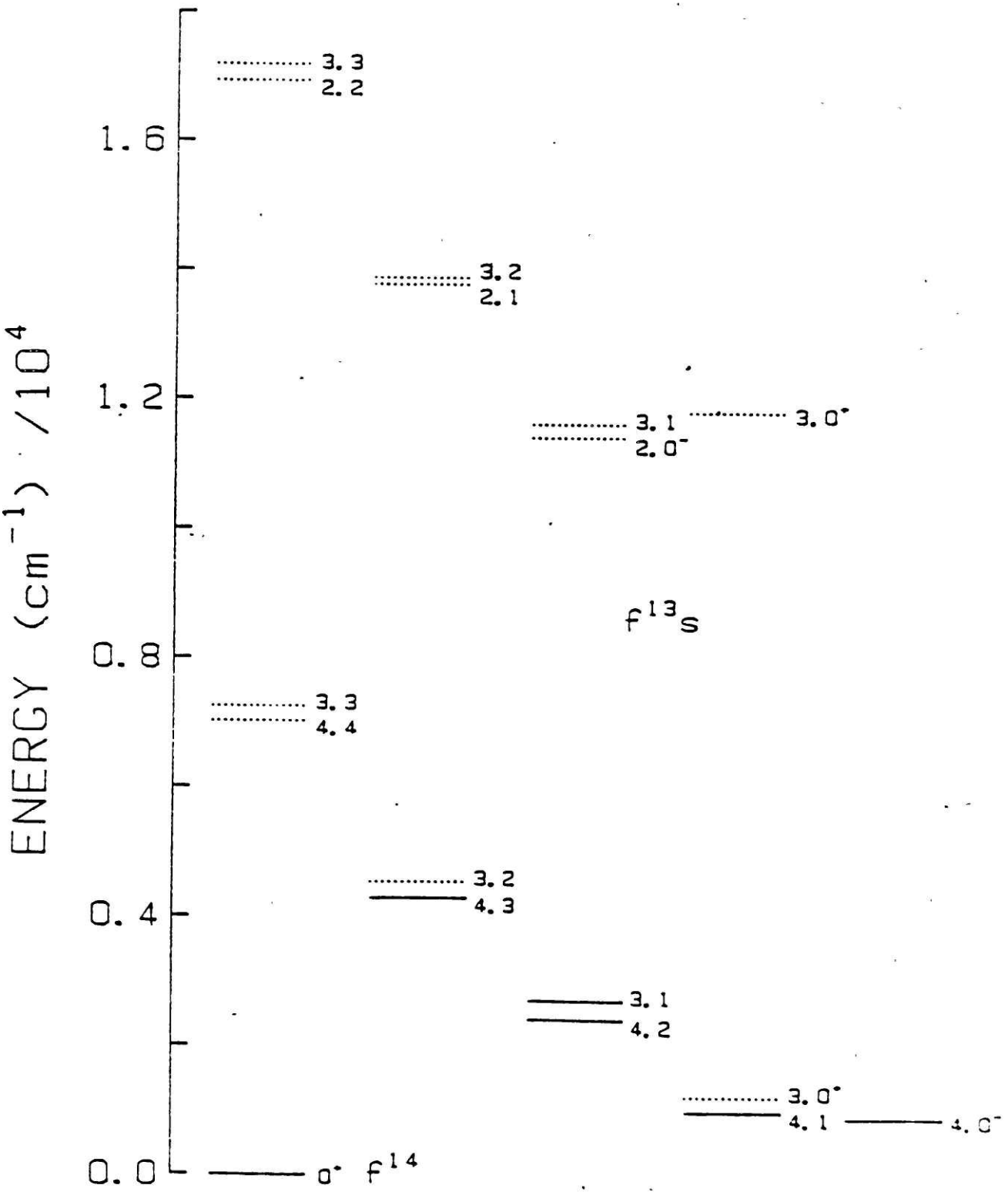
Table 7-XXIII: YbO Ligand Field Fit

Parameter	Value (cm^{-1})	Uncertainty
$E(4f^{13}6s)-E(4f^{14})$	8370	70
ζ_{4f}	2899	
G_3	300	
B_0^2	10,300	360
B_0^4	1,070	
B_0^6	380	

The fit to the data is very good, showing that the ligand field model works well. Figure 7-3 shows the energy level diagram for the f^{14} and $f^{13}s$ configurations of YbO.

In the ion, the separation of the centers of gravity of the $4f^{13}6s$ and $4f^{14}$ configurations is 39,369 cm^{-1}. In the molecule this is shifted by ΔB_0^0 to give a predicted separation of 29,244 cm^{-1}. The fit shows a separation of 8370 cm^{-1}, indicating that the $f^{13}s$ configuration is stabilized much more than predicted

Figure 7-4: Energy Level Diagram for f^{14} and $f^{13}s$ Configuration in YbO



by Ligand Field Theory. The extra stabilization is probably due to the s/p hybridization of the Yb^{2+} 6s orbital similar to that found for the $X^2\Sigma^+$ states of the alkaline earth halides [8].

Although much less is known about the upper levels than the lower in the transitions, some statements can be made. I will discuss what is known about the upper states for some of the laser transitions.

The two upper states of the 5793, 6032 and 6067 Å bands probably come from a configuration which can be written as M^+O^-. This statement is based on the low ω_e ($\sim$550 cm^{-1}) values.

An interesting facet of the 5929, 5925, 6021 and 4518 Å bands is that $\Delta\Omega = \pm 1$ for all but one of the fluorescence bands, and that band is quite weak. One possible explanation starts with both the upper and lower state having the same J_F, so that the transtion involves the non-f electron. In the lower state this is an s, or really a σ electron. The upper state will have both σ and π orbitals, but the two should have different energies. Perhaps the gap between the two is such that the energy levels probed with a rhodamine 6G laser coming from the (4,3) level are the π states. Then the transition would be from a π upper state to a σ lower state, always giving $\Delta\Omega = \pm 1$.

All of the states excited in the rhodamine 6G region, except those with f^{14} as the lower state, exhibited fluorescence which extended far to the blue. There are not any transitions involving the lower energy states in the $f^{13}s$ configuration. Perhaps there are not any levels which have transtion strength to the $f^{13}s$ configuration until about 21,000 cm^{-1} above the ground state.

To further understand the electronic structure of YbO, several things

could be done. The first would be to do a ligand field calculation which included configuration interaction. Such a calculation could explain the difference between the fitted and calculated values of ΔB_0^0 and B_0^2. It could also identify the unknown 0^+ state observed in .5715 Å fluorescence band. Since the low lying levels are now well understood, additional experimental work could be done on the unraveling the upper states. However the high density of states would make this a formidable and time consuming project.

7.11 Conclusion

Ligand Field Theory has successfully predicted the ground state of YbO and the ordering of the low energy states. However the failure of the simple ligand field calculation to correctly predict the separation states of the f^{14} and $f^{13}s$ configuration indicates that a more sophisticated calculation is necessary. The next step will be a calculation that accounts for configuration interactions with the inter-configurational ligand field matrix elements.

References

[1] P. W. Atkins.
Molecular Quantum Mechanics.
Oxford University Press, New York, 1983.

[2] J. M. Brown and B. J. Howard.
An Approach to the Anomalous Commutation Relations of Rotational Angular Momenta in Molecules.
Mol. Phys. 31(5):1517, 1976.

[3] A. Carrington, D. H. Levy and T. A. Miller.
Electron Resonance of Gaseous Diatomic Molecules.
Adv. Chem. Phys. 18:149, 1970.

[4] G. Herzberg.
Molecular Spectra and Molecular Structrue.
Van Nostrand Reinhold Company, New York, 1950.

[5] J. T. Hougen.
The Calculation of Rotational Energy Levels and Rotational Line Intensities in Diatomic Molecules.
NBS Monograph 115, National Bureau of Standards, June, 1970.

[6] W. C. Martin, R. Zalubas and L. Hagen.
Atomic Energy Levels-The Rare-Earth Elements.
Technical Report NSRDS-NBS 60, Institute for Basic Standards. National Bureau of Standards, 1978.

[7] Michael Dulick, private communication.

[8] S. F. Rice, H. Martin and R. W. Field.
The Electronic Structure of the Calcium Halides. A Ligand Field Approach.
J. Chem. Phys., accepted , 1985.

Chapter 8
The Future of Ligand Field Theory

8.1 Conclusions

I'd like to review the record of Ligand Field Theory in predicting the properties of EuO and YbO. The theory has correctly predicted the ground states of both molecules. Futhermore it has successfully ordered the low lying states of YbO, strongly indicating that the states do come from the $f^{13}s$ configuration. The spin-spin constant of the ground state of EuO can be explained in part with a second order spin-orbit perturbation using a basis set from the Eu^{2+} ion, a result in agreement with the theory. This evidence shows the ability of Ligand Field Theory to predict many molecular properties, some of them quite subtle, of these extremely complex molecules. Because of strong perturbations, the hyperfine splitting observed in the EuO transition could not be assigned to either the upper or lower state, preventing a test of the predicted zero splitting for the ground state due to its f^{14} parent configuration. The failure of a simple ligand field calculation to accurately predict the separation of the f^{14} and $f^{13}s$ configurations in YbO indicates the need for a more sophisticated calculation.

The work on EuO and YbO has shown how useful the ligand field model can be as a qualitative aid in understanding the spectra. However, the theory has been extended and is proving useful as a quantitative tool. The main extension is a method of including configuration interaction within the

framework of the theory. As discussed for EuO and YbO, only the ligand field interactions within a single configuration were used in calculating the energy levels of the molecule. However, there are also ligand field matrix elements between configurations which differ by a single orbital, and their inclusion in calculations dramatically improves the results.

As a demonstration of the power and general applicabilty of Ligand Field Theory I'm going to review some other molecules for which it has been used successfully. Many of these applications test properties which depend on the parent atomic ion electron configuration, such as the hyperfine or Zeeman splitting. The success of the theory in these cases suggests that the ligand field approach is a valid model.

In LaF the relative energies of the the s^2 and ds configurations were unknown. Ligand Field Theory, using the configuration interaction mechanism discussed above, predicts that the s^2 state will have the lowest energy. Recent spectroscopic work has shown that there is a $^1\Sigma$ state lower in energy than a $^3\delta$ state. [6]

Many of the CaX electronic properties, excited state energies, dissociation energies, permanent dipole moments and transition moments can be explained in terms of Ligand Field Theory [5].

The CeO and PrO ground states and the hyperfine splitting of the ground state of PrO can be fit with the theory [1], [3], [2].

The ground state of HoO could come from the f^{11} or the $f^{10}s$ configuration. Ligand Field Theory predicts $f^{10}s$ as the ground state configruation which is supported by the large hyperfine splitting of the ground state and by the pattern of the Ω values of the low lying states. [7]

The predicted Zeeman splitting for levels of CeO has been confirmed experimentally.

LaO wavefunctions calculated with Ligand Field Theory successfully predict the hyperfine splitting in LaO [4].

This list demonstrates the applicability of Ligand Field Theory to the calcium monohalides and the lanthanide monoxide and mono halide molecules. The theory is now being used to make predictions for the transition metal hydrides. Experimental results thus far are promising.

References

[1] M. Dulick.
Spectroscopic Studies of the Diatomic Rare-Earth Oxides.
PhD thesis, M. I. T., 1982.

[2] M. Dulick and R. W. Field.
Electronic Structure of PrO. An Analysis Summary.
J. Mol. Spec., accepted , .

[3] C. Linton, M. Dulick, R. W. Field, P. Carette, P. C. Leyland and R. F. Barrow.
Electronic States of the CeO Molecule. Absorption, Emission and Laser Spectroscopy.
J. Mol. Spec. 102:441, 1983.

[4] Michael Dulick, private communication.

[5] S. F. Rice, H. Martin and R. W. Field.
The Electronic Structure of the Calcium Halides. A Ligand Field Approach.
J. Chem. Phys., accepted , 1985.

[6] H. Schall, C. Linton and R. W. Field.
Laser Spectroscopy of LaF: Determination of the Separation of the Singlet and Triplet State Manifolds.
J. Mol. Spec. 100:437, 1983.

[7] Y. C. Liu, H. Schall, R. W. Field and C. Linton.
Laser Spectroscopy of HoO: Examination of the Low-Lying States.
J. Mol. Spec. 104:72, 1984.

Appendix A

Programs for Data Collection

Scanning Programs
These are the programs written to scan the laser and collect data. I wrote these programs a long time ago and have made many changes, so it's possible there could be problems in trying to fit this particular set together.

SCNCOM.C
FORMAT

```
scncom()
/*This is the root segment which calls all the programs
that scan the cw lasers.  Root segment means that this part of
the scanning program always stays in core and calls the other
programs which may have to be read off the disk (see system
users manual under overlays if you're lost).  First of all the
global variables have to be declared, since they will also remain in
core at all time.  Then the various other programs are called which
do the dirty work.
*/

#include    "sy:graph.lib"
#include    <sy:std.h>
#include    "sy:sam.h"      /* contains the structure that contains many
                               of the scan parameters */
#define     ADACAD  2           /* this tells whether the ADAC a/d is in the
                               unipolar (measures Ø volts to +V volts) or
                               bipolar (measures -V to +V).  See defini-
                               tion of adoffset below to see why this
                               makes a difference */
#define     ADACØDA 2           /* same deal as above.  If ADACØDA = 1 then
                               d/a #Ø is in unipolar mode, if ADACØDA =
                               2 then in bipolar mode */
```

```
/*   THE GLOBAL VARIABLES */

struct      head1   h1 = {};
                    /* this is a structure that contains the information
                       about the scan that is written into the first
                       block of the scan.  For more information about the
                       structure, see the file QANDA.C */
int adoffset = 0;   /* I want the data from the a/d to behave in the
                       following way: for the lowest possible input
                       voltage the a/d should return a zero.  For the
                       highest a 4095.  For example if the a/d is set for
                       -5 to +5 volts, then -5 should be 0 and +5 should
                       be 4095.  The digital a/d works this way.  However
                       the ADAC a/d works this way only when set to
                       voltages from 0 to some voltage V (unipolar
                       operation) and does not work this way for -V to
                       +V (bipolar operation).  The ADAC returns a 04000
                       for -V, a 0000 for 0v and a 03777 for +V.  The
                       simple solution is to add in a 04000 and mask out
                       the upper 4 bits.  If ADACAD = 2 (bipolar) and
                       the a/d in the system is an ADAC, that's
                       just what will be done (adoffset = 04000 = 2048
                       decimal).  If ADACAD = 1 or if there is a digital
                       a/d in the system, adoffset = 0 so there is no
                       net change to the data when adoffset is added to
                       it */
int *b1 = 0;        /* the buffer which the interrupt will
                       place the data in  */
int *b1max = 0;     /* a pointer to the space after the highest
                       location of buffer b1 */
int *b1ir = 0;      /* a movable pointer which the interrupt
                       routine uses to fill the buffer b1 with
                       data from the a/d's. */
int *b1mp = 0;      /* a movable pointer which the main program
                       (as opposed to the interrupt routine) uses
                       to transfer data from b1 to b2 or b3. */
```

```
int *b2 = 0;          /* one of the two buffers which are an
                         intermediate between b1 and the disk */
int *b2max = 0;       /* a pointer to the word location after the
                         highest word of the buffer b2 */
int *b3 = 0;          /* the second of the two intermediate
                         buffers */
int *b3max = 0;       /* a pointer to the word location after the
                         highest word of the buffer b3 */
int *bin = 0;         /* this is either b2 or b3, whichever happens
                         to be taking the data from b1.  The other
                         buffer will be in the process of having
                         its data read to the disk  */
int *binmax = 0;      /* a pointer to the location following the
                         last word in whichever of b2 or b3 is
                         the input buffer */
int *binp = 0;        /* a pointer to the spot in the input buffer
                         that the next data obtained from b1 will
                         be placed  */
int blk = 0;          /* this is the block in the disk file that the
                         computer is actually writing data to at a
                         given time */
int *bout = 0;        /* this is either b2 or b3, whichever buffer
                         is having its data read to the disk */
int *boutmax = 0;     /* the maximum of the output buffer i.e. a
                         pointer which points to the last element
                         in bout*/
int chan = 0;         /* this is the RT-11 channel number of the new
                         data file that will be created   */
int clock = 0;        /* a flag that tells whether there is a real time
                         clock (rtc) in the system.  If clock = 0, no
                         rtc; if clock = 1, there is a rtc*/
int dabegin = 0;      /* the number needed to set the d/a to its lowest
                         possible voltage.. =0 for the digital and for
                         ADAC in unipolar mode, =1 for ADAC in bipolar
                         mode */
int datflg = 0;       /* this is a flag that is set by the interrupt
                         routine when it takes data and cleared by
                         digest() when it reads data */
```

```
int gain[16] = {};  /* Contains information to be loaded into the A/D
                       command and status register when the interrupt
                       routine (ADIR.MAC) is taking in data.  For a
                       description of what each bit in the ADCSR
                       means, see the ADAC 1012 manual or the Digital
                       documentation on the A/D and preamp, depending
                       on which board you're using. */
int dabuf = 0;      /* this is the number loaded into the d/a buffers
                       by the routine adir.mac in order to scan the
                       laser.  */
int scanon = 0;     /* this is a flag which tells the interrupt routines
                       whether to take data (scanon = 1) or not to take
                       data (scanon = 0) */


_main()
{
    extern  struct  head1   h1;     /* this is the structure that is
                                       written into the first block of
                                       each scan*/
    extern  int     adoffset;       /*see above difinition */
    extern  int     dabuf;
    extern  int     clock;          /* hardse() will set to tell if a
                                       real time clock is in the system*/
    extern  int     dabegin;        /* hardse() will set this to the
                                       value needed to set the d/a to
                                       it's lowest value */
    extern  int     gain[];         /* see above definition */
    extern  int     scanon;         /* a flag that tells the interrupt
                                       routines to take data (scanon = 1)
                                       or not to take data (scanon = 0) */
    int     *pi;                    /* general purpose integer pointer*/
    char    c;
    int     mmu;                    /* flag, hardse sets to 1 if memory
                                       management and floating point
                                       processor are in the system,
                                       0 if not */
```

```
   scanon = 0;              /* just in case, make sure don't try to
                               scan */
   if (hardse (gain, &clock, &mmu, &dabegin, &adoffset, ADACAD, ADAC0DA)\
< 0)
           c = 'N';
   else
           c = 'Y';
   /* hardse returns a -1 if a critical piece of equipment
      is missing.  If not it sets up gain, clock, dabegin and
      adoffset */
   if (mmu == 1)
           setd();          /* if have floating point processor, set it
                               to do double arithmetic. */
   while (toupper (c) == 'Y')
           {
           printm (CLRSCR);         /* clear screen */
           printm (CUR_HOME);       /* send cursor home */
           header();        /* header prompts the user for the scan
                               parameters and creates a data file on
                               the disk */
           printm (CLRSCR);/* clear the screen */
           printm (CUR_HOME);       /* send cursor home */
           takdat();        /* takdat reads the scan parameters
                               (they're global variables), sets up the
                               clock and takes the scan. */
           clearl();
           printm ("\nDo you want to take another scan (Y or N)?\n");
           c = ttyin();
           clearl();
           }
}
```

HARDSE.C

```
/*  This program is called by scncom to set up for possible hardware
configurations of the system.  hardset() checks for 6 hardware items:
```

(1) terminal — must be using either a VT105 or a VT55 for the graphics to work properly.

(2) a/d — can have either a digital analog to digital converter or an ADAC a/d.

(3) d/a — can have either a digital d/a or an ADAC

(4) clock — either there is a KWV11-A in the system or there is not.

(5 & 6) memory management and floating point processor — if the machine is an LSI-11/23 then it will probably have a memory management chip and a floating point processor. If not, it won't.

hardset finds out what items are in the system and their make by calling hardware(). It then uses the information in the following ways:

terminal: if the terminal is a VT55 or a VT105 in VT52 mode, everything is o.k. If the VT105 is in ANSI mode, hardset() puts it into VT52 mode. If the terminal isn't a VT55 or a VT105, hardset() returns a -1 to signal a fatal lack of hardware.

a/d: the buffer gain(), which is passed from scncom, is set up differently for the ADAC and the digital board. Basically the gain for the ADAC preamp must be set in the CSR while the digital does not have a preamp. Also the ADAC a/d in bipolar mode (measures from -V to +V) will return the data with an undesired offset. I compensate for this by adding the offset in the interrupt routine. hardse() initilizes the offset.
If an a/d does not exist, fatal lack of hardware.

d/a: the starting number to start the scan differs for the digital board and the ADAC board. hardset() sets this number (*dabegin). If no d/a, fatal hardware error.

clock: hardset() simply returns a flag telling the calling program

whether a real time clock is on the system or not.

mm and fpp hardset() returns a flag telling the calling program whether there is a memory management chip

ARGUMENTS: gain this is a buffer of 16 integers that is loaded into the adcsr by the routine adir.mac in order to trigger the a/d

*clock the adress of a flag. *clock = 0 if no real time clock on the system, = 1 if there is a real time clock (i.e. something at CLKCSR responded. See the file ADRESS.LIB for the location of CLKCSR).

*mmu the address of a flag. = 1 if memory management and floating point processor appear to be present, =0 otherwise.

*dabegin the address of the integer that will be loaded into DAC0 to start the scan. dabegin=174000 for the ADAC in bipolar mode, =0 for the digital d/a or for the ADAC in unipolar mode.

*adoffset the adress of the offset that will be added to all numbers taken from the a/d. For a digital or ADAC a/d in unipolar mode this number is zero. For the ADAC in bipolar mode this number is -2048 decimal or 04000.

adacad looks like gibbersh but is meant to represent ADAC a/d. If = 2 then ADAC is in bipolar mode, if = 1 then a/d is in unipolar mode. Affects the setting of adoffset when an ADAC a/d is in the system

adac0da stands for ADAC d/a #0. If = 2 then d/a is in the bipolar mode, if = 1 then d/a is in the unipolar mode. Affects setting of dabegin.

RETURNS: 1 if everything o.k., -1 if wrong terminal or no a/d or d/a

```
                    on system.

*/

hardset (gain, clock, mmu, dabegin, adoffset, adacad, adac0da)
int gain[];
int *clock;
int *mmu;
int *dabegin;
int *adoffset;
int adacad;
int adac0da;
    {
    int     termid;
    int     atod;
    int     dtoa;
    int     i,j;
    int     error;  /* this is the flag returned to the calling program*/

    error = 1;
    hardware (&termid, &atod, &dtoa, clock, mmu);
    if (termid < 1  ||  termid > 3)
            {
            putfmt ("You must have a VT55 or VT105 terminal.\n");
            error = -1;
            }
    else if (termid == 2)
            {               /* VT105 in ANSI mode so put into VT52 mode*/
            printm ("\033[?21");    /* put into VT52 mode for graphics */
            termid = termtype();
            if (termid != 1)
                    {
                    putfmt ("VT105 will not go into VT52 mode.\n");
                    error = -1;
                    }
            }
    if (atod == 0)
```

```
            {
            putfmt ("You must have an analog to digital converter in \
the computer.\n");
            error = -1;
            }
    else
            {
            /* now I want to set up gain[] */
            for (i = 0; i < 16; i++)
                    gain[i] = (i<<8) + 1;   /* this will go into the a/d
                                               csr.It is the multiplexer
                                               channel # */
            if (atod == 2)
                    {
                    /* have an ADAC a/d so have to set preamp gain */
                    for (i = 0; i < 16; i++)
                            gain[i] = gain[i] | 030;
                    /* now set up offset for ADAC board */
                    if (adacad == 2)
                            *adoffset = -2048;      /* ADAC board in
                                                       bipolar mode*/
                    else
                            *adoffset = 0;          /* ADAC in unipolar
                                                       mode */
                    }
            else
                    adoffset = 0;   /* digital board so set offset to 0*/
            }
     if (dtoa == 0)
            {
            putfmt ("You must have a digital to analog converter in the \
computer.\n");
            error = -1;
            }
     else
            {
            if (dtoa == 1)
                    *dabegin = 0;   /* digital board */
```

```
        else
                if (adac0da == 2)
                        *dabegin = -2048;       /*bipolar ADAC board*/
                else
                        *dabegin = 0;           /* unipolar ADAC */
        }
    return (error);
    }
```

TERMTY.C

```
/*
    This program is to tell the calling what type of terminal is being
used.  It uses the DEC identify terminal command, ESC Z.  A VT55 will respond
with an ESC/E, a VT105 with a ESC[?1;6c if it is in ANSI mode, and an ESC/Z
if it is in VT52 mode, where ESC is an octal 33.  There are no spaces in the
response.

ARGUMENTS:  none
RETURNS:    1 if the terminal is a VT55
            2 if the terminal is a VT105 in VT52 mode
            3 if the terminal is a VT105 in ANSI mode
            0 if the terminal is not one of the above
*/


#include    "sy:adress.lib"
#define             MASK    010000

termtype()
{
    int     i, j;
    int     *jsw;
    char    cbuff[20];
    char    c;
    int     termid;
    int     nchar;
```

```
    /* need to go into special terminal mode so that characters sent
       from the terminal are available immediately to the program */
    jsw = JSW;
    *jsw = *jsw | MASK;      /* put into special term mode */
    /* Now make sure there that there aren't any other characters in
       the I/O buffer that havn't been responded to yet */
    while (ttinr() > 0)
            ;
    /* now send command to terminal telling it to identify itself */
    ttyout ('\033');         /* put out escape */
    ttyout ('Z');            /* put out Z to terminal */
    /* now get the terminal's response.  */
    /* the for loop is to make sure the terminal has time to respond */
    for (nchar = 0, i = 0; i < 100; i++)
            {
            if ((j = ttinr()) > 0)
                    {
                    cbuff[nchar++] = j;
                    i = 0;
                    }
            }
    if (nchar == 0)
            termid = 0;
    cbuff[nchar] = '\0';     /* make it into a string */
    if (nchar == 3  && cmpstr (cbuff, "\033\057\105"))
            termid = 1;              /* VT55 */
    else if (nchar == 3 && cmpstr (cbuff, "\033\057\132"))
            termid = 2;              /* VT105  in VT52 mode */
    else if (nchar == 7 && cmpstr (cbuff, "\033[?1;6c"))
            termid = 3;              /* VT105 in ANSI mode */
    else
            termid = -1;
    *jsw = *jsw & !MASK;     /* reset job status word */
    return (termid);
}
```

```
.title      setd
;   This routine puts the floating point processor into double
;precision mode.  Don't ask me why the damn compiler doesn't do
;it but in at least one case it did not.  Therefore for safety's
;sake, call this routine at the beginning of a program which will do
;double precision stuff.  Of course you cannot use this program on a
;machine without the floation point processor.  To check, see if the
;machine responds to locations occupied by the memory management chip.
;Since the floating point registers are contained in the mm chip, if
;it's not there, there isn't any memory management. */


;ARGUMENTS: none

;RETURNS:   nothing of any particular interest


SETD::      setd
    rts     pc

.end
```

HEADER.C

```
/*  This program (header.c) along with qanda are designed to query
   the user and find out the scan parameters for a computer controlled
   scan.  They also open the data file and write the user supplied infor-
   mation in ASCII just before the binary scan data.

    Normally there will be a header file from the last scan already on
   the default disk.  It is called SETUP.SCN.  It contains the scan infor-
   maiton supplied by the last user.  The first block (256 words) has
   the scan parameters in binary, so that the computer can easily access
   them.  The next 15 blocks are available for the scan parameters plus other
   scan information in ASCII so that a person can easily access them.  DO NOT
   TAMPER WITH THE SCAN PARAMETERS BY EDITING THE ASCII INFORMATION ONLY!!!
   Not only is it dishonest, but the computer won't know what you've done
   and will use the old data.  On the other hand, if you really want to
```

```
   screw someone else up, tamper with their binary header without letting
   them know by editing their ASCII header.  In other words don't do any-
   thing to scan parameters in a header file, except with this program,
   unless you know just what you're doing.

    Anyway, header.c takes the old file (SETUP.SCN), fills in the
   scan parameters with the binary data, and asks the user if he wants
   to change anything.
   qanda.c is then called to pose the questions and record the answers
   in a new header file pointed to by aout.  At the end of the questioning
   a new SETUP.SCN file is created with the new header and a new data
   file is opened also with the exact same new header.

*/
#include    "sy:prog.req"
#include    <sy:std.h>
#include    "sy:sam.h"
#define             BSIZE   8192    /* size of the input and output buffers in
                                       bytes  */

header()
    {
    int     f,i;            /* general purpose integers */
    char    c;              /* general purpose character*/
    char    *pc;            /* general purpose character pointer */
    int     *pi;            /* general purpose integer pointer */
    float   *pf;            /* general purpose floating point pointer*/

    int     change[40];     /* this is the array that contains the
                               line numbers of the items the user
                               choses to change */
    char    *aout;          /* this is a pointer to the beginning of
                               the output header file, i.e. after
                               editing  */
    char    *ain;           /* this is a pointer to the beginning of
                               the old header file, the one you're
                               about to change  */
    int     size;           /* the size of the header file created in
```

```
                        qanda() in bytes */
int     schan;          /* this is the channel number of the new
                           setup.scn file that will be created
                        */
char    *oldfil = "SETUP.SCN";
                        /* this is the name of the temporary
                           header file */
extern  struct  head1   h1;     /* this structure contains all the
                                   variables that are written in the
                                   first block of each scan.*/
extern  int     chan;   /* this is the RT-11 channel number of the new
                           data file that will be created   */
static  char    firbuf[BSIZE];
static  char    secbuf[BSIZE];



chan = -1;              /* this is to make sure that originally chan
                           is not a legal channel number */
ain = firbuf;
aout = secbuf;
for (i = 0; i < 512; i++)
        *(ain + i) = *(aout + i) = '\0';
for (; i < BSIZE; i++)
        *(ain + i) = *(aout + i) = ' ';
/* the two above for loops put nulls in the binary section of the
   header file and blanks in the ascii portion */
*(aout + BSIZE - 1) = EOF;      /* this is so qanda() knows where
                                   the output buffer ends */
schan = reopen (oldfil);        /* reopen is a SAMLIB routine that
                                   reopens an existing file and
                                   returns the channel or an error
                                   flag */
if (schan < 0)
        {
        if (schan == -1)
                {
                putfmt ("No RT-11 I/O channels are available.  Exit the \program and fix the problem\n");
```

-212-

```
                return(-1);
                }
        else    {
                putfmt (" %p%p ",\
"There is no old header file (SETUP.SCN) on the default storage disk.\n",\
"You must create a new one.\n");
                for (i = 0; i < 39; i++)        /*set change[] such
                                                that all options in qanda
                                                will be asked */
                        change[i] = i+1;

                *(ain + 512 + 17) = '\0';
                *(ain + 512 + 18) = EOF;
                /* the above three assignments set up an EOF
                   qanda() will recognize */
                size = qanda (ain,aout,change,&schan);
/* qanda is the routine that asks the questions  */
                }
        }
    else    {               /*if there was an old file, do the following*/
        READM(schan,0,ain,BSIZE/2);     /* read the old file in */
        WAITM (schan);                  /* wait until finished */
        CLOSEM (schan);                 /* close old file */
/* Next the scan parameters which effect what the program will do later
   on must be read in.  They are all passed to other parts of the program
   as external or global variables.
*/
        cpybuf (&h1, ain, sizeof(h1));  /* copy over old variables*/
        /* now edit the old file */
/* the first thing to do is display the old file so the user
knows what numbers correspond to which parameters */
        putfmt ("\n%.3p","1. ");
        for ( i = 512, f = 2; *(ain + i) != EOF; i++)
                {
                if (*(ain + i) != '\0')
                        putch(*(ain + i));
                else
                        putfmt ("%.2i. ",f++);
```

```
                }
/* the above for loop prints the ASCII part of the input file with one
modification, it numbers each parameter with f.  A parameter is noted by
being followed (or led if you prefer) by a null:'\0\'.  This is the
same way qanda will read the file.  The 512 added to ain is the number of
bytes from the beginning of the file to the beginning of the ASCII header.
    For more information on the construction of the ASCII header,
see the subroutine qanda.
*/
            putfmt ("Type in this number if you want to \
change all the parameters.\n\n");
/* note that the for loop will type one more line number than there
are lines.  The above putfmt statement makes use of that extra line. */
            putfmt ("%p%p%p%p",\
"Enter the numbers of the parameters you wish to change as integers.\n",\
"Seperate them by any non digit you wish.  After the last entry, hit the\n",\
"carriage return.\n",\
"Which parameters do you wish to change?\n");
            i = getint (&change[0], 38);/* a SAMLIB routine */
                                        /* getint returns the number
                                           of integers entered */
            bubble(change,i);           /* bubble is a SAMLIB
                                           routine which puts the
                                           numbers in change in
                                           ascending order */
            if (change[i-1] == f-1)
                    {       /* if the user wants to change all
                               the parameters, set change[] up*/
                    for (i = 0; i < 39; i++)
                            change[i] = i + 1;
                    }
            change[i] = 0;
            size = qanda(ain,aout,change,&schan);
            }

/* So by now the file has been edited once.  It's time to redit if the user
wants to */
```

```
    for (;;)        /* keep rediting until the user says not to */
            {
            putfmt ("\nDo you want to check the file? (Y or N)\n");
            c = ttyin();
            clearl();               /* a SAMLIB routine, it clears the
                                       rest of the characters in the line
                                       out of the standard input buffer */
            if (c == 'N'  ||  c == 'n')
                    break;
            /* type the file out */
            putfmt ("\n%.3p","1. ");
            for (i = 512, f = 2; *(aout+i) != EOF; i++)
                    {
                    if (*(aout + i) != '\0')
                            putch(*(aout+i));
                    else
                            putfmt ("%.2i. ",f++);
                    }
            putfmt ("Type this number if you want to change all the \
parameters.\n\n");
                    putfmt ("%p%p%p%p%p",\
"Enter the numbers of the parameters you wish to change as integers.\n",\
"Seperate them by any non digit you wish.  After the last entry, hit the\n",\
"carriage return.  Enter each integer only once.\n",\
"Type a -1 if you don't want to change any parameters.\n\n",\
"Which parameters do you wish to change?\n");
            i = getint (change, 39);
            bubble (change,i);
            if (change[0] == -1)
                    break;
            if (change[i-1] == f-1)
                    {       /* if the user wants to change all
                               the parameters, set change[] up*/
                    for (i = 0; i < 39; i++)
                            change[i] = i + 1;
                    }
            change[i] = 0;  /* set the last integer to 0 */
            pc = ain;
```

```
        ain = aout;
        aout = pc;
        for (i = 512; *(aout + i) != EOF; i++)
                ;
        *(aout + i--) = ' ';
        *(aout + i) = ' ';
        /* since out was the input file, it has an '\0'EOF where
           it's not supposed to.  Have to find it and clear it
           and then (in the next command) reset the new EOF to the
           the end of the file */
        *(aout + BSIZE - 1) = EOF;      /* reset the EOF to the last
                                           available space in the
                                           file */
        PURGEM (chan);
        /* since the user is rediting the program, the old channel
           has to be closed before qanda() opens a new one */
        size = qanda(ain,aout,change,&schan);
        }
    WRITEM(schan,0,aout,(size/2 +1));
    WRITEM(chan,0,aout,(size/2 +1));
    WAITM (schan);
    WAITM (chan);
    CLOSEM(schan);  /* close the header file */
    return(0);
    }
```

QANDA.C

```
/*  This program is used with header() to initilize the scan parameters
for a computer controlled scan and to create a binary and ASCII header file
for the data taken by the scan.  qanda(), short for questions and answers
actually asks the questions and creates the header file.  header() sets
up the necessary buffers for qanda().
    qanda takes an old header file, whose starting location is given
by ain, and creates a new header file, whose starting location is given by
aout.  The files consists of a bunch of parameters such as scan time, data
file name, or comments.  The user has already chosen which parameters he
wishes to change from their value in the old header file (this is done
```

in header()). These changes are passed as an array of integers in change[].
    header() and qanda are designed so that it
is extremely easy to change the components in the header file. The construction of a header file is as follows:
    The first 256 words are the scan parameters in binary that the computer will need to know at a later time. To see the format of these variables see the structure definition for head1 in the file sy:sam.h.
    The rest of the 256 words are empty so that other scan parameters can be added as necessary.
    After this comes the ASCII header file. This is all the scan parameters and comments in ASCII. Therefore this part of the file can be used as a notebook to write down all important information.
    The structure of the ASCII portion is as follows:
DESCRIPTOR: user supplied answers
a carrage return line feed
following the line feed is a NULL, a binary 0.
DESCRIPTOR: etc. until the end of the file is reached
    At the end of the header file is an end of file, EOF.

    The Descriptor is provided by qanda(). The user supplied answers are typed in by the user in response to a question from qanda(). The null marks the end of a parameter and is a very important part of the file since qanda uses that null to tell where each scan parameter begins.
    As I said, the changes the user wants to make are specified by the integers in change[] from header(). Just assume this has already been done. qanda() checks to see whether it's supposed to change a parameter by comparing testing the elements of change with the following statement:

```
    if(change[j] == k++)
            change the parameter
            increment j (so the next element of change is tested)
    else    since you don't want to change the parameter, copy the
            old one
    if(change[j] == k++)  test for the next parameter
            etc.
```

j is the index of change[] which is incremented everytime qanda() changes something
k is a counter for each of the parameters.
    Instead of saying

```
        if(change[j] == 1)
                change the file name
        etc.
```

which would have been inflexible, I chose to use a running index k which is initilized at 1 and incremented with each test. This allows each section of code which is concerned with only one parameter to be relocated to any position the user desires. It also makes insertion of new parameters very simple. Note that some paramater modification modules do have to be located in a certain order relative to each other and one, the file name module, must always be the last one. An example of the relative ordering restriction is that the number of channels must always be determined before identity of the channels can be asked for the identity module has to know how many channels it needs to ask about.

Now for the skeleton of each module. "out" is the pointer to the present location in the output file, "in" is the location of the pointer to present location in the input file.

```
    if(change[j] == k++)
    {
            if the above statement is true, the user has chosen to
            alter the parameter which this particular module deals
            with.  At this part of the code, there should be something
            that writes the descriptor in the output file.
            for(;;)
                    {
                    I usually use an infinite for loop for the question
                    and the users response to the question.  If the
                    response is reasonable, a break is executed and the
                    result is written in the output file.  If the response
                    is unreasonable, the user is given a prompt and
                    queried again.
                    if(response is reasonable)
                            break;
                    otherwise give prompt
                    }
            now the response must be written in the output file and the
            pointer out must be updated so that it points to the next
```

```
        empty byte of the output file.  Also note that the last
        three characters of a parameter must be  CR\n\0, that is a
        carrage return  line feed  null.
        Next the input file pointer must be updated so that it skips
        over the parameter that was just changed.  Since all
        parameters end with a null, this can be done simply by the
        following command:
        if (*in != EOF)
                in += lenstr(in) + 1.;
        I check for EOF so that if there was no input file, the
        end of file marker is not skipped.
        The +1 is so that the pointer skips over the terminating
        null in the input file.
        Finally, since this parameter has been changed the index
        to change[] must be updated.  This can be done with a
        j++;
        }
else    {
        if the user doesn't want to change this particular parameter,
        then the information should be copied over from the input
        file to the output file and both pointers, in and out,
        should be incremented.
        }
```

To summarize, your module should look like this:

```
if(change[j] == k++)
        {
         write new parameter to output file
         make sure last two characters of out are a carrage return
           followed by a null
        if (*in != EOF)
                in += lenstr(in) + 1;
        j++;
        }
else
        {
        pc = cpystr (out, in, NULL)+1;
         pc now points at the character after the null
```

```
            in += pc - out;
            out = pc;
            }

    next module
```

Remember if you add a parameter that you want the computer to know about if it looks at the file again, you should add this parameter to the binary header file after the last previous binary parameter. You must then change some other part of the program down the line so the computer reads in these values.

Now for a comment on how change[] is set up in header(). header() takes the input file and prints it out with the addition that before each parameter it prints a number. It knows where each parameter begins because a parameter is terminated with a null so after each null another parameter begins. The user then types the numbers of the parameters he wants to change. qanda just counts through the parameters until it comes to one the user wanted to change. If there is no input file, header will fill change[] with sequential integers. And fill in the first two characters of in as \Ø EOF. Header must also put an EOF in the last available byte of out and make sure there aren't any EOF's before it.
*/

```
#include    "sy:prog.req"
#include    <sy:std.h>
#include    "sy:sam.h"

qanda (ain, aout, change, pschan)
char        *ain;           /* a pointer to the bottom of the input file */
char        *aout;          /* a pointer to the bottom of the output file */
int change[];       /* the integer array which contains the changes
                       to be made to the input file */
int *pschan;        /* this is the channel that the file SETUP.SCN is
                       written on.  */
{
```

```
extern  int     chan;   /* this is the channel number of the new
                           data file that will be created   */
extern  struct  head1   h1;
                        /* head1 is defined in the file sam.h.  It
                           is a structure which contains the var-
                           iables which are written in the first
                           block of each scan*/
extern  int     clock;  /* a flag, = 1 if have real time clock in
                           the system, = 0 if do not*/
int     gpint[20];      /* buffer for comment section of header*/
int     e,f,i;  /* general purpose integers */
double  dtemp;  /* general purpose double */
char    c;      /* you guessed it, general purpose character */
int     *pi;    /* general purpose integer pointer */
char    *pc;    /* general purpose character pointer */
float   *pf;    /* general purpose floating point pointer */

char    *out;           /* out is the pointer to the next available
                           location in the output file  */
char    *in;            /* in is the pointer to the next parameter in
                           the old file that the program will act
                           upon  */
char    fname[16];      /* This will contain the name of the header
                           file */
int     k;              /* As you saw earlier the integers in change[]
                           contain the changes to be made.  k is the
                           identifying number of each module which
                           change[j] is compared against.  */
int     j;              /* j is the index for the array change[]  */
int     namflg;         /* if namflg is set to 1, the name should
                           be changed.  If not, then the name should
                           not be changed  */

out = aout + 512 + 17;  /* aout will not be changed in this program.
                           It will always point to the beginning of
                           output file.  out is incremented 512 bytes
                           so that it points to the beginning of
                           the ASCII file, and then an additional
```

```
                        17 bytes to skip room for the file name,
                        which will be filled in at the end of
                        quanda()  */
in = ain + 512 + 17;    /* same deal with in.  ain will not be changed
                        in is incremented so that it points to
                        the parameter past the first one, which
                        is always the file name.  */

k = 1;          /* start with first parameter since at top of file */
j = 0;          /* initilize index */


if (change[j] == k++)
        {
        namflg = 1;
        j++;
        }
else
        namflg = 0;
/* the above module tests whether the user wants to change the file
   name. If he does, the flag namflg is set so that later on the
   module that actually changes the file name will do so.  Look at
   the last module for more information on how the file name is
   handled  */
/***********   NUMBER OF CHANNELS ****************/
if ( change[j] == k++)  /* this module gets the number of channels*/
        {
        out = cpystr (out, pc = "Number of Channels: ", NULL);
        /* cpystr is a Whitesmiths subroutine which will copy strings
           from one place to another.  In this case the descriptor
           "Number of Channels" is being written in the output file*/
        for (;;)
                {
                /* this is an infinite loop which will can be exited
                   only by giving a reasonable number of channels  */
                h1.nchan = -1;  /* if something goes wrong and
                                   h1.nchan isn't set, h1.nchan =-1 so
                                   can't get out */
```

```
                putfmt ("\n%p?\n",pc);  /* ask Number of channels? */
                getfmt ( "%i", &(h1.nchan));
                if (h1.nchan >= 1  &&  h1.nchan <= 16)
                        break;
                putfmt ( "Please enter an integer from 1 to 16.\n");
                }
        out += itob( out, h1.nchan, 10);
        /* itob changes an integer characters and returns the number
           of characters  created. */
        *out++ = CR;
        *out++ = '\n';
        *out++ = '\0';  /* always end a parameter with a CR LF
                           followed by a null  CR = carriage return
                           and LF = line feed = \n*/
        if (*in != EOF)
                in += lenstr (in) + 1;
                /* if there is an input file, go to the next
                   parameter.  If there isn't, then in will be
                   pointing to the end of the file, so don't
                   increment it  */
        j++;    /* go on to next change */
        }
    else
        {
        pc = cpystr (out, in, NULL)+1;
        /* pc now points at the character after the null */
        in += pc - out; /* pc - out is the number of char. copied */
        out = pc;
        }

    /*************** CHANNEL ARRANGEMENT **************/
    if (change[j] == k++)
        {
        out = cpystr (out, "Channel Arrangement:\tDEVICE\t\tCHANNEL#\
\016\n", "\t\t\tFP:\t\t", NULL);
        for (;;)
                {
                h1.fpchan = -1;
```

```
                putfmt ( "\nIs the Fabry-Perot on channel 0, 1 or 2\
?\n");
                getfmt ( "%i", &(h1.fpchan));
                if (h1.fpchan >= 0  &&  h1.fpchan <= 2)
                        break;
                putfmt ("Please enter an integer from 0 to 2.\n");
                }
        out += itob (out, h1.fpchan, 10);
        *out++ = CR;
        *out++ = '\n';
        out = cpystr (out, "\t\t\tIodine:\t\t", NULL);
        for(;;)
                {
                h1.i2chan = -1;
                putfmt ("\nIs the iodine cell on channel 0, 1 or \
2?\n");
                getfmt ( "%i", &(h1.i2chan));
                if ( h1.i2chan >= 0  &&  h1.i2chan <= 2  && h1.i2chan\
 != h1.fpchan)
                        break;
                putfmt ( "Please enter an integer from 0 to 2 \
not equal to the fp channel\n");
                }
        out += itob (out, h1.i2chan, 10);
        *out++ = CR;
        *out++ = '\n';
        out = cpystr (out, "\t\t\tMain Data:\t", NULL);
        for (i=0; i == h1.fpchan || i == h1.i2chan;  i++)
                ;
        h1.mdchan = i;
        putfmt ("\nThe main data must be on channel %i.\n", i);
        out += itob(out, h1.mdchan, 10);
        *out++ = CR;
        *out++ = '\n';
        if (h1.nchan > 3)
                {
                h1.adchan = 3;
                putfmt("\nThe auxiliary data must be on sequential \
```

```
channels starting with channel 3.\n");
                }
            for (i = 3; i < h1.nchan; i++)
                {
                out = cpystr( out, "Channel ",NULL);
                out += itob (out, i, 10);
                *out++ = ':';
                *out++ = ' ';
                putfmt("What is on channel %i?\n",i);
                out += getlcr(out,70);
                /* getlcr is a SAMLIB routine which takes in
                   characters until it reaches a line feed or
                   until it takes in the number of characters
                   specified in it's second argument.  getlcr()
                   does the same thing as Whitesmith's getlin()
                   except getlin() returns the line feed only,
                   while getlcr() returns the carriage return
                   line feed at the end of the input line */
                }
            *out++ = '\0';
            if (*in != EOF)
                in += lenstr (in) + 1;
            j++;
            }
    else
            {
            pc = cpystr (out, in, NULL)+1;
            /* pc now points at the character after the null */
            in += pc - out;
            out = pc;
            }

    /************** SCAN TIME ******************/
    if (change[j] == k++)
            {
            out = cpystr (out, pc = "Scan Time (minutes:seconds)",": ",\
                        NULL);
            for (;;)
```

```
                {
                for(;;)
                        {
                        putfmt ( "\n%p?\n", pc);/* ask question */
                        if (clock == 0)
                                {
                                putfmt ("Since you do not have a real\
 time clock, the line time clock must be used.\n");
                                putfmt ("Therefore the scan time will\
 be rounded to the nearest multiple of 68.3 \nseconds.\n");
                                }
                        f = getlin (out, 10) - 1;/*f index of \n*/
                        for ( i = 0; i < f-1; i++)
                                {
                                c = *(out+i);
                                if ( !isdigit(c) && c != ':')
                                        break;
                                }
                        /* the above for loop checks to make sure
                           the user response is made up of valid
                           characters, ie digits or a semi-colin.
                           If an invalid character is encountered,
                           the for loop is exited before i=f-1.
                           The program will now test for this  */
                        if (i == f-1)
                                break;  /* if i=f-1 characters ok */
                        putfmt ("Please enter the scan time in the \
form integer:integer or just integer.\n");
                        }
                h1.stime = 0.0;
                e = btod(out, f, &dtemp);
                /* btod is a Whitesmiths subroutine that converts
                   a character buffer at out into a double which
                   it stores at dtemp. It converts characters until
                   f have been converted or until it reaches a non
                   digit, and returns the number of characters
                   converted.  Therefore if the scan time is in
                   just seconds, all of it will be converted.  If
```

```
                    it is in minutes:seconds, only the minutes will be
                    converted.  The number of characters to be
                    converted is f-1 because getlin also takes in
                    a newline */
                if (e != f) /* if e!=f then have min:sec */
                        {
                        dtemp *= 60.0;  /* mul min by 60 to get sec */
                        h1.stime = dtemp;
                        dtemp = 0.0;
                        btod (out+e+1, f-e-1, &dtemp);
                        }
                h1.stime += dtemp;
                if ( h1.stime >= 20.0  &&  h1.stime <= 18000.0)
                        break;
                        /* minimum scan time is 20 seconds, max is
                           1 hour.  Arbitray maximum  */
                putfmt ("The scan time must be at least 20 seconds\
and no more than 1 hours.\n");
                }
        if (clock == 0)
                {       /* clock == 0, must use l.t.c.  Make sure
                           o.k. for 60 Hz clock */
                i = (h1.stime/h1.npoints) * 60.0 + 0.5;
                /* i now equals the number of 60 Hz ticks/point */
                i = i * (h1.npoints/60.0) + 1.0;
                if (i < 69)
                        i = 69;
                h1.stime = i;
                }
        /* now convert the scan time back to a buffer and write into
           out */
        i = h1.stime/60.0;      /* i now equals the # of minutes */
        if (i != 0)
                {
                out += itob(out, i, 10);/* write minutes into out*/
                *out++ = ':';
                }
        e = h1.stime;           /* have to use % operator with ints*/
```

```
                i = e % 60;             /* i now equals the number of secs */
                if ((i/10) == 0)
                        *out++ = '0';   /* if secs a single digit, this will
                                           put a 0 in front of digit */
                if (i == 0)
                        *out++ = '0';
                else
                        out += itob(out, i, 10);
                *out++ = CR;            /* have to make sure have CR LF at*/
                *out++ = '\n';          /* end of parameter */
                *out++ = '\0';
                if (*in != EOF)
                        in += lenstr (in) + 1;
                j++;
                }
        else
                {
                pc = cpystr (out, in, NULL)+1;
                /* pc now points at the character after the null */
                in += pc - out;
                out = pc;
                }


        /*************** MONOCHROMATER *****************/
        if (change[j] == k++)
                {
                for(;;)
                        {
                        out = cpystr(out, "Monochromater(angstroms): ", NULL);
                        putfmt ("\nEnter the approximate monochromater \
reading in angstroms.\nFor example 5890.5.\n");
                        f = getlin(out, 10);
                        btod( out, f-1, &dtemp);
                        h1.monoch = dtemp;
                        if(h1.monoch > 0.0  && h1.monoch < 15000.0)
                                break;
                        putfmt("Please enter a number between 0 and 15000.\n"\
```

```
);
                }
            f = dtof (out, dtemp, 5, 2);
            /* The monochromater reading is taken in as characters because
               getfmt can't take in floating numbers.  It is converted
               to a floating number with btod.  It is then converted
               back and written in the output file.  I chose to do this
               extra conversion to make sure the person sees the same
               number as the computer.  */
            out += f;
            *out++ = CR;
            *out++ = '\n';
            *out++ = '\0';
            if (*in != EOF)
                    in += lenstr (in) + 1;
            j++;
            }
    else
            {
            pc = cpystr (out, in, NULL)+1;
            /* pc now points at the character after the null */
            in += pc - out;
            out = pc;
            }

    /************* FREE SPECTRAL RANGE **********************/
    if (change[j] == k++)
            {
            out = cpystr (out, "Free Spectral Range (MHz): ", NULL);
            putfmt ("\nWhat is the free spectral range of the Fabry-\
Perot in MHz?\n");
            for(;;)
                    {
                    f = getlin(out, 10);
                    btod( out, f-1, &dtemp);
                    h1.fpfsr = dtemp;
                    if(h1.fpfsr > 50.0 && h1.fpfsr < 300000.0)
                            break;
```

```
                putfmt("Please enter a number between 50 and 300000.\
\n");
                }
            f = dtof (out, dtemp, 6, 3);
            /* The free spectral range is taken in as characters because
               getfmt can't take in floating numbers.  It is converted
               to a floating number with btod.  It is then converted
               back and written in the output file.  I chose to do this
               extra conversion to make sure the person sees the same
               number as the computer.  */
            out += f;
            *out++ = CR;
            *out++ = '\n';
            *out++ = '\0';
            if (*in != EOF)
                in += lenstr (in) + 1;
            j++;
            }
    else
            {
            pc = cpystr (out, in, NULL)+1;
            /* pc now points at the character after the null */
            in += pc - out;
            out = pc;
            }

    /************** MOLECULE ***********************/
    if (change[j] == k++)
            {
            out = cpystr (out, "Molecule: ", NULL);
            putfmt ("\nWhat molecule are you working on?\n");
            out += getlcr (out, 80);
            *out++ = '\0';
            if (*in != EOF)
                in += lenstr (in) + 1;
            j++;
            }.
    else
```

```
                {
                pc = cpystr (out, in, NULL)+1;
                /* pc now points at the character after the null */
                in += pc - out;
                out = pc;
                }

        /*************** MAIN DATA SCALE **********/
        if (change[j] == k++)
                {
                out = cpystr (out, "Main Data Scale: ",NULL);
                putfmt ("\nEnter the scale setting of the instrument \
measuring the main data channel.\n");
                f = getlin(out, 10);
                btod( out, f-1, &dtemp);
                h1.dscale = dtemp;
                f = dtoe (out, dtemp, 1, 3);
                out += f;
                *out++ = CR;
                *out++ = '\n';
                *out++ = '\0';
                if (*in != EOF)
                        in += lenstr (in) + 1;
                j++;
                }
        else
                {
                pc = cpystr (out, in, NULL)+1;
                /* pc now points at the character after the null */
                in += pc - out;
                out = pc;
                }

        /*************** MAIN DATA PM TUBE ***************/
        if (change[j] == k++)
                {
                out = cpystr (out, "Main Data PM Tube Voltage: ",NULL);
                putfmt ("\nWhat is the PM tube voltage for the main data \
```

```
channel?\n");
            out += getlcr (out, 80);
            *out++ = '\0';
            if (*in != EOF)
                    in += lenstr (in) + 1;
            j++;
            }
    else
            {
            pc = cpystr (out, in, NULL)+1;
            /* pc now points at the character after the null */
            in += pc - out;
            out = pc;
            }

    /************* COMMENTS ***************/
    /* IMPORTANT this must be the second to the last module, just before
       the file name  */
    if (change[j] == k++)
            {
            out = cpystr (out, "COMMENTS:", NULL);
            *out++ = CR;
            *out++ = '\n';
            while (*in != '\n' && *in != EOF && *in != '\0')
                    in++;
            /* in should now point at \n at end of COMMENTS: line */
            if (*in != EOF)
                    in++;
            /* in now points to the first comment */
            for (i = 1, pc = in; *pc!=EOF && *pc!='\0'; pc++)
                    {
                    /* first display the comments with numbers for each
                       line */
                    if (*(pc-1) == '\n')    /*new line so put out number*/
                            {
                            putfmt ("(%i) ", i);
                            i++;
                            }
```

```
                putch(*pc);
                }
        e = 0;
        if (pc != in)   /* if() checks to see if were old comments*/
                {
                /* now find out what the user wants to change */
                putfmt ("Enter the numbers of the lines you want to \
delete, seperated by spaces\n");
                e = getint (gpint, 20);
                bubble (gpint, e);      /*put the changes into order*/
                }
        gpint[e] = -1;  /* this makes it easy to
                           find end of list */
        for (i=1, f=0; *in!='\0' && *in!=EOF && *(out+3)!=EOF; in++)
                {
                /* i keeps track of line number that in presently
                   points to and f is index for gpint, the lines
                   to be deleted */
                if ( i != gpint[f])
                        *out++ = *in;   /* copy character */
                if ( *in == '\n')
                        {
                        if (i == gpint[f])
                                f++;
                        i++;            /* on new line */
                        }
                }
        putfmt ("Type any new comments, when you are through type \
the escape key and hit return.\n");
        clearl();
        while ((c=ttyin()) != ESC && *(out+3) != EOF)
                *out++ = c;
        if (*(out+3) == EOF)
                putfmt ("Output buffer too small to hold all \
old comments.\n");
        }
    else    {
        while (*(out+3) != EOF && *in != EOF)
```

```
                *out++ = *in++;
            if (*(out+3) == EOF)
                putfmt ("Output buffer too small to hold all \
old comments.\n");
            if (*(out-1) == '\0')
                out--;
            }

    /* time to close the output buffer */
    if (*(out-1) != '\n')    /* if the user didn't close with a CR LF */
            {                /* put one in */
            *out++ = CR;
            *out++ = '\n';
            }
    *out++ = '\0';
    *out++ = EOF;            /* the end of file marker */
    /* Now is my chance to stick in any parameters that I don't want the
       user to have to type in, and still have these appear in the ascii
       header file if the file is ever typed out.  Note that header() and
       qanda() never look beyond the EOF marker.  Therefore I can stash
       anything I want to after the EOF and they'll never see it.  However
       when the file is typed out, the EOF is ignored. */
    /*************** DATE ****************/
    out = cpystr (out, "Date: ", NULL);
    h1.date = DATEM;         /* get the system date */
    datee (h1.date, &e, &f, &i);     /* expand the date out to day, month,
                                        year */
    out += itob (out, f, 10);        /*copy the month over */
    *out++ = '/';
    out += itob (out, e, 10);        /* now get the day */
    *out++ = '/';
    i -= 1900;                       /* strip off the 1900 from the year*/
    out += itob (out, i, 10);        /* finally the year */
    *out++ = CR;
    *out++ = '\n';
    *out++ = '\0';
```

```
    h1.datblk = (out - aout)/512 + 1;/* h1.datblk will be the first block
                                        of data on the data file.  It is
                                        also the length of the header
                                        file in blocks */

    /* now fill in all the other members of h1 */
    h1.id = 052525;
    h1.stype = 1;           /* scan type is one */
    h1.version = 1;         /* version type */
    h1.dtype = 1;           /* the data will be integers */
    h1.npoints = 4096.0;    /* the d/a has 4096 steps so will take that
                               many points */
    /* finally initilize the variables in h1 that the user doesn't know
       yet */
    h1.beginbin = 0.0;
    h1.endbin = h1.npoints;
    h1.coflag = -1;
    for (i = 0; i < 6; i++)
            h1.coeff[i] = 0.0;
    h1.fpcoeff = 0.0;
    h1.fptype = 0;
    h1.i2peak = 0.0;
    h1.i2num = 0;
    h1.i2wn = 0.0;

    /* Now put the binary data into the output file */

    cpybuf (aout, &h1, sizeof(h1));

    clearl();       /* make sure input cleared */
    /******************** FILE NAME ***********************/
    if (namflg == 1)
            {
            putfmt ("\n%p%p", "Please enter the scan name.  For example: \
RK0:SCAN01.DAT or just \n", "SCAN01.DAT.  The default device is DK.\n");
            f = getlin (aout+512,15) - 1;   /* the -1 is to skip \n */
            for (i = f+512; i < 528; i++)
                    *(aout + i) = ' ';
```

```
                /* fill out the name with blanks */
            *(aout + i) = '\0';
            }
    else
            cpystr (aout + 512, ain + 512, NULL);
            *(aout + 512 + 14) = ' ';
            *(aout + 512 + 15) = ' ';
            /* the above assignment removes the CR\n from the old file
               name.  If you didn't do this, an error would be
               returned when you tried to convert the f.n. to
               rad50 with devblk */
    /******** OPEN DATA FILE ********/
    /* you will notice that a channel is always opened.  Therefore
       the calling program must be sure to purge chan and schan if
       it calls qanda() more than once.  Otherwise all the channels
       would soon be used up */
    while((chan = enterf (aout+512, (h1.datblk + (h1.nchan*16) + 1))) < 0)
            {
            /* the length of the file in blocks is h1.datblk, the length
               of the header file, plus the number of channels of data
               times 16 since each channel will have 4096 points taken
               = 4096 words = 16 blocks.  If enterf returns a negative
               number there is a problem in opening a channel */
            if (chan == -1)
                    putfmt ("Problem with file name, please renter.\n");
            else if (chan == -2)
                    putfmt ("No channels available exit& fix problem.\n");
            else if (chan == -3)
                    putfmt ("Please enter a new file name. \n");
            else if (chan == -4)
                    putfmt ("%p%p%p",\
"\nThere is not enough contiguous free space on the disk or the directory\n",\
"is full.  Reenter the file name with a device that has sufficient\n",\
"room.\n");

            /* now get the new file name */
            f = getlin (aout+512,15) - 1;   /* the -1 is to skip \n */
            for (i = f+512; i < 528; i++)
```

```
                *(aout + i) = ' ';
                /* fill out the name with blanks */
            *(aout + i) = '\0';
            }

    *(aout + 512 + 14) = CR;
    *(aout + 512 + 15) = '\n';
    *(aout + 512 + 16) = '\0';
    /* put a carriage return after the file name and before the
       \0 */

    /********** OPEN SETUP.SCN CHANNEL (schan) ***********/
    cpystr (fname, pc = "DK :SETUP.SCN", NULL);
    remove (pc);                    /* delete the old file */
    while ((*pschan = enterf (fname,h1.datblk)) < 0)
            {
            if (*pschan == -1)
                    putfmt ("Bad device name.  Please reenter\n");
            else if (*pschan == -2)
                    putfmt ("No channels.Exit program & fix problem.\n");
            else if (*pschan == -3)
                    putfmt ("You must delete the old file or switch to \
a different device.\n");
            else if (*pschan == -4)
                    putfmt ("There is insufficient room on the DK \
 device.\n");
            putfmt ("%p%p%p%p","Please enter a device name for a disk \
with sufficient room.\n","For example type \n", "DK return \n",\
 "or DY1 or RK0.\n");
            f = getlin (fname, 3);
            if (!isalpha(fname[2])) /* if the third character isn't a */
                    fname[2] = ' '; /* letter, replace it with a space */
            }
    return(out - aout);     /* returns size in bytes */
}
```

```
/*****************     takdat
    A Stephen McDonald production
    This is part of the software that scans and takes data from
cw laser experiments.  By the time this program, takdat(), is called
the user has chosen the desired scan parameters.  As its name implies
takdat() and the programs it calls take the data in and scan the
laser.  Also preliminary work is done on the data, a trivial example
being the data is plotted on the VT105.
    All takdat() does is set up the various buffers for taking the
data in, initilize the clock and D to A convertor and then calls digest()
which actually "digests" the data.  The hard work is done by the interrupt
routine, ADIR, which must be in core for this part of the program.

*/

#include    "sy:adress.lib" /* has the address of the d/a */
#include    "sy:sam.h"
#define     B1SIZE  1024    /* the size of the interrupt routine's buffer in
                           bytes */
#define     B2SIZE  1024    /* the size of the intermediate buffers in bytes
                           this must be a multiple of 512, the size of one
                           block on the disk  */

takdat()
{
    extern  struct  head1   h1;
                            /* the structure which contains the infor-
                               mation written in the first block of
                               each scan.*/
    extern  int     *b1;    /* the buffer which the interrupt will
                               place the data in  */
    extern  int     *b1max; /* a pointer to the space after the highest
                               location of buffer b1 */
    extern  int     *b1mp;  /* a movable pointer which the main program
                               (as opposed to the interrupt routine) uses
                               to transfer data from b1 to b2 or b3. */
    extern  int     *b1ir;  /* a movable pointer which the interrupt
                               routine uses to fill the buffer b1 with
```

```
                                data from the a/d's. */
extern  int     *b2;    /* one of the two buffers which are an
                           intermediate between b1 and the disk */
extern  int     *b2max; /* a pointer to the word location after the
                           highest word of the buffer b2 */
extern  int     *b3;    /* the second of the two intermediate
                           buffers */
extern  int     *b3max; /* a pointer to the word location after the
                           highest word of the buffer b3 */
extern  int     *bin;   /* this is either b2 or b3, whichever happens
                           to be taking the data from b1.  The other
                           buffer will be in the process of having
                           its data read to the disk  */
extern  int     *binmax;/* a pointer to the location following the
                           last word in whichever of b2 or b3 is
                           the input buffer */
extern  int     *binp;  /* a pointer to the spot in the input buffer
                           that the next data obtained from b1 will
                           be placed  */
extern  int     *bout;  /* this is either b2 or b3, whichever buffer
                           is having its data read to the disk */
extern  int     *boutmax;/* the maximum of the output buffer */
extern  int     clock;  /* a flag set by hardse().  If clock = 0,
                           there is not a real time clock on the
                           system and the line time clock must be
                           used. */
extern  int     scanon; /*a flag that tells the interrupt routine
                           whether to take data.  If scanon = 0,
                           the interrupt routine will not take
                           any additional data.  If scanon = 1,
                           the int routine will take data */

extern  int     dabegin;/* the number that will set the d/a to
                           its lowest setting*/
int     *pi;            /* general purpose integer pointer*/
int     oflow;          /* this is needed if the line time clock must
                           used.  It is the two's complement of the
                           number of 60 Hz ticks that
```

```
                        should go by before a point is taken */
int     i;              /* general purpose integer */
static  int     firbuf[B1SIZE/2];
static  int     secbuf[B2SIZE/2];
static  int     thrbuf[B2SIZE/2];


/* now it's time to initilize all of the above pointers */
b1 = firbuf;
b1max = b1 + B1SIZE/2;          /* b1 is a circular buffer, so have
                                   to know top and bottom.  b1 will
                                   always point to bottom and b1max
                                   will always point to top  Divide
                                   B1SIZE by 2 because it's in bytes
                                   and b1 and b1max are integer
                                   pointers*/
b1mp = b1ir = b1;               /* b1mp and b1ir are the pointer used
                                   to remove data and place data in b1
                                   They are both initilized to the
                                   bottom of b1 */
b2 = secbuf;
                                /* b2 will be the input buffer first.
                                   b2 always points to the bottom of
                                   buffer b2.  binp is the pointer
                                   used to read data into bin.  */
bin = binp = b2;
b2max = binmax = b2 + B2SIZE/2;
b3 = thrbuf;
bout = b3;
b3max = boutmax = b3 + B2SIZE/2;
scanon = 0;                     /* just make sure won't scan */
if (clock == 0)                 /* must use l.t.c so set oflow */
        {
        /* if clock == 0 then don't have a real time clock board
        i.e. a KWV11-A in the system.  Therefore have to use the
        60 Hz. line time clock */
        oflow = -(60.0)/(h1.npoints/h1.stime);
        /* this is just saying ticks/point = (ticks/sec)/(points/sec)
```

```
        */
        if (abs(oflow) < 1)
                oflow = -1;     /* just to be safe */
        i = LTIV;               /* the address of the interrupt vector
                                   for the ltc */
        }
else
        {                       /* have a real time clock so set up*/
        i = CLKIV;              /* the address of the int vector */
        pi = CLKCSR;
        if (h1.stime < 1200.0)  /* if less than 20 min */
                {
                *pi++ = 0122;   /* set to tick at 100 kHz, interrupt
                                   enabled, repeated interval */
                *pi = -100000.0/(h1.npoints/h1.stime);
                                /* above sets up clock buffer */

                }
        else
                {
                *pi++ = 0142;   /* tick at 1 kHz for slow scans */
                *pi = -1000.0/(h1.npoints/h1.stime);
                                /* above sets up clock buffer */
                }
        }
intset (clock, oflow, i, ADCSR, DAC0, CLKCSR, h1.nchan);
/* this sets up the interrupt routines to start taking data.  Note
   that if the line time clock is being used, the interrupts start
   immediately because can't turn interrupts off.  No data is taken
   because scanon = 0 */

digest ();
if (clock == 1)
        {
        pi = CLKCSR;
        *pi = 0;        /* turn rtc off if you have one */
        }
intclr (i);
```

```
    return;                 /* call digest() and return when it returns */
}


ADIR.MAC
.title      adir


CLOCKIV:    .word   0       ;the interrupt vector for either the line
                            ;time clock (ltc) or the real time clock
                            ;(rtc) depending on which clock you're
                            ;using
ADCSR:              .word   0       ;the location of the a/d control status regis.
ADBUF:              .word   0       ;the location of the a/d buffer
DAC0:               .word   0       ;the location of the buffer for d/a #0
CLKCSR:             .word   0       ;the location of the csr of the rtc
NCHAN:              .word   0       ;the number of channels of data being taken
OLDIV:              .word   0       ;the old interrupt vector for the clock
OLDPS:              .word   0       ;the old ps for the clock
OFLOW:              .word   0       ;the ltc ticks at 60Hz.  To take data at a
                            ;lower rate than 60Hz an interrupt routine
                            ;the 2's complement of the number of 60Hz
                            ;ticks per point is loaded into the counter
                            ;LTCOUNTER, which is then incremented until
                            ;it reaches zero.  LTCOUNTER is then
                            ;reinitialized and a data point is taken
LTCOUNTER:  .word   0

.globl      scanon                  ;a flag telling whether data shold be taken
                            ;or not.  if = 0, don't take data

;   intset sets up the interrupt vectors for introu.
;The arguments to intset are in the following format:
;16(sp) contains the number of channels of data to be taken
;14(sp) contains the address of the r.t.c csr
;12(sp) contains the address of d/a #0
;10(sp) contains the address of the a/d csr
;6(sp)  contains the address interrupt vector of the clock being used
```

```
;4(sp)  contains the 2's complement of the number of times the l.t.c will
;   tick before data is taken
;2(sp)  contains the variable clock, clock = 0 means use ltc, =1 use r.t.c.
; (sp)  the return address

INTSET::
    mov     6(sp),  CLOCKIV ;get the clock interrupt vector
    mov     10(sp), ADCSR   ;get the a/d control/status register address
    mov     #2,     ADBUF
    add     ADCSR,  ADBUF   ;a/d buffer = a/d csr + 2
    mov     12(sp), DAC0    ;get the d/a address
    mov     16(sp), NCHAN   ;get the number of channels
    mov     6(sp),  r0
    mov     (r0)+,  OLDIV   ;save the old interrupt vector
    mov     (r0),   OLDPS   ;save the old processor status word
    mov     #200,   (r0)    ;install new processor status word
    tst     2(sp)           ;see if l.t.c or r.t.c
    bne     RTC             ;if clock # 0, go to RTC (real time clock)
    mov     4(sp),  OFLOW   ;if using ltc, get the over-flow word
    mov     4(sp),  LTCOUNTER;initilize the counter
    mov     #LTIME, -(r0)   ;get the line time routine to be the int. rou.
    br      RETURN          ;go back
RTC:        mov     #RTIME, -(r0)   ;have a real time clock.  Put rtc rou. into iv
    mov     14(sp), CLKCSR  ;save the clock csr
RETURN:     rts     pc


;   Since one of two clocks can be used, the real time clock or the
;line time clock, there are a few small differences between the two
;interupt routines needed to handle them.  These differences are taken care
;of at the very beginning of the interrupt after which they both jump to a
;standard data processing routine, INTROU

LTIME:      ;this is the ltc routine
    tst     scanon
    beq     INTOVER ;if scanon = 0, don't take any data
    inc     LTCOUNTER       ;increment the counter
    bne     INTOVER ;if have not counted up to zero, don't take
```

```
                        ;data
    mov     OFLOW,LTCOUNTER ;time to take data so initialize counter
    br      INTROU          ;and go get the data

RTIME:      ; the real time clock routine
    bic     #200,   @CLKCSR ;clear the overflow flag in the clock
    tst     scanon          ;see if should take data
    beq     INTOVER ;if scan over, return without taking data
    br      INTROU          ;go take the data
INTOVER:rti

;   This routine is an interrupt routine designed to take data
; from the analogue to digital converters and place it in a small buffer
; (b1).  It then steps the d/a converter to scan the laser.
;   The clever part of this routine is that b1 is a circular buffer.
; This means when a pointer reaches the top of the buffer, it is reset
; to the bottom.  I choose this so that it would be easy to read and write
; essentially at the same time, i.e. it makes first in first out operation
; possible. Here's the setup; this interrupt routine fills up b1 using the
; pointer b1ir.  The main program reads the data with the pointer b1mp.
; The main program will read the data until b1mp=b1ir, at which point
; it will have caught up with the data that has come in.  b1ir points at
; the next available space in the buffer, i.e. the space into which
; data will be placed at the next interrupt.  The interrupt
; routine reads in data unless b1ir=b1mp, which means if it continues
; to write it will write over data the main program has not yet removed.
; If this happens you must either slow down the data input rate, increase
; the size of b1, or make the routines in your main program faster so that
; it removes the data more quickly from b1.


.globl      adoffset        ;the offset for the a/d.  =2048 for the ADAC in
                    ;bipolar mode, =0 for the ADAC in unipolar or for
                    ;the Digital a/d in any mode
.globl      b1ir            ;buffer 1 interrupt routine pointer, b1ir points to
                    ;the next location in b1 to be filled from the a/d
                    ;by this interrput routine.
```

```
.globl      b1mp            ;buffer 1 main program pointer, b1mp points to the
                    ;next location to be read by the main program.
.globl      datflg          ;data flag.  This routine sets the datflg in the
                    ;following way; datflg=1 if data is ready, datflg=-1
                    ;if the incoming data has overrun the data already
                    ;collected before it can be read by the main program.
                    ;The main program sets datflg=0 when it reads data.
.globl      b1              ;address of beginning of buffer 1.  This is a constant
                    ;as soon as the program is compiled.
.globl      b1max           ;address of maximum of b1.  This is a constant as
                    ;soon as the program has been compiled
.globl      gain            ;An array which contains the gain for the channels.
                    ;If the variable "gain" = 030, 020, 010 or 0 this
                    ;corresponds to an actual gain of 1, 2, 4 or 8
                    ;The variable gain must also have the channel number
                    ;in the upper byte and a one in the lowest byte.
                    ;When arranged in this seemingly arbitrary manner,
                    ;the variable gain can be loaded directly into the
                    ;A/D csr.
.globl      dabuf           ;the number that is loaded into the d/a buffer.  this
                    ;is initilized in the routine scncom.c


;   And now, what you've all been waiting for THE INTERRUPT ROUTINE!!!
INTROU:
    mov     r0,     -(sp)   ;push r0 and r1 onto stack so they're free
    mov     r1,     -(sp)
    mov     r2,     -(sp)   ;r2 too
    mov     #gain,  r2      ;r2 will point to the gain for each channel
    mov     @ADBUF, r0      ;this move is just to clear the a/d ready
                            ;flag.  Don't care about what's in ADBUF yet.
    mov     (r2)+,  @ADCSR  ;set the gain ,set the multiplexer to 0
                            ;and start the conversion
    mov     b1ir,   r0      ;r0 will be the pointer while taking in data
    mov     NCHAN,  r1      ;r1 will be the channel counter
DATIN:      tstb    @ADCSR          ;test to see if conversion finished
    bpl     DATIN           ;if not, go back and wait until it is
```

```
    mov     @ADBUF, (r0)    ;take the data from the a/d and put it in b1
    add     adoffset,(r0)   ;add in the a/d offset
    bic     #170000,(r0)+   ;clear out upper bits
    mov     (r2)+,  @ADCSR  ;set ADCSR for next conversion and start it
    cmp     r0,     b1max   ;is r0 at the top of the buffer?
    bne     2$              ;if no, go to 2$
    mov     b1,     r0      ;if at top, reset r0 to the bottom
2$: cmp     r0,     b1mp    ;is r0 about to overrun unread data?
    beq     ERROR           ;if overrun, report error and leave
    dec     r1              ;otherwise count down the channels
    bne     DATIN           ;if haven't taken in all channels, go back

;so have all the data, time to return
    mov     r0,     b1ir    ;update b1ir, it now points at the next slot
                            ; to be filled.
    mov     (sp)+,  r2      ;restore old r2
    mov     (sp)+,  r1      ;restore old r1
    mov     (sp)+,  r0      ;restore old r0
    mov     #1,     datflg  ;set data flag
    inc     dabuf           ;increment the d/a counter
    mov     dabuf, @DAC0    ;step the d/a
3$: rti                     ;go home

; if here, the interrupt routine is taking in data faster than the
; main program is removing it.  The interrupt routine will now stop
; the clock and set the data flag to -1 to inform the main program
; about the problem.  If this error occurs you must 1) take data
; more slowly; 2) change the size of the buffer b1 or 3) change
; the main program so that it removes data more quickly
ERROR:      mov     #-1,    datflg  ;tell main program about screw up
    clr     scanon          ;stop the scan
    rti                     ;and quit


;INTCLR restores the old interrupt routine.
INTCLR::mov 2(sp),  r0      ;get the location of the interrupt vectors
    tst     (r0)+           ;increment r0
    mov     OLDPS,  (r0)    ;restore the old processor status
```

```
    mov     OLDIV,  -(r0)   ;restore the old interrupt vector
    rts     pc              ;and return

.end
```

DIGEST.C

```
/*          digest
    A Stephen McDonald creation
    This is part of the software that is supposed to scan the
cw lasers and take in data over the a/d's.
This program sets a pointer and then calls the programs that
deal with storing or crunching or displaying the data brought in by
the interrupt routine ADIR.  */


#include    "sy:graph.lib"
#include    "sy:adress.lib"
#include    "sy:prog.req"
#include    "sy:sam.h"
#include    <sy:std.h>
digest()
{
    extern  struct  head1   h1;
                    /* h1 contains the variables that are
                               written in the first block of each
                               scan */
    extern  int     *b1;    /* the bottom of the interrupt routine
                               input buffer */
    extern  int     *b1ir;  /* the pointer that the interrupt routine
                               uses while taking in data from the
                               a/d's.  It points to somewhere in
                               the buffer b1  */
    extern  int     *b1mp;  /* this is the pointer the main program
                               uses to empty the data from b1  */
    extern  int     *b1max; /* points to the word after the top of the
                               b1 buffer */
    extern  int     *bin;   /* the bottom of the input buffer that
```

```
                        buffil() uses when taking data from
                        the interrupt input buffer */
extern  int     *binmax;/* the pointer to the word past the last space
                        of the buffer pointed to by bin */
extern  int     *binp;  /* the pointer to bin that buffil() uses */
extern  int     blk;    /* this is the block on the disk that the
                        computer is writing data to */
extern  int     chan;   /* the channel number of the data file */
extern  int     clock;  /* if = 1, then using real time clock */
extern  int     datflg; /* this is a flag that is set by the interrupt
                        routine when it takes data and cleared by
                        digest() when it reads data */
extern  int     dabegin;/* the number necessary to set the d/a to
                        its lowest voltage for the beginning of the
                        scan */
extern  int     dabuf;  /* this is a counter which is steped and
                        loaded into the d/a by ADIR to scan the
                        laser */
extern  int     scanon; /* if scanon = 0, int. routines will not take
                        data.  If scanon = 1, will take data */
int     *tblir;         /* a temporary pointer.  It points to where
                        blir was at the time the main program
                        began to process a batch of data  */
float   nproc;          /*the number of points that have been
                        processed */
int     i, j;
int     *pi;    /* general purpose integer pointer */
int     c;


if (clock == 1)
        {
        pi = CLKCSR;
        *pi += 1;               /* start the clock */
        }
for (;;)        /* stay in this loop until the user has a scan he
                   wants to keep */
        {
```

-248-

```
blk = h1.datblk;          /* set blk to start writing data
                             at the first data block */
dabuf = dabegin;          /* set d/a counter such that at the
                             start of the scan voltage will
                             be correct */
pi = DAC0;
*pi = dabegin;            /* clear the d/a convertor */
putfmt("\nTo end the scan early, hit the return button.\n\n");
clearl();                 /* clear input line */
ready();                  /* tell user computer is ready */
clearl();                 /* get rid of any extra trash */
printm (CLRSCR);          /* clear the screen of all print */
printm (CUR_HOME);        /* send the cursor to the top left
                             corner of the screen */
printm (GRA_ON);          /* turn the waveform generator on */
printm (G01);             /* enable dual strip charts */
printm (GMARK);           /* enable graph markers */
printm (GLX);             /* set x to zero for start of scan */
gsend (0);
printm (GRA_OFF);
datflg = 0;               /* set the data flag to zero */
nproc = 0;                /* set the number of points processed
                             to zero */
scanon = 1;               /* let the scan begin */
while (nproc < h1.npoints)
        {
        /* nproc is the number of points the main program has
           processed, h1.npoints is the number of points that
           are supposed to be taken from each channel.
           Note that this program can be interrupted at
           any point without it losing data.  Any additions
           must have the same capability, or else set the
           processor status with the SAMLIB routine
           setps(). */
        if (ttinr() > 0)
                {
                clearl();
                scanon = 0;     /* stop the int routines.*/
```

```
        break;          /* if the user types a
                           return during the scan
                           he wants to terminate
                           the scan early */
        }
while (datflg == 0)     /* wait for data from
                           interrupt routine */
        ;
setps(07 << 5);         /* can't have interrupt
                           until after clear
                           datflg.  Otherwise
                           could miss a datflg =
                           -1 and hang up so set
                           PS such that no int
                           allowed. */
if (datflg == -1)       /* make sure not taking
                           data too fast */
        {
        setps (0);      /* set ps back down */
        break;          /* and get out, note that
                           the int. rou. has reset
                           scanon to zero */
        }
datflg = 0;             /* tell the int rou data is
                           cleared */
tblir = blir;
/* The crunching subroutines need a pointer to tell
   them how much data to remove from bl, the interrupt
   input buffer.  Since there are several subroutines,
   and since blir is changed by the interrupt routine,
   a temporary pointer is needed so that all the
   subroutines get the same data on each pass.
   */
setps(0);
/* at this point I'll check to see if all the data
   points have been taken.  Since bl is a circular
   buffer, I may have tblir > blir or tblir < blir.
   To determine the number of points waiting to be
```

```
   processed, I have to treat the two possibilities
   differently.  i will contain the number of
   points the interrupt routine has taken in but
   has not processed */
if (tb1ir >= b1mp)
        {
        i = tb1ir - b1mp;
        i /= h1.nchan;  /* i now equals the # of sets
                           of data points */
        if (i >= (h1.npoints - nproc))
                /*if the # taken >= number supposed to
                   be taken, then .... */
                {
                scanon = 0;     /*stop scanning*/
                i = h1.npoints - nproc;
                b1ir = tb1ir = b1mp + i*h1.nchan;
                /* set it so will process correct
                   number of points */
                }
        }
else    /*tb1ir < b1mp which means the int pointer has
           rolled over the top of b1 and has been
           reset to the bottom */
        {
        i = (b1max - b1mp) + (tb1ir - b1);
        i/= h1.nchan;
        if (i >= (h1.npoints - nproc))
                {
                scanon = 0;
                i = h1.npoints - nproc;
                b1ir = tb1ir = b1mp + i*h1.nchan;
                if (b1ir >= b1max)
                        b1ir = tb1ir = b1 + (tb1ir -\
b1max);
                /*if b1ir is still > b1max, roll it
                   around to the bottom */

                }
```

```
                }
            nproc = nproc + i;
            buffil (b1mp, tb1ir);
                                    /* pass the pointers to the
                                       subroutines and crunch
                                       away.    */
            display (b1mp, tb1ir);  /*graph the spectra. */
            b1mp = tb1ir;           /* update the main pointer
                                       to b1 to show have
                                       processed the data */

            i = h1.stime*(1.0 - (nproc/h1.npoints));
            /* i is now the number of seconds remaining in the
               in the scan*/
            prttim (i);     /* tell user seconds */
            }
        /* through scanning */

        /* finish writing data */
        if (WAITM (chan) < 0)   /* wait until chan clear */
            putfmt ("Trouble in paradise. Error returned on\
wait request\n");
        if (WRITEM (chan, blk, bin, (binp - bin)) < 0)
            putfmt ("Error returned on last write request.\n");
                                /*write the last buffer full of data*/
        if (WAITM (chan) < 0)           /* wait until all written */
            putfmt ("Error returned on last wait request.\n");

        pi = DAC0;
        *pi = dabegin;          /* reset the D/A convertor */

        /* if user stopped scan early (h1.npoints > nproc) have to set
           scan time and number of points to shortened values */
        if (h1.npoints > nproc)
            {
            h1.stime = h1.stime * (nproc/h1.npoints);
            h1.npoints = nproc;     /* record # of points taken */
            h1.endbin = h1.npoints;
```

```
                for (i = 0; i < 256; i++)
                        *(bin + i) = 0;
                cpybuf (bin, &h1, sizeof(h1));  /* copy new header
                                                   block */
                WRITEM (chan, 0, bin, 256);     /* and write to file*/
                WAITM (chan);
                }

        if (datflg == -1)       /* check for error */
                {
                printm (GCLEAR);
                putfmt ("%p%p%p%p%p",\
"   The interrupt routine is taking data in faster than it can\n",\
"be processed.  You have three choices:\n",\
"1) take data more slowly\n",\
"2) increase the size of the interrupt buffer b1\n",\
"3) make the data processing part of the program faster\n");
                PURGEM (chan);
                return;
                }

        clearl();               /* get rid of extra characters */
        putfmt ("%p%p",\
"   Do you want to redo the last scan and write over the old \n",\
"data? (Y or N)\n");
        c = ttyin();            /* take in a character */
        printm (GCLEAR);        /* clear the graph off the
                                   screen */
        if (c != 'Y'  &&  c != 'y')
                break;
        binp = bin;             /*have to reset the pointers*/
        blmp = blir = b1;       /*before taking the scan
                                   again */
    }
    CLOSEM (chan);  /* this is from the library file prog.req.  It
                       closes the channel that data has been
                       taken on, thereby making the data file
                       permanant  */
```

```
    WAITM (chan);
    return;
}
```

BUFFIL.C

```
/*          buffil
    A Stephen McDonald creation
    This is part of the software that's supposed to scan the cw
lasers and take data in over the a/d's.  This routine is called by
digest.  It writes the data that is coming in onto a disk.  To do this
as quickly as possible, double buffering is used, the two buffers are
called b2 and b3.  A third buffer, b1, is the input buffer into which
an interrupt routine, ADIR, stuffs the incoming data and from which
buffil() and other like minded programs remove data.
*/

#include    "sy:prog.req"
buffil (low, high)
register    int     *low;   /* low points to the beginnig of the new data
                               in b1 */
int *high;              /* high points to the word past the top of the new
                           data in b1 */
{
    extern  int     *bin;   /* this is a pointer to either b2 or b3,
                               whichever is the input buffer  */
    extern  int     *bout;  /* this points to the one that bin doesn't.
                               bout is the buffer being written on the
                               disk. */
    extern  int     *binmax;/* the top of the input buffer */
    extern  int     *boutmax;/* the top of the output buffer */
    extern  int     *binp;  /* the pointer used to read data into bin */
    extern  int     *b1max; /* the top of the interrupt input buffer */
    extern  int     *b1;    /* the bottom of the input buffer */
    extern  int     blk;    /* this is the block on the disk file that
                               is currently being written to */
    extern  int     chan;   /* the channel for the output of the file
                               to the disk */
```

```
    for ( ; ;)
            {
            if (low == b1max)
                    low = b1;       /* if at top of interrupt buffer,
                                       reset to bottom */
            if (low == high)/* do the loop until have read all new data */
                    break;
            if (binp == binmax)     /* if the input buffer (b2 or b3)
                                       is full, then time to use
                                       the double buffering and switch
                                       buffers */
                    {
                    chango();       /* switch the two buffers */
                    WAITM (chan);   /* wait until the last buffer has
                                       finished spilling it's guts to the
                                       disk */
                    WRITEM (chan, blk, bout, (boutmax - bout));
                    /* both WAITM and WRITEM are from the file PROG.REQ.
                       WAITM waits until all activity on the specified
                       channel has ceased, and WRITEM puts a write request
                       in the I/O que and then returns immediately to the
                       calling program.   Note that since bout boutmax
                       and bout are integer pointers, boutmax - bout
                       gives the number of words between the two, not
                       the number of bytes that you would expect if they
                       were integers */
                    blk += (boutmax - bout)/256;    /* increment the block
                                                       pointer to the file
                                                       on the disk */
                    }
            *binp++ = *low++;
            }
    return;
}
/*                  chango
    A Stephen McDonald creation
    This is part of the software that's supposed to scan the cw
```

```
lasers and take data in over the a/d's.  This routine is called by
buffil().
    Since I'm double buffering, there is an input buffer and
an output buffer.  chango() switches the identity of these buffers
between two permanant buffers, b2 and b3.
*/

chango()
{
    extern  int     *bin;
    extern  int     *bout;
    extern  int     *binp;
    extern  int     *binmax;
    extern  int     *boutmax;

    int     *temp;

    temp = bin;
    bin = bout;
    bout = temp;

    temp = binmax;
    binmax = boutmax;
    boutmax = temp;

    binp = bin;

    return;
}


DISPLY.C
/********** display
/* A Stephen McDonald creation
    This is part of the software that's supposed to scan the cw
lasers and take data in over the a/d's.  This routine is called by
digest.  display() prints the spectra on the VT105 screen.
*/
```

```
#include    "sy:graph.lib"
#include    "sy:sam.h"

display (low, high)
int *low;           /* the pointer to the beginning of the
                       new data */
int *high;          /* the pointer to the end of the new data */
{
    extern  struct  head1   h1;
                            /* h1 is the structure that contains all
                               the variables that are written in
                               the first block of each scan */
    extern  int     *b1;    /* the bottom of the interrupt buffer */
    extern  int     *b1max; /* the word past the end of the interrupt
                               buffer */
    static  npoint= 0;      /* This is a counter that keeps track of how
                               many points have been processed*/
    int     i;              /* This is a counter that keeps track
                               of how many sets of points have been
                               processed this time around.  It is
                               returned to digest() so that
                               digest can keep track  */
    int     nskip = 4;      /* Instead of having display graph every
                               point (nskip = 1) it can graph
                               every other point (nskip = 2)
                               or every third point (nskip = 3)
                               etc. */
    int     offset = 57;    /* this is the seperation between the graphs
                               on the screen.  Y values can be from 0
                               to 230, so if you want to put two graphs
                               on the screen, one will go from 0 to 114
                               and the second from 115 to 229 (offset =
                               115), if you want three, offset = 76 and
                               for four graphs offset = 57.  */
    int     scfactor = 72;  /* this is a scaling factor to reduce the data
```

```
                        which can be from 0 to 4096 to something
                        that will fit on the graph.  If you are
                        graphing four spectra, then you want the
                        maximum value to be 57 so that all four
                        can fit on the screen without overlaping.
                        Therefore scfactor = 72 since 4096/72 = 57
                        (give or take a small remainder ).  */
register        int     *pi;    /* the pointer that will put out
                                   the data */

i = 0;
while (low != high)
        {
        npoint += 1;    /* increment the number of sets of data
                           points processed.  */
        i += 1;         /* increment the number of sets of data
                           points processed this go around */
        if ((npoint % nskip) == 0)
                {                       /* want to plot every
                                           nskip'th point.  If the
                                           number of points taken
                                           can be divided evenly
                                           by nskip, then it's time
                                           to plot some points */
                printm (GRA_ON);/* turn graphich on */
                pi = low;       /* load up the pointer with the
                                   location of the first point
                                   you want to put out */
                printm (GL0);   /* tell the terminal that data for
                                   graph zero is about to come */
                gsend  (*pi++/scfactor);
                if (pi == b1max)/* have to make sure don't run over*/
                        pi = b1;/* the top of the buffer */
                printm (GL1);   /* tell the terminal that data for
                                   graph 0 is about to come */
                gsend ( (*pi++/scfactor) + offset);
                /* send the data value scaled down by scfactor with
                   gain set by the user in the beginning of digest
```

```
            and with an offset so that the spectra so not
            overlap */
        if (pi == b1max)
                pi = b1;
        printm (GL0);    /* the data is sent graph 0 graph 1
                            graph 0 graph 1 because that's
                            the way the terminal has to have
                            it for dual strip chart */
        gsend ( (*pi++/scfactor) + 2*offset);
        if (pi == b1max)
                pi = b1;
        printm (GL1);
        if (h1.nchan > 3)
                {
                gsend ( (*pi++/scfactor) + 3*offset);
                if (pi == b1max)
                        pi = b1;
                }
        else
                gsend (3*offset);        /* this is necessary
                                            to make sure graph
                                            1 is plotting the
                                            same x values an
                                            graph 0. */
        printm (GRA_OFF);                /* now turn graphics
                                            off */

        }
low += h1.nchan;
if (low >= b1max)
        low += b1 - b1max;
/* Want low to point to the beginning of a set of
   data, where a set is one point from each channel.
   Therefore if low goes over b1max, I have to make
   sure it points to the beginning of the next set
   of data after b1.  The low += b1 - b1max can
   also be read as low equals b1 plus (low minus
   b1max) */
}
```

```
    return (i);
}


/***                 prttim
    This is a part of the program scncom.sav.  It's job is to
print the time on the screen in the format
min: XXX sec: XX
where X is a digit.  a carriage return but not a line feed appears at the
end of the output, allowing a kind of digital clock on the screen


ARGUMENTS:  time:   the time in seconds that you want put on the screen.
                    should be an integer.

RETURNS:    nothing

*/

#include    <sy:std.h>


PRTTIM.C
prttim(time)
int time;
    {
    int     i;
    char    outbuf[30];/*the string that will be sent to the terminal*/
    char    *pc;            /* character pointer */


    pc = cpystr (outbuf, "min: ", NULL);
    i = time/60;    /* i is now the # of minutes */
    time = time % 60;       /* time is now the number of secs. left over*/
    if (i == 0)
            *pc++ = '0';    /* if no minutes, put in a zero */
    else
            pc += itob (pc, i, 10); /* convert to a buffer */
```

```
pc = cpystr (pc, " sec: ", NULL);
if (time/10 == 0)
        *pc++ = '0';    /* if seconds has only one digit, write in a
                           zero to make it two */
pc += itob (pc, time, 10);      /* now convert seconds to a buffer */
*pc++ = CR;             /* put in a carriage return but no line feed*/
*pc = '\200';           /* and terminate for printm */
printm (outbuf);
return;
}
```

Appendix B
Hamiltonian for an Octet Sigma State

```
      SUBROUTINE MATRIX(BLOCK,P,H,J,MAXP)
C     THIS IS A HAMILTONIAN FOR AN OCTET SIGMA, OCTET SIGMA
C     TRANSITION.
C     HYPERFINE IS NOT INCLUDED.
C     THE HAMILTONIAN IS SET UP FOR CASE B WHERE N IS THE
C     IMPORTANT QUANTUM NUMBER SO FOR EACH SPIN COMPONENT N IS
C     CALCULATED FROM THE GIVEN J.
C
C     BLOCK = WHICH HAMILTONIAN YOU WANT. BLOCK = 1 FOR E PARITY
C      LOWER STATE, 2 FOR F PARITY LOWER STATE, 3 FOR E PARITY
C     UPPER STATE AND FOR F PARITY UPPER STATE.
C     P() =   ARRAY OF PARAMETERS
C     P(1) =  B"
C     P(2) =  D"
C     P(3) =  SPIN SPIN CONSTANT LAMDA FOR THE LOWER STATE
C     P(4) =  SPIN ROTATION CONSTANT GAMMA FOR THE LOWER STATE
C     P(5)-P(6) = RESERVED FOR LOWER STATE HYPERFINE PARAMETERS
C     P(7) = T(v), = UPPER STATE(J=0) - LOWER STATE (J=0)
C     P(8) =  B'
C     P(9) =  D'
C     P(10)=  LAMDA'
C     P(11)=  GAMMA'
C     P(12)-P(14) = RESERVED
C
C     H()     IS THE HAMILTONIAN FOR THE GIVEN J.  MATRIX RETURNS
C     THE             VALUES IN H()
C     J = THE J VALUE
C
C     MAXP() IS AN ARRAY WHICH HOLDS THE DIMENSION OF THE
C                   HAMILTONIANS WHICH IS 8X8 FOR BOTH STATES
C     AS MENTIONED, THIS IS A CASE B HAMILTONIAN.  HOWEVER THE
C     HAMILTONIAN IS ARRANGED IN BLOCKS OF J BECAUSE EXCLUDING
C     HYPERFINE THERE ARE NO OFF DIAGONAL ELEMENTS IN J WHILE
C     SPIN SPIN GIVES OFFDIAGONAL ELEMENTS IN N.


      REAL*8 H(12,12), P(75)
      INTEGER MAXP(10)
      REAL*8 J, DOUBLE
      REAL*8 DSQRT
      INTEGER BLOCK
      INTEGER W
      INTEGER N
      INTEGER I1, I2, I3

C     FIRST ZERO OUT THE HAMILTONIAN
      DO 20 I1 = 1,12
```

```
                DO 10 I2 = 1, 12
                        H(I1,I2) = 0.0
10              CONTINUE
20      CONTINUE
        IF (BLOCK .LE. 2) THEN
                W = 0
        ELSE
                W = 7
        END IF
        IF ((BLOCK .EQ. 2) .OR. (BLOCK .EQ. 4)) GO TO 60
C       E PARITY FIRST
C       START WITH THE DIAGONAL ELEMENTS
C       J = N - 1/2
        N = J + 0.5
        H(4,4) = P(1+W)*N*(N+1) - P(2+W)*N*N*(N+1)*(N+1) +
     *  5*P(3+W)*(1-12.0/(2*N-1) + 9.0/(2*N+3)) -
     *  .5*P(4+W)*(N+16)
C       NEXT J = N + 3/2
        N = J - 1.5
        H(3,3) = P(1+W)*N*(N+1) - P(2+W)*N*N*(N+1)*(N+1)
     *  + 3*P(3+W)*(1 - 7.0/(2*N-1) + 20.0/(2*N+3))
     *    +.5*P(4+W)*(3*N-12)
C       J = N - 5/2
        N = J + 2.5
        H(2,2) = P(1+W)*N*(N+1) - P(2+W)*N*N*(N+1)*(N+1)
     *  - P(3+W)*(1 + 45.0/(2*N-1))
     *    - .5*P(4+W)*(5*N+12)
C       J = N + 7/2
        N = J - 3.5
        H(1,1) = P(1+W)*N*(N+1) - P(2+W)*N*N*(N+1)*(N+1)
     *  - 7*P(3+W)*(1 - 3.0/(2*N+3))
     *    + 3.5*P(4+W)*N
C       NOW FOR THE OFF DIAGONAL ELEMENTS
C       J = S + N.  WHEN N =1, J CAN ONLY EQUAL 9/2,7/2,5/2 I.E.
C       N+7/2, N+5/2 AND N+3/2.  NO J = N+1/2 ALLOWED.  SIMILAR
C       RESTRICTIONS EXIST FOR THE OTHER J'S AT LOW N.
        IF (J .LT. 1.5) GO TO 50
        N = J + 2.5
        DOUBLE = (N+J+4.5) * (N+J+3.5) * (N+J-4.5) *
     *          (N+J-3.5) * (N-J+3.5) * (N-J+2.5) * (J-N+4.5) *
     *          (J-N+5.5)/((2*N-1) * (2*N-1) * (2*N-3) * (2*N+1))
        H(4,2) = 0.5*P(3+W)*DSQRT(DOUBLE)
        IF (J .LT. 2.5) GO TO 40
        N = J + 0.5
        DOUBLE = (N+J+4.5) * (N+J+3.5) * (N+J-4.5) *
     *          (N+J-3.5) * (N-J+3.5) * (N-J+2.5) * (J-N+4.5) *
     *          (J-N+5.5)/((2*N-1) * (2*N-1) * (2*N-3) * (2*N+1))
        H(3,4) = 0.5*P(3+W)*DSQRT(DOUBLE)
        IF (J .LT. 3.5) GO TO 30
        N = J - 1.5
        DOUBLE = (N+J+4.5) * (N+J+3.5) * (N+J-4.5) *
```

```
     *          (N+J-3.5) * (N-J+3.5) * (N-J+2.5) * (J-N+4.5) *
     *          (J-N+5.5)/((2*N-1) * (2*N-1) * (2*N-3) * (2*N+1))
        H(1,3) = 0.5*P(3+W)*DSQRT(DOUBLE)
        H(3,1) = H(1,3)
30      H(4,3) = H(3,4)
40      H(2,4) = H(4,2)
50      GO TO 90

60      CONTINUE
C       NOW FOR THE F PARITY BLOCK
C       J = N + 1/2
        N = J - 0.5
        H(4,4) = P(1+W) * N*(N+1) - P(2+W)*N*N*(N+1)*(N+1)
     *  + 5*P(3+W)*(1-9.0/(2*N-1) + 12.0/(2*N+3)) +
     *   .5*P(4+W)*(N-15)
C       J = N - 3/2
        N = J + 1.5
        H(3,3) = P(1+W)*N*(N+1) - P(2+W)*N*N*(N+1)*(N+1)
     *  + 3*P(3+W)*(1 -20.0/(2*N-1) + 7.0/(2*N+3))
     *    - .5*P(4+W)*(3*N + 15)
C       J = N + 5/2
        N = J - 2.5
        H(2,2) = P(1+W)*N*(N+1) - P(2+W)*N*N*(N+1)*(N+1)
     *  - P(3+W)*(1 - 45.0/(2*N+3))
     *    + .5*P(4+W)*(5*N-7)
C       J = N - 7/2
        N = J + 3.5.
        H(1,1) = P(1+W)*N*(N+1) -  P(2+W)*N*N*(N+1)*(N+1)
     *  - 7*P(3+W)*(1 + 3.0/(2*N-1))
     *    - 3.5*P(4+W)*N
C       NOW FOR THE OFF DIAGONAL ELEMENTS
        IF (J .LT. 1.5) GO TO 90
        N = J + 3.5
        DOUBLE = (N+J+4.5) * (N+J+3.5) * (N+J-4.5) *
     *          (N+J-3.5) * (N-J+3.5) * (N-J+2.5) * (J-N+4.5) *
     *          (J-N+5.5)/((2*N-1) * (2*N-1) * (2*N-3) * (2*N+1))
        H(3,1) = 0.5*P(3+W)*DSQRT(DOUBLE)
        IF (J .LT. 2.5) GO TO 80
        N = J + 1.5
        DOUBLE = (N+J+4.5) * (N+J+3.5) * (N+J-4.5) *
     *          (N+J-3.5) * (N-J+3.5) * (N-J+2.5) * (J-N+4.5) *
     *          (J-N+5.5)/((2*N-1) * (2*N-1) * (2*N-3) * (2*N+1))
        H(4,3) = 0.5*P(3+W)*DSQRT(DOUBLE)
        IF (J .LT. 3.5) GO TO 70
        N = J - 0.5
     *          (N+J-3.5) * (N-J+3.5) * (N-J+2.5) * (J-N+4.5) *
     *          (J-N+5.5)/((2*N-1) * (2*N-1) * (2*N-3) * (2*N+1))
        H(2,4) = 0.5*P(3+W)*DSQRT(DOUBLE)
        H(4,2) = H(2,4)
70      H(3,4) = H(4,3)
80      H(1,3) = H(3,1)
```

```
90      IF (BLOCK .GT. 2) THEN
                H(1,1) = H(1,1) + P(7)
                H(2,2) = H(2,2) + P(7)
                H(3,3) = H(3,3) + P(7)
                H(4,4) = H(4,4) + P(7)
        END IF
        RETURN
        END

        SUBROUTINE NAMEIT (NAM, M)
C       ASSIGNS NAMES TO PARAMETERS, M IS MAXIMUM NUMBER OF NAMES
        REAL*8  NAM(M)
        REAL*8 NAMES(75) /2HB", 2HD", 4HLAM", 4HGAM", 4HNULL,
     *  4HNULL,4HT(v),2HB',2HD',4HLAM',4HGAM',1HO,1HO,1HO,
     *  1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,
     *  1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,
     *  1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,
     *  1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,
     *  1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,1HO,
     *  1HO/
        DO 10 I = 1, M
                NAM(I) = NAMES(I)
10      CONTINUE
        RETURN
        END
```