

Simultaneous Two-Ion Mass Spectroscopy

Szymon M. Rusinkiewicz

Submitted to the Department of Physics
in Partial Fulfillment of the
Requirements for the Degree of

Bachelor of Science

at the
Massachusetts Institute of Technology
May 1995

©1995 Massachusetts Institute of Technology
All Rights Reserved

Signature of Author

Certified by

Professor David E. Pritchard
Thesis Supervisor

Accepted by

Professor Aron Bernstein
Thesis Coordinator

Simultaneous Two-Ion Mass Spectroscopy

Szymon M. Rusinkiewicz

Abstract

It is possible to compare the masses of single ions with an accuracy of one part in 10^{11} (an order of magnitude better than the current state of the art) by comparing the cyclotron frequencies of two ions of different species confined simultaneously in a Penning trap. Computer simulations were used to study the motion of two simultaneously trapped ions in a trap with predicted, partially controllable, imperfections; the results were checked against theoretical predictions for ideal Penning traps. Attention was devoted to schemes that offer high precision by placing both ions in the same orbit.

Thesis Supervisor: Dr. David E. Pritchard
Professor of Physics

Contents

1	Introduction	7
1.1	Previous Results	7
1.2	Applications of Increased Precision	7
2	Single Ion Measurements	9
2.1	Theory	9
2.2	Measurement and Data Analysis Techniques	10
2.3	Motivation for Simultaneous Two-Ion Trapping	11
3	Two-Ion Theory and Simulations	13
3.1	Basic Two-Ion Theory	13
3.2	Anharmonicity	14
3.3	Elliptical Common Modes	19
3.4	Cooling and Heating Drives	20
3.5	Ring Modulation and Common Mode Cooling	23
4	Two Ion Implementation	27
4.1	Electronics	27
4.2	Two-Ion Procedures and Results	27
	Appendix: Code Listing	31
	References	41

Introduction

Precise knowledge of the atomic masses of ions is useful in many areas of active research. By studying single ions confined in a Penning trap the ICR group at MIT has determined the atomic masses of several atoms and molecules with an accuracy of approximately one part in 10^{10} . It is anticipated that simultaneously trapping two ions will allow for further reduction in uncertainty, by reducing the errors due to magnetic field fluctuations. This thesis examines the expected benefits of simultaneous trapping, presents the results of computer simulations of the behavior of two simultaneously trapped ions of nearly equal mass, and describes our implementation of this scheme.

1.1 Previous Results

The precise measurements obtained by this group using single-ion trapping have

- improved the atomic mass table. The current project has already improved the best measurements of several elements by as much as three orders of magnitude. [DiFilippo 1994]
- helped in the calibration of gamma-ray energies, through measurements of the change in mass that accompanies γ -ray emission. Such measurements can ultimately lead to more precise values for the molar Planck constant $N_A h$ and the fine structure constant α .
- demonstrated classical squeezing of amplitude noise below the thermal noise level. [DiFilippo 1992]
- provided a precise atomic mass of ^{28}Si . Together with the masses of ^{29}Si and ^{30}Si , as well as lattice-spacing measurements in Silicon crystals, this could be used to establish a Silicon-based “atomic” mass standard, to replace the current “artifact” standard.

1.2 Applications of Increased Precision

There are many more applications of precise mass measurements, and this project could make substantial contributions if its accuracy could be improved by one or two orders of magnitude. In particular, measurements at such an accuracy would

-
- yield the atomic masses of elements such as Silicon, Sulphur and Titanium, for comparison with gamma rays emitted in neutron capture experiments.
 - help determine the rest mass of the electron neutrino — better upper bounds on this mass can be obtained from the ${}^3\text{H}^+ - {}^3\text{He}^+$ mass difference.
 - allow the determination of chemical binding energies in cases where this is difficult to accomplish by other means. The relation $E = \Delta mc^2$ would permit these energies to be “weighed” directly.

Single Ion Measurements

So far, most of our efforts have been focused upon a mass measurement method in which one species of ion at a time has been trapped. Here we present some basic theory behind the physics of a Penning Trap, and the techniques used in this single-ion scheme. [Boyce, Brown & Gabrielse]

2.1 Theory

The Penning trap consists of three electrodes, together forming a complete hyperboloid of revolution (Figure 2.1). The two endcaps are usually held at the same potential, which is generally above that of the ring (for a positive ion trap). This produces a quadrupole electric field resulting in axial confinement of a positive ion, while a large magnetic field produces radial confinement.

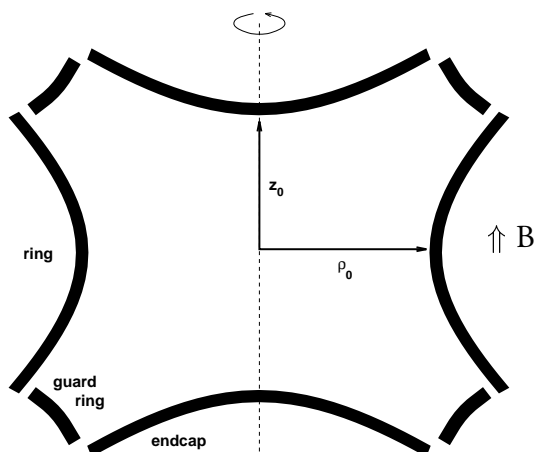


Figure 2.1: Schematic of Trap

A trapped ion has three possible modes of oscillation. First, there is the axial mode, with frequency

$$\omega_z = \frac{qV_T}{md^2}, \quad (2.1)$$

where

$$d = \sqrt{\frac{\rho_0^2}{4} + \frac{z_0^2}{2}} \quad (2.2)$$

is a “characteristic size” of the trap. Typically, the mode with the highest frequency is the cyclotron mode, due mostly to the magnetic field. If there were no electric field, this mode would have frequency

$$\omega_0 = \frac{qB}{m}, \quad (2.3)$$

but inside the trap it is perturbed to

$$\omega_{c'} = 1/2 \left(\omega_0 + \sqrt{\omega_0^2 - 2\omega_z^2} \right). \quad (2.4)$$

The other radial mode is the magnetron, a drift around the center of the trap with frequency

$$\omega_m = 1/2 \left(\omega_0 - \sqrt{\omega_0^2 - 2\omega_z^2} \right), \quad (2.5)$$

which is typically the lowest of the three modes.

Since the free-space cyclotron frequency ω_0 is inversely proportional to the mass of an ion, we can determine the mass ratios of ions in the trap by comparing their frequencies. In terms of the frequencies of the measured trap modes,

$$\omega_0^2 = \omega_{c'}^2 + \omega_z^2 + \omega_m^2. \quad (2.6)$$

We see from this equation that the ultimate precision of our determination of ω_0 is limited by our ability to measure the highest-frequency mode, the trap cyclotron, while the measurement of the frequencies of the other two modes is much less critical.

2.2 Measurement and Data Analysis Techniques

Our goal is to measure the ratio of the masses of two kinds of ions, through equation 2.3. To do this, we must measure the frequencies of the three normal modes of each ion, under trap conditions that are as similar as possible. Our detector, however, is only sensitive to axial vibrations, and can therefore only measure the axial frequency directly. Since the cyclotron frequency is the largest, and hence the one that must be measured most accurately (Equation

2.6), we use a pulse-and-phase method to find its frequency. That is, we apply a pulse that excites the cyclotron mode, and some time later apply a coupling pulse that transfers the phase of the cyclotron mode to the axial. Thus, we can determine the phase that accumulated in the cyclotron mode during the time between the pulses.

By repeating the above with a series of ever-longer delay times, we can gain enough information to be able to “unwrap” the phases; that is, we determine how many complete cycles the cyclotron phase underwent during the delay time. Finally, we determine the cyclotron frequency by dividing the total accumulated phase by the time between the excitation and coupling pulses.

The final mass ratio is obtained by alternating the frequency measurements for two species of ions. Because of the drift in magnetic field (and other systematic drifts), we swap species of ions in the trap several times over a night’s run. We can then empirically fit to the field drift, and thus compensate for its effects. Nevertheless, the largest source of uncertainty in this scheme is the fluctuation in the magnetic field, despite the fact that we do the runs during the night to minimize the magnetic field noise from the subway, a nearby elevator, and other sources. [Boyce, DiFilippo 1994]

2.3 Motivation for Simultaneous Two-Ion Trapping

Since the major limiting factor in the current precision is the time variation of the magnetic field, further increases in precision require some new methods to overcome its effects. Two possibilities for doing this are improving the stability of the fields inside the trap and simultaneously trapping both species of ions. The former could be accomplished by shielding the trap against external fluctuations by means of, for example, a superconducting shield. Adding such a shield, however, is likely to require substantial changes to the apparatus, as well as special attention to securing the shield against mechanical vibrations, displacements, and tilts, which would otherwise limit its usefulness.

Simultaneous two-ion trapping also has some shortcomings. If the ions are to be confined in separate traps, there are many systematic errors associated with the difficulty of controlling the uniformity of the magnetic field across such large distances. If they are confined in the same trap, the electrostatic interaction of the two ions adds complexity. Furthermore, we must be careful to keep the ions in orbits of the same root-mean-squared radius, to avoid systematic errors due to field inhomogeneity.

There are several advantages to the single-trap two-ion scheme. [Cornell 1990, Kuchnir] It would not only greatly reduce the error associated with the variation of the magnetic field (as well as other time-varying parameters, such as the trap voltage), but would also greatly reduce the number of new ions that have to be made, thus requiring only one set of procedures to eliminate bad ions (i.e., inadvertently produced ions of the wrong species), as well as making comparisons of hard-to-produce ions more practical. In addition, it requires little additional

hardware and no changes to the trap. Therefore, since mid-1994 we have focused, to a large extent, on understanding and implementing the (single-trap) two-ion scheme. The principal results are presented in the following chapters.

Two-Ion Theory and Simulations

The equations of motion of two simultaneously trapped ions are difficult to solve analytically, especially if one considers the effect of imperfections of the trap. Therefore, to aid in understanding of the actual behavior of the ions, I wrote a computer program to simulate the ions by numerically solving their equations of motion. The derivation of these equations and the results of the simulations are now presented.

3.1 Basic Two-Ion Theory

The simplest approximation to the two-ion problem is to consider only the radial motion of ions in an ideal Penning trap. Thus, for the time being we ignore any anharmonicity in the trap, as well as the axial motion of the ions. We write the radial position of an ion as the complex number x , whose real and imaginary parts are the radial Cartesian coordinates of the ion. The equations of motion are

$$\ddot{x}_1 = i\frac{qB}{m_1}\dot{x}_1 - \frac{qV_T}{2m_1d^2}x_1 + \frac{k_{Coul}q^2}{m_1}|x_1 - x_2|^{-3}(x_1 - x_2) \quad (3.1)$$

$$\ddot{x}_2 = i\frac{qB}{m_2}\dot{x}_2 - \frac{qV_T}{2m_2d^2}x_2 + \frac{k_{Coul}q^2}{m_2}|x_1 - x_2|^{-3}(x_2 - x_1). \quad (3.2)$$

These are the same as the single-ion equations, except for the Coulomb repulsion term.

We may now perform the standard separation of variables, writing (for each ion)

$$x_m = \frac{\omega_{c'}}{\omega_{c'} - \omega_m} \left(x + \frac{i}{\omega_{c'}} \dot{x} \right) \quad (3.3)$$

$$x_{c'} = \frac{-\omega_m}{\omega_{c'} - \omega_m} \left(x + \frac{i}{\omega_m} \dot{x} \right), \quad (3.4)$$

where $\omega_{c'}$ and ω_m are given by equations 2.4 and 2.5. We substitute these into equations 3.1 and 3.2, to obtain the equations of motion for the cyclotron and magnetron modes separately. Since the cyclotron amplitude is usually small, we are chiefly concerned about the magnetron

modes:

$$\dot{x}_{m_1} = i\omega_{m_1} x_{m_1} + F_{R_1} + \frac{ikq^2}{m_1(\omega_{c_1}' - \omega_{m_1})} |x_{m_1} - x_{m_2}|^{-3} (x_{m_1} - x_{m_2}) + F_{A_1} \quad (3.5)$$

$$\dot{x}_{m_2} = i\omega_{m_2} x_{m_2} + F_{R_2} + \frac{ikq^2}{m_2(\omega_{c_2}' - \omega_{m_2})} |x_{m_1} - x_{m_2}|^{-3} (x_{m_2} - x_{m_1}) + F_{A_2}. \quad (3.6)$$

Here, F_{A_i} and F_{R_i} account for anharmonicity and a rotating coordinate frame, respectively, and in the simplest case may be set equal to zero.

Now, to make the simulation more efficient, we may introduce a rotating coordinate frame. That is, we place ourselves in a reference frame that is rotating at a frequency near that of the magnetron modes, and observe only the motion relative to that frame. Since the changes to the ions' motion occur at frequencies slow relative to the magnetron frequencies, this will speed up the simulation by eliminating the uninteresting magnetron motion, and just modeling the changes to that motion resulting from the ion-ion interaction. In addition, the results of the simulations are more easily understandable.

Therefore, we assume that we are in a coordinate frame rotating with some angular frequency ω_r . We now observe that all of the forces acting on the ions are rotationally symmetric. Because of this symmetry of the forces, the only change to equations 3.5 and 3.6 is that we add terms of the form

$$F_{R_1} = -i\omega_r x_{m_1} \quad (3.7)$$

$$F_{R_2} = -i\omega_r x_{m_2}, \quad (3.8)$$

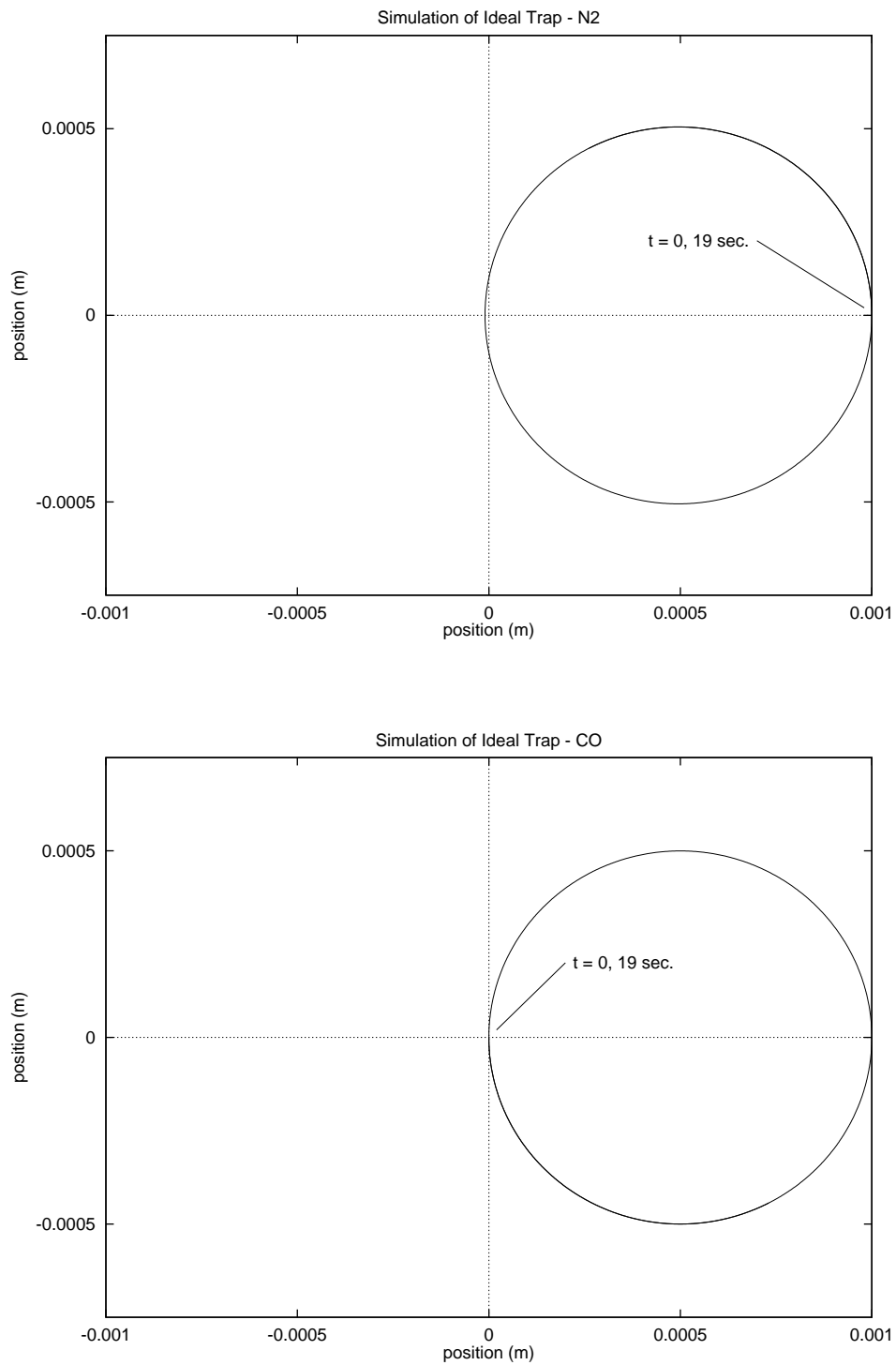
as is easily found from the separation of variables transformation.

The motion resulting from these idealized equations is shown in Figure 3.1. The figure depicts a simulation of singly ionized N_2 and CO, with an initial separation of 1 millimeter. We see the magnetron modes of the two ions coupled into a “sum” and a “difference” mode. The former is the motion of the center of mass of the system, and moves in a circle around the center of the trap (Our frame of reference rotates at a frequency $\bar{\omega}_m$ close to the average magnetron frequency, so the center of mass appears stationary). The latter is the motion of the ions around their center of mass. Depending on the initial conditions, it is possible for these coupled modes to have various amplitudes, and it is also possible to change the amplitudes with “cooling” or “heating” drives.

3.2 Anharmonicity

Let us now look at the effects of a non-ideal trap on the motion of the ions. We assume that the trap potential can be written as follows:

$$V = \frac{V_r}{2} \sum_{n \text{ even}} C_n \frac{r^n}{d^n} P_n(\cos \theta), \quad (3.9)$$



**Figure 3.1: Two-Ion coupling in ideal trap –
Note that the radius of N_2 is larger than that of CO
by an amount greater than the mass ratio of the two.**

where r is the radius of the ion, i.e. $|x|$. C_4 and C_6 are constants describing the strength of the z^4 and z^6 dependence of the electric field. In an ideal quadrupole trap, they would be zero. However, measurement has shown that in our trap C_6 is significantly different from 0, and C_4 is controllable by the setting of the guard ring voltage. Typical values for C_6 are on the order of $4 \cdot 10^{-4}$, while $C_4 \approx -6.3 \cdot 10^{-3} V_G$, where V_G is the guard ring voltage. So, we use these experimentally determined values for C_4 and C_6 , and observe the effect on the ions' motion. The effect on equations 3.5 and 3.6 is to introduce additional terms corresponding to the fourth- and sixth-order corrections to the electric field. For ion 1, the term is

$$F_{A_1} = \frac{i}{\omega_{c_1}' - \omega_{m_1}} \omega_{z_1}^2 \left(3C_4 \frac{\langle z^2 \rangle}{d^2} - \frac{3}{4} C_4 \frac{|x_{m_1}|^2}{d^2} + \frac{15}{16} C_6 \frac{|x_{m_1}|^4}{d^4} \right) x_{m_1}, \quad (3.10)$$

assuming small axial amplitude. The term F_{A_2} is analogous.

The anharmonicity of the trap also has an effect on the axial frequencies of the ions. Thus, as the ions move in and out of the center of the trap, their axial frequencies change. From equation 3.9 we can determine that (for each ion)

$$\omega_z = \omega_{z_0} \cdot Z_A \cdot Z_E, \quad (3.11)$$

where

$$Z_A = \sqrt{1 - 3C_4 \frac{|x_m|^2}{d^2} + \frac{45}{8} C_6 \frac{|x_m|^4}{d^4}} \quad (3.12)$$

is the factor due to anharmonicity.

Z_E is a further perturbation to the axial frequencies, due to the electrostatic effects of the ions on each other. Thus, the presence of one ion creates an effective anharmonicity at the location of the other ion, and consequently changes its axial frequency. For small axial amplitudes, the magnitude of this perturbation is

$$Z_E = 1 - \left(\frac{k_{Coul} q^2}{4m \omega_{z_0}^2 |x_1 - x_2|^3} \left(1 - \frac{9(\langle z_1^2 \rangle + \langle z_2^2 \rangle)}{2|x_1 - x_2|^2} \right) \right) \quad (3.13)$$

What are the effects of this anharmonicity on the ions' motion? First, the equations predict a variation in the axial frequencies of the ions as they move in and out of the center of the trap, as shown in Figure 3.2. This is consistent with the observed axial frequency variations of as much as 1 Hz when there are two ions in the trap. Another behavior predicted by the simulations is the presence of “decoupled” modes, such as the one shown in Figure 3.4. That is, if the anharmonicity is large enough, the ions' motion will no longer decompose into sum and difference modes, but will execute nearly independent single-ion motion for certain starting conditions. Consequently, the radius of the ions will not change much from the initial value. Having the ions in such a mode during mass measurements would clearly be unacceptable,

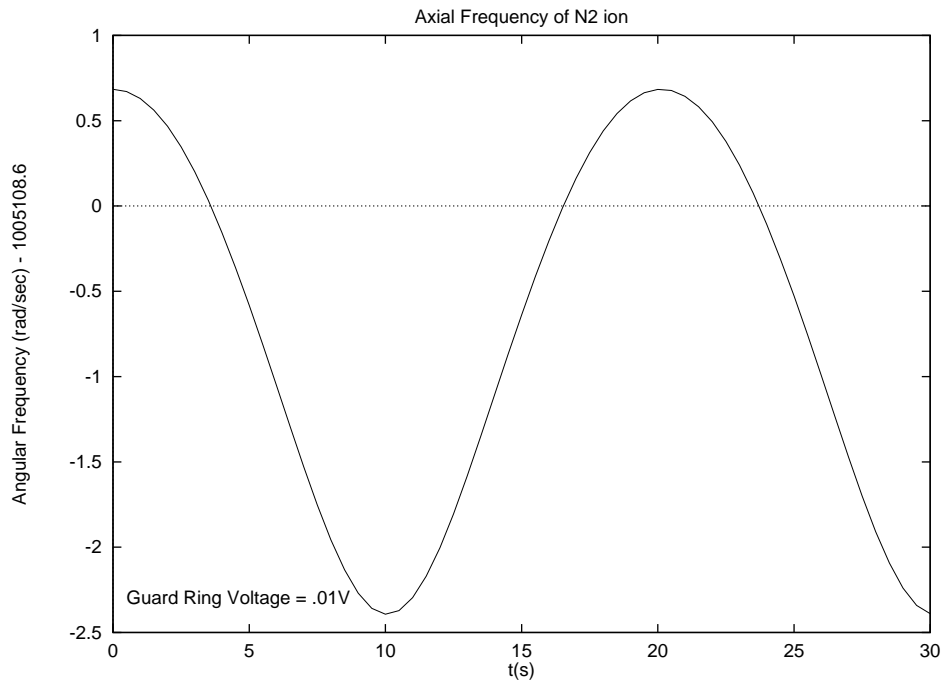


Figure 3.2: Effect of anharmonicity on axial frequencies

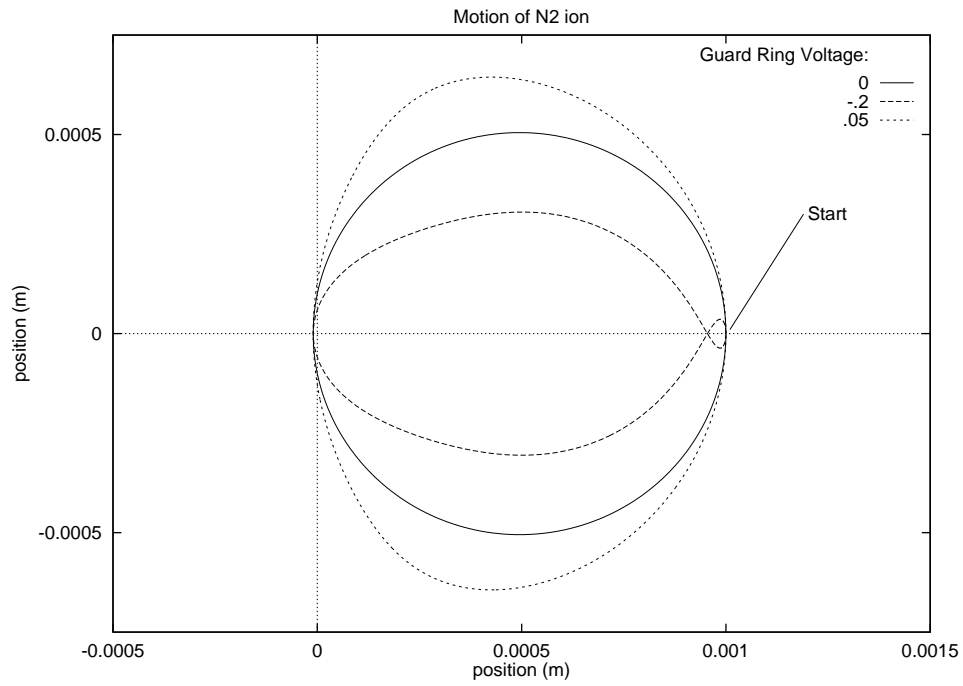


Figure 3.3: Effect of anharmonicity on orbits

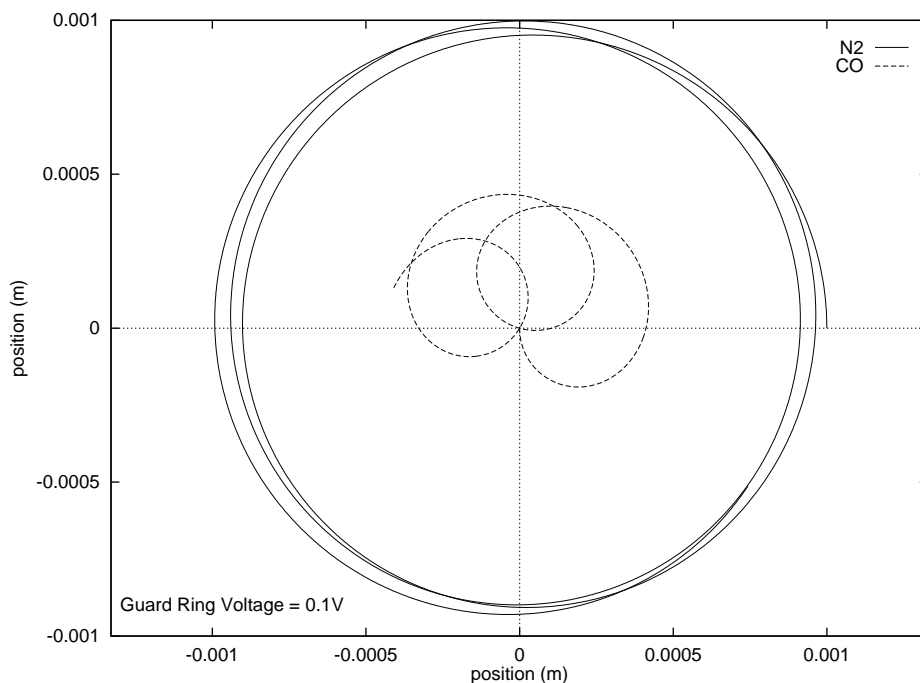


Figure 3.4: Typical Decoupled Mode

because of the large systematic difference in radii. Therefore, we must be especially careful in setting the guard rings to ensure that the ions do not get into an uncoupled mode.

A further effect of anharmonicity is that it causes axial frequency to be dependent on axial amplitude. Thus, as the ions decay by transferring their axial energy into the detector, they change their frequency. This shift in frequency is measurable, and in fact provides the best method of estimating the anharmonicity actually present in the trap. Furthermore, measuring the magnitude of these “chirps” as a function of magnetron radius provides an estimate of C_6 .

The simulations also indicate the possibility of reducing the systematic error due to a nonuniform magnetic field through careful choice of the guard ring voltage. Since the ratio of the average of the squares of the radii of the two ions varies with the guard ring voltage, it is possible to set it so that the two ions have the same average r^2 . This would compensate for a second-order radial nonuniformity in the magnetic field (which is the lowest order to affect the average cyclotron frequencies). As shown in Figure 3.5, in our simulations of N_2 and CO with an initial separation of 1 mm, this voltage was approximately -0.2 volts. This is, unfortunately, rather large, and may indicate that this suggestion is unsuitable for general use, due to the other undesirable effects operating with such a large anharmonicity would cause. A possible workaround, though, would be to reset the guard ring voltage before the mode-coupling π -pulse, which would alleviate the problems with chirped axial ringdown.

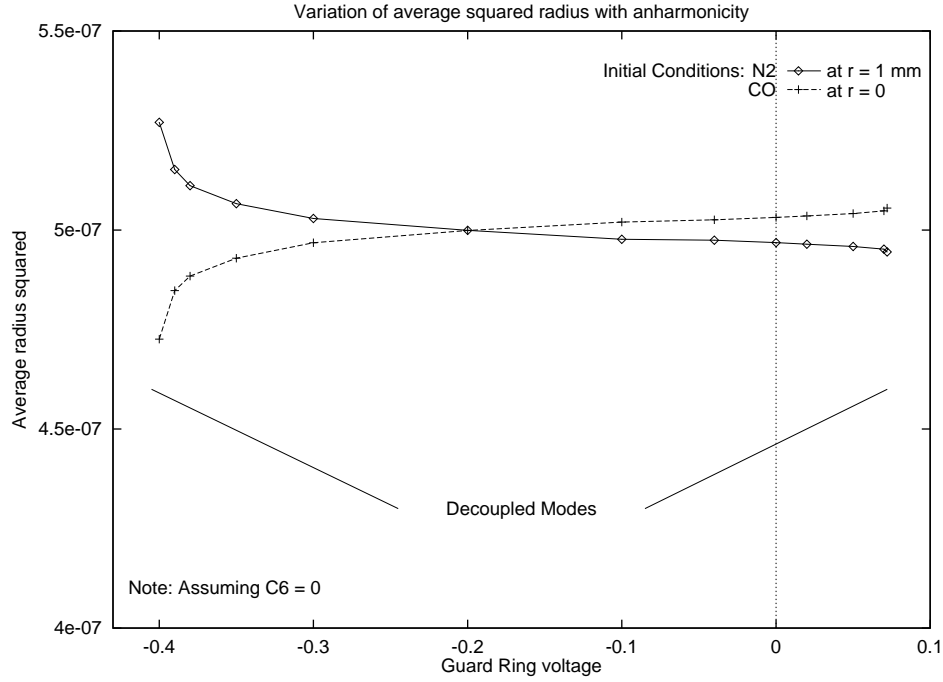


Figure 3.5: Variation of average squared radius with anharmonicity

3.3 Elliptical Common Modes

A particularly striking illustration of the instability associated with large anharmonicity, as well as a valuable check of the correctness of the simulations against theory, comes from considering the motion of the sum mode in a frame of reference in which the difference mode is stationary. Theoretical arguments predict that the sum mode will move in ellipses, with eccentricity dependent on C_4 .

An approximate derivation of this behavior starts by writing

$$x_1 = \delta(1 + \sigma)e^{i\omega t} \quad (3.14)$$

$$x_2 = -\delta(1 - \sigma)e^{i\omega t}, \quad (3.15)$$

where δ (a constant) is the magnitude of the difference mode, and σ is the ratio of the magnitude of sum mode (presumed small) to that of the difference mode. Substituting these into

$$-i\dot{x}_1 = \frac{\kappa(x_1 - x_2)}{|x_1 - x_2|^3} + x_1 f(|x_1|) \quad (3.16)$$

$$-i\dot{x}_2 = \frac{\kappa(x_2 - x_1)}{|x_1 - x_2|^3} + x_2 f(|x_2|), \quad (3.17)$$

where $f(|x|)$ is the component of the force due to anharmonicity, we obtain

$$\sigma\omega - i\dot{\sigma} = (1 + \sigma)f(|\delta(1 + \sigma)|) + (1 - \sigma)f(|-\delta(1 - \sigma)|). \quad (3.18)$$

Since σ is assumed small, we may approximate this by

$$\sigma\omega - i\dot{\sigma} = \delta\text{Re}(\sigma)f'(|\delta|) + \sigma f(|\delta|). \quad (3.19)$$

Finally, writing $\sigma = a + bi$, we get

$$\dot{b} = \left(\delta f'(|\delta|) - (\omega - f(|\delta|)) \right) a \quad (3.20)$$

$$\dot{a} = (\omega - f(|\delta|)) b \quad (3.21)$$

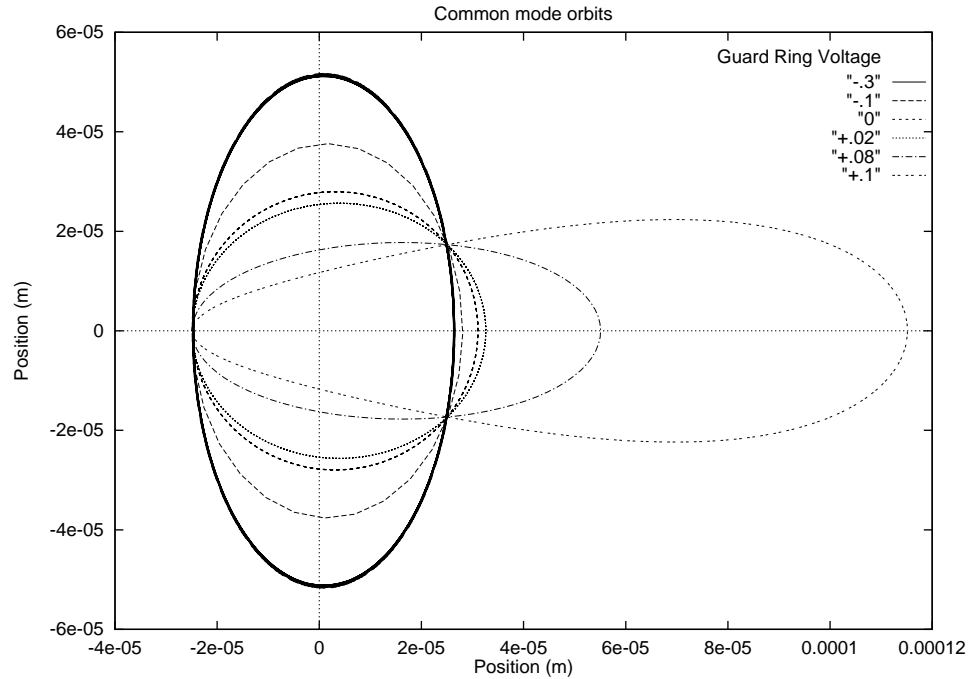


Figure 3.6: Elliptical Common Mode Orbits – seen from reference frame in which the difference mode is stationary.

This describes an ellipse, whose eccentricity depends on f and f' , which in turn depend on the anharmonicity (Figure 3.6). These equations break down for large anharmonicities, and we see this departure from the elliptical shape in the plot for guard ring voltage equal to .1 V.

3.4 Cooling and Heating Drives

Let us now turn to refining the simulation so that it is able to predict the effects of heating and cooling drives applied to the trap. The first improvement necessary is the introduction of a separate variable to describe the state of the measurement circuitry, and of equations to describe the coupling of the axial modes of the ions to it.

We model the detection circuitry as a tuned circuit with capacitance C and inductance L^* . The variable we track is the electric field at the center of the trap, denoted by E . The equations describing the coupling then become

$$\ddot{z}_1 + \omega_{z_1}^2 z_1 = \frac{qE}{m_1} \quad (3.22)$$

$$\ddot{z}_2 + \omega_{z_2}^2 z_2 = \frac{qE}{m_2} \quad (3.23)$$

$$\ddot{E} + E \cdot \frac{1}{LC} + \Gamma \dot{E} = \frac{q(z_1 + z_2)}{\mathcal{M}}, \quad (3.24)$$

where \mathcal{M} is an “effective mass” of the coil[†], given by

$$\mathcal{M} = LC^2 \frac{V}{E}, \quad (3.25)$$

and Γ is the power decay constant of the tuned circuit.

Once we have these equations, we may introduce the effect of the cooling or heating drive. This is a drive that is applied to the guard rings, creating a field that couples the axial and radial modes. It must have a frequency near the sum of the magnetron and axial modes (for cooling) or their difference (for heating), if it is to couple the axial and magnetron modes, and to either reduce (“cool”) the magnetron radius or increase (“heat”) it. As a direct result of the drive, the ions will be driven in the axial direction with a frequency ω_D given by

$$\omega_D = \omega_{cool} - \overline{\omega_m} \quad (3.26)$$

or

$$\omega_D = \omega_{heat} + \overline{\omega_m}. \quad (3.27)$$

Therefore, just as we changed into rotating magnetron coordinates, we may write

$$z_i = \zeta_i e^{i\omega_D t} \quad (3.28)$$

$$E = \varepsilon e^{i\omega_D t}, \quad (3.29)$$

where ε and the ζ_i vary slowly with time.

The actual cooling drive produces a force of the form

$$F = m\kappa(z\hat{x} + x\hat{z}) \cos \omega t, \quad (3.30)$$

* Since the capacitance is extremely small, we often speak of the tuned circuit as just the “coil.”

† Note that \mathcal{M} does not have units of mass.

where κ measures the strength of the applied drive. In the adiabatic approximation, the equations describing the entire cooling or heating process become

$$\dot{x}_i = -i\frac{\Omega}{2}\zeta_i + (\text{R. H. S. of Equation 3.5/3.6}) \quad (3.31)$$

$$\dot{\zeta}_i = i\delta_i\zeta_i - \frac{iq\varepsilon}{2m_i\omega_D} - i\frac{\Omega}{2}x_{m_i} \quad (3.32)$$

$$\dot{\varepsilon} = \left(i\delta_\varepsilon - \frac{\Omega}{2}\right)\varepsilon - \frac{iq(\zeta_1 + \zeta_2)}{2\mathcal{M}\omega_D}, \quad (3.33)$$

where $\delta_i = \omega_{z_i} - \omega_D$ and $\delta_\varepsilon = \sqrt{1/LC} - \omega_D$. Here $\Omega = \kappa/(2\sqrt{\omega_z\omega_m})$ is the Rabi frequency, and parameterizes the strength of the heating or cooling drive.

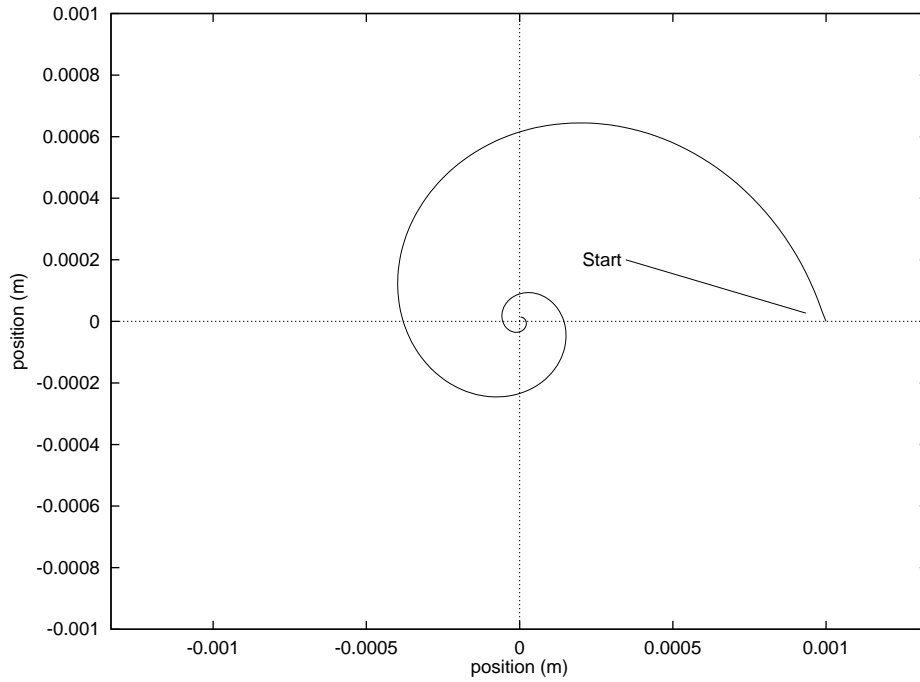


Figure 3.7: Single Ion Cooling - No Anharmonicity

For single ions, the simulations produce the expected results, giving standard cooling (Figure 3.7) and heating (Figure 3.8) behavior. Note that the latter simulation has been performed with anharmonicity present, and that as the ion increases radius, its axial frequency changes enough (due to the anharmonicity) that the difference of the axial and magnetron frequencies no longer matches the frequency of the heating drive. As a result, the rate of heating slows. This phenomenon of the ion's moving out of resonance with the heating or cooling drive is significant, since it limits the effectiveness of heating or cooling at a fixed frequency, and prompts us to consider cooling or heating drives that change frequency with time. We will examine these shortly; for now let us consider two ion cooling.

The simplest two-ion cooling configuration is “sympathetic cooling.” In this scenario, the cooling drive is applied directly to one ion. As it cools, it interacts with the other ion, causing

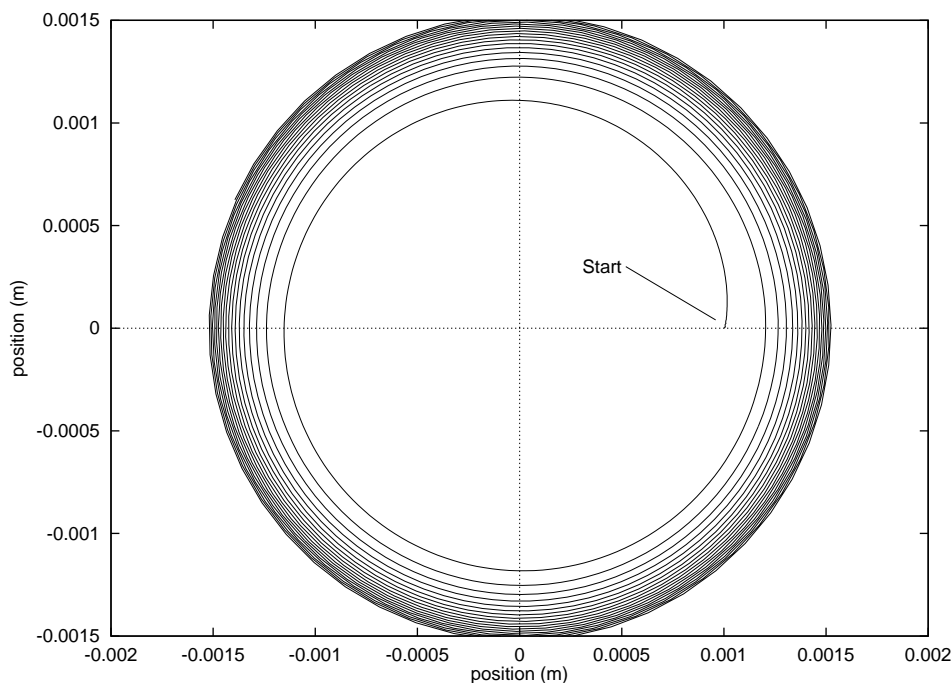


Figure 3.8: Single Ion Heating with Anharmonicity

it to cool sympathetically. The effect is that both ions cool at the same rate. Unfortunately, the sum and difference modes of the ions also cool at approximately the same rate. Since we typically wish to put the ions at a relatively constant radius, our main intent is to cool the common mode while leaving the difference mode at the desired separation. To do this requires a more sophisticated cooling scheme.

3.5 Ring Modulation and Common Mode Cooling

Our ultimate goal is to be able to perform a pulse-and-phase measurement on both ions simultaneously (or nearly so). Therefore, we need some way of applying the correct pulses to both ions, as well as detecting both axial signals. Because our detector is sensitive to such a narrow range of frequencies, we adopt the method of modulating the ring voltage. That is, in addition to the DC voltage between the ring and endcaps, we apply a smaller AC voltage at a frequency ω_{mod} .

The effect of this modulation is that the axial motion of an ion, originally at frequency ω_z , acquires sidebands at $\omega_z \pm n\omega_{mod}$, for all integers n . For a given applied AC voltage, the strength of these sidebands is relatively large for $n = 1$, and decreases rapidly with increasing n , so we concentrate on the $n = 1$ sidebands. So, let us assume that we adjust the DC ring voltage such that the axial frequencies of the two ions are ω_{z_1} and ω_{z_2} , where the coil is centered at the average of these two frequencies. Then, we modulate the coil with $\omega_{mod} \approx |\omega_{z_2} - \omega_{z_1}|/2$. This

creates sidebands of *both* ions on the coil, and so we are able to see the signals of both ions simultaneously.

Since ring modulation is an important part of the two-ion procedures, we certainly want to include it in our simulations. There are two approaches to modeling it. We could analytically work out the effects of the ring modulation on the ions, and the interaction with heating or cooling drives. Alternatively, we could directly model the change in ring voltage with time, and keep the original equations of motion. Although the latter approach requires more computer time, it is conceptually more straightforward and does not require trade-offs in making certain approximations. Therefore, this is the method we choose in doing our simulations.

Once we have ring modulation, we may test out the method proposed by Eric Cornell [Cornell 1992] for cooling the sum and difference modes separately. The recipe for cooling just the sum mode is the following:

- Adjust the DC ring voltage such that the average of the axial frequencies of the ions is equal to the coil frequency.
- Modulate the ring at a frequency equal to half the difference of the axial frequencies of the ions.
- Apply a cooling drive with frequency equal to the coil frequency plus the average magnetron frequency of the ions.

This creates overlapping sidebands of the ions at the coil, and the cooling drive is directed at these sidebands.

The computer simulations have demonstrated that this method does work to cool the sum mode, but is not very effective in the presence of anharmonicity, as shown in Figure 3.9. This is because as the ions cool, their axial frequencies change, and as a result the cooling drive is no longer in resonance with the sidebands of the ions, and the rate of cooling drastically slows. In addition, because there is some “bleeding” of the drive directly to the ions, the difference mode cools some, so the net difference in the modes is smaller than we would like.

A possible solution is to sweep the frequency of cooling drive. As shown in Figure 3.10, this cools the common mode almost completely. While it does also substantially cool the difference mode, the difference between the net effect on the two modes is very pronounced, and by heating the ions a little before the cooling, it should be possible to achieve the desired configurations of ions to make useful measurements.

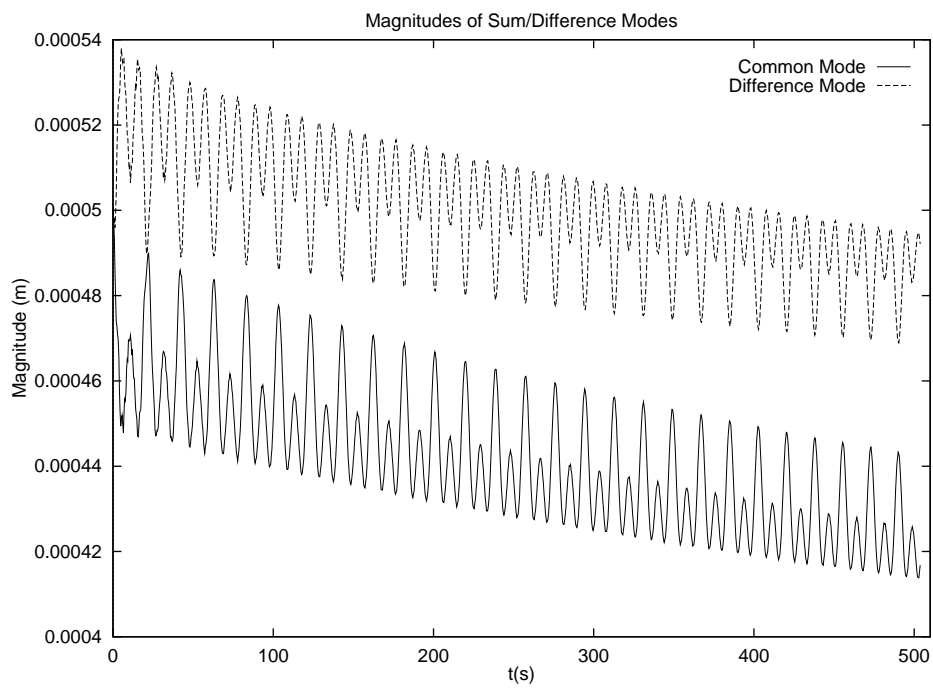
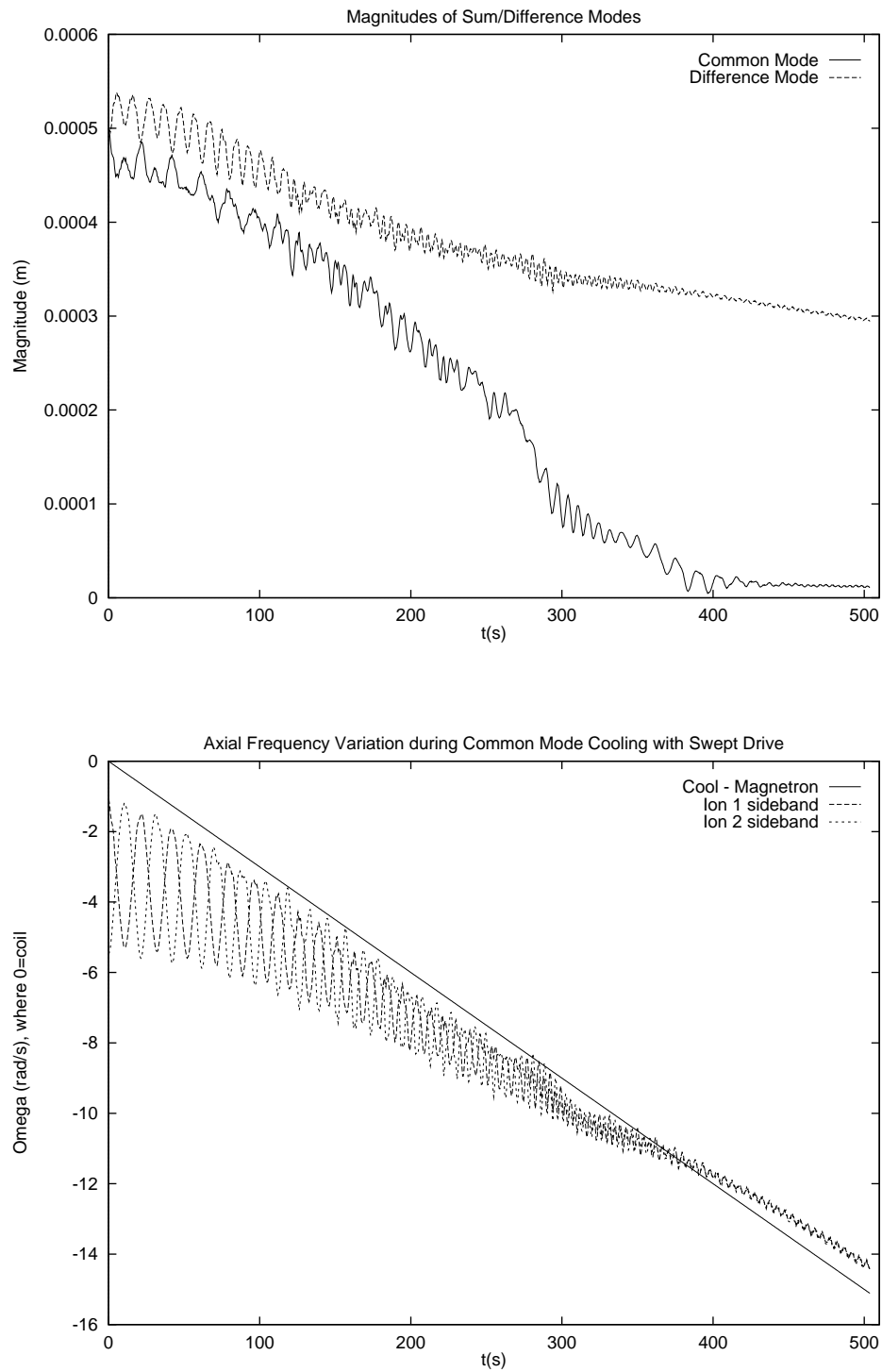


Figure 3.9: Two Ion Sum Mode Cooling with Anharmonicity

**Figure 3.10: Cooling Common Mode with Swept Drive**

Two Ion Implementation

As was mentioned in Section 2.3, one of the advantages of the simultaneous trapping scheme is that there is little hardware required beyond that needed for the previous, single-ion scheme. Here we describe the electronics that have been built to support the new configuration, and present some results of the two-ion experiments.

4.1 Electronics

The primary new piece of hardware necessary to implement the two-ion scheme consists of electronics to generate the necessary excitation and coupling pulses. Three mixers and a filter are used to generate signals at the appropriate frequencies from the output of signal generators running at $\overline{\omega}'_c$ and $\frac{\Delta\omega'_c}{2}$ (the average and half difference trap cyclotron frequencies) and $\overline{\omega}_\pi$ and $\frac{\Delta\omega_\pi}{2}$ (the average and half difference axial-cyclotron coupling frequencies). Thus, we generate signals at $\omega_{c'_1}$ and $\omega_{c'_2}$ to excite the ions, and ω_{π_1} and ω_{π_2} to couple the cyclotron modes to the axial. In addition, there is a signal generated at

$$\frac{\Delta\omega'_c}{2} - \frac{\Delta\omega_\pi}{2} = \frac{\Delta\omega_z}{2},$$

which is used to modulate the ring. As described in Section 3.5, this is precisely the frequency of modulation necessary to create sidebands of both ions at the frequency of the detector. A schematic of the electronics is shown in Figure 4.1.

4.2 Two-Ion Procedures and Results

With the above electronics in place, we are indeed able to make and see two ions in the trap at the same time. Figure 4.2 shows the power spectrum of a signal produced by exciting (through a cyclotron pulse followed by a coupling pulse) both ions (one N_2 and one CO) at the same time, and watching their sidebands. The frequency of modulation has been set such that these sidebands very nearly overlap, and we see two peaks corresponding to the signals of the two ions.

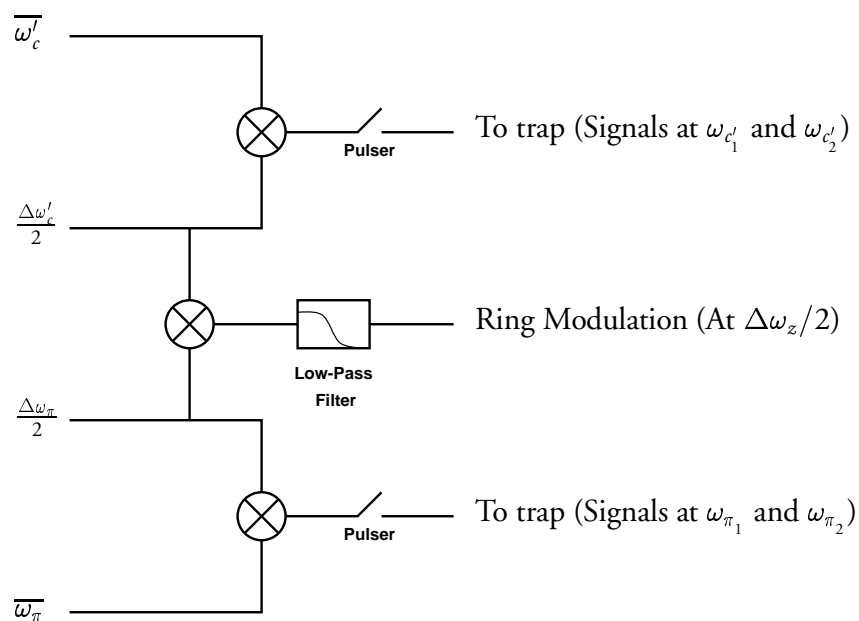


Figure 4.1: Schematic of Two-Ion Control Circuitry

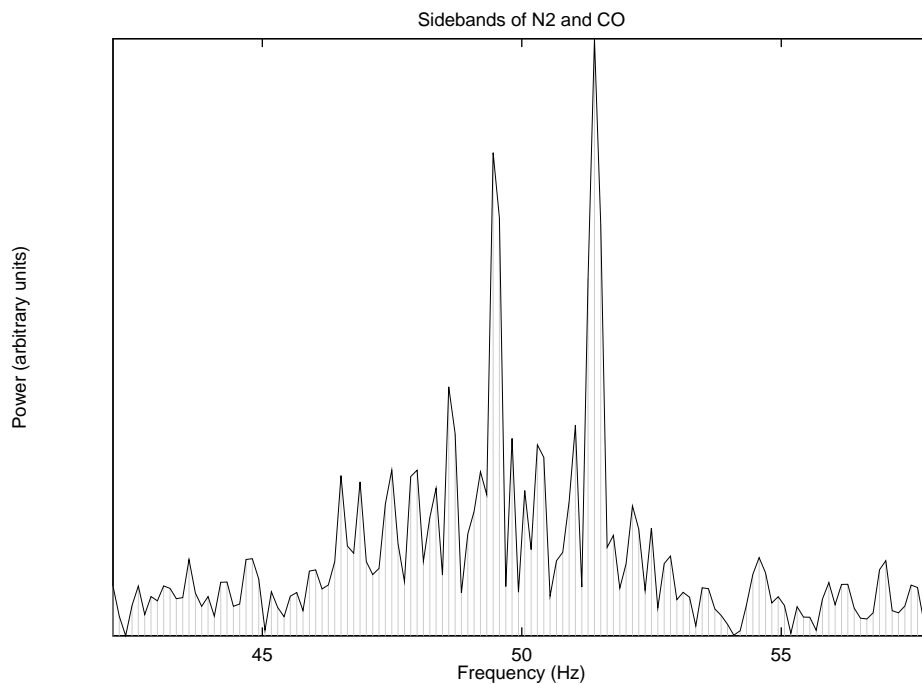


Figure 4.2: Power spectrum of sidebands of simultaneously trapped ions

We are also able to see the results of doing the pulse-and-phase procedure on both ions simultaneously. Figure 4.3 shows the phase accumulated in the cyclotron motion, for several runs using the same delay between the excitation and coupling pulses. As can be seen, the variation in the phases is quite large, but the variation in the difference in the phases accumulated by the two ions is much smaller. Thus, we see the realization of the predicted benefit of the two-ion scheme, as the errors in the individual ions' phases caused by field fluctuations cancel out to a large extent.

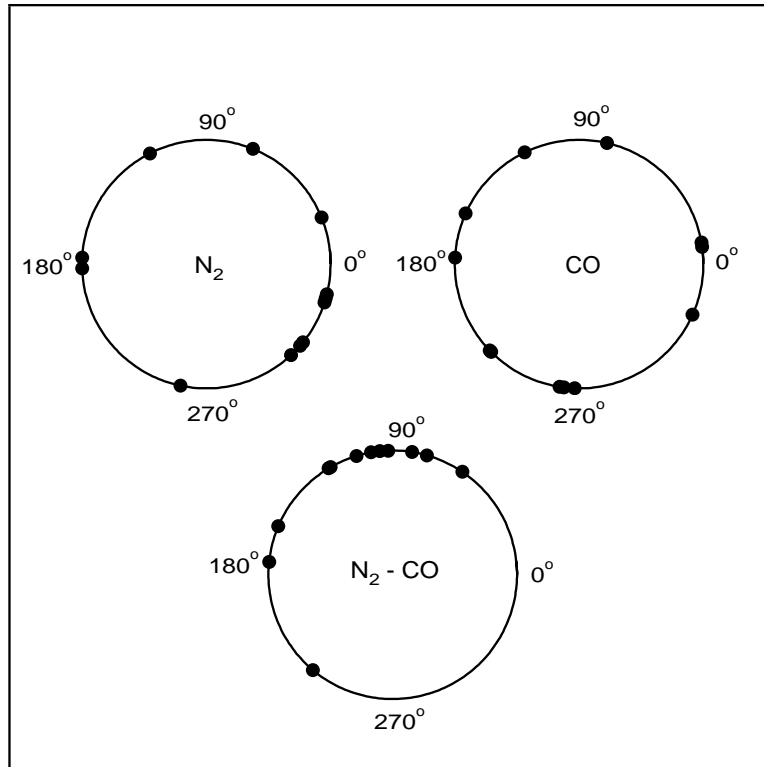


Figure 4.3: Accumulated cyclotron phases

It is of some concern that the variation in the phases is as large as it is. These data were taken during the daytime, so we expected a fair degree of magnetic field noise. Nevertheless, even the fluctuation in the difference of the phases is so large that we may begin to lose track of multiples of 360° in unrolling the phases. This phase variation is probably caused by the difficulty encountered in cooling the common mode motion of the ions, since the presence of both common and difference modes causes axial frequencies to vary with time.

Common mode cooling has proven to be fairly difficult in practice, since (as can be seen from Figure 3.10) the rate of this cooling is fairly slow if the initial common mode amplitude is large. With our present anharmonicity, as well as the large common mode amplitudes commonly produced by the process of making two ions, it seems that the proposed scheme for common mode cooling is not very effective.

A possible alternative to the common mode scheme is to measure the position of the ions inside the trap, and directly apply a magnetron excitation pulse of the right strength to bring the ions into the desired configuration. If this can be done with sufficient accuracy to bring the ions “reasonably close” to a pure difference mode, the remaining portion of common mode motion should cool much more effectively with the scheme of Section 3.5.

Even when we succeed in cooling the common mode motion, it is possible that we may still see large variations in the cyclotron phases, because of the influence of one ion on the other. We may then be forced to abandon our original plan of simultaneously applying the π -pulses to both ions, and pulse the ions separately. This is not a serious limitation, since throughout the majority of the time that phase is accumulating in the cyclotron mode, the ions will still be evolving simultaneously, and switching the order in which the ions are π -pulsed in successive runs should cause systematic errors to cancel out.

Appendix: Code Listing

```
/* trap.c: Simulate ions in trap */
/* Note: everything is in SI units */
#include <stdio.h>
extern double sqrt(double);
extern double sin(double);
extern double cos(double);
extern void rk();
#define SQR(x) ((x)*(x))
#define CUBE(x) ((x)*(x)*(x))
#define DIST(a1,b1,a2,b2) (sqrt(SQR((a1)-(a2))+SQR((b1)-(b2))))
#define ABS(x) ((x)<0?-(x):(x))
#define PI 3.141592653589793238

/* Parameters of the trap */
#define mr 1.0004 /* m1/m2 */
#define qbm 8.225e8 /* qB/(1 amu) */
#define m1 (28*mr) /* Atomic Mass 1 */
#define m2 28 /* Atomic Mass 2 */
#define kqm .1389 /* k_Coulomb*q^2/(1 amu) */
#define d 5.49e-3 /* Trap size */
#define wzo10 (160000*2*PI/(1+(mr-1)/2))
#define wzo20 (160000*2*PI) /* Axial freq at center of trap at t=0 */
#define wzo1 (wzo10*ac)
#define wzo2 (wzo20*ac) /* Axial freq at center of trap */
double vguard=.01; /* Guard ring voltage */
#define c4 (-6.3e-3*vguard) /* Anharmonicity */
double c6=4e-4; /* Anharmonicity */
double bigomega=1; /* Cooling constant */

/* Misc variables to keep track of the simulation */
int printevery=1, /* How often do we print the position */
    runfor=1000, /* How long the simulation runs */
    t=0; /* Counter of the number of steps run so far */
double dt=0.1; /* Length in seconds of time step */

/* The ions */
double x1=0, y1=0, x2=0, y2=0;
/* Axial (complex) amplitude – Real and imaginary parts */
double z1r=0, z1i=0, z2r=0, z2i=0;
```

```

/* The coil */
double er=0, ei=0;

/* Rotating reference frame */
double wrot,addrot=0;

/* Derived quantities */
#define wo1 (qbm/m1) /* Free space cyc 1 */
#define wo2 (qbm/m2) /* Free space cyc 2 */
#define ww1 (sqrt(SQR(wo1)-2*SQR(wzo1))) /* trap cyc - mag */
#define ww2 (sqrt(SQR(wo2)-2*SQR(wzo2)))
#define wm1 ((wo1-ww1)/2.) /* Mag freq 1 */
#define wm2 ((wo2-ww2)/2.) /* Mag 2 */
#define rdist2(x,y) ((SQR(x)+SQR(y))/SQR(d)) /* r^2/d^2 */

#define csign 1 /* 1=cool ; -1=heat */
#define wcool0 ((2.*wrot+wzo10+wzo20)/2.+0.0) /* Cooling drive */
#define wcool (wcool0-0.00*t*dt)
#define wd (wcool-csign*wrot)
#define delta1 (wz1-wd)
#define delta2 (wz2-wd)
#define wcoil (wcool0-csign*wrot)
#define deltae (wcoil-wd)
#define mcoil 1.8e-26 /* The "effective mass" of the coil */
#define gamma 50 /* Coil width */
#define q 1.602e-19
#define amu 1.6605e-27
/* Ring modulation */
#define wmod (ABS(wzo10-wzo20)/2.-0.0)
#define modindex 0.5 /* Modulation Index */
#define modamp (modindex*wmod/wcool0) /* frac change in wz */
int N; /* # of dt's per wmod period */
double *actable; /* Table of precomputed sines */
#define ac (actable[t%N]) /* Effect of Ring modulation */

#define dw1 (wm1-wrot) /* Difference between mag freq and rotation */
#define dw2 (wm2-wrot)
#define z21 (SQR(z1r)+SQR(z1i)) /* Square of axial amplitude */
#define z22 (SQR(z2r)+SQR(z2i))
#define sep (DIST(x1,y1,x2,y2)) /* Separation of the ions */
#define sep2 (SQR((x1)-(x2))+SQR((y1)-(y2))) /* Square of separation */
/* Correction to axial freq due to other ion */
#define wz1c (1.-kqm/m1/SQR(wzo1)/2./CUBE(sep)*(1-(9./8*z21/sep2+9./8*z22/sep2)/
(1+9./8*z21/sep2+9./8*z22/sep2)))
#define wz2c (1.-kqm/m2/SQR(wzo2)/2./CUBE(sep)*(1-(9./8*z22/sep2+9./8*z21/sep2)/
(1+9./8*z22/sep2+9./8*z21/sep2)))
/* The actual axial frequency - includes 1st order anharmonicity corrections */
#define wz1 (wzo1*wz1c*sqrt(1.-3.*c4*rdist2(x1,y1)+45./8*c6*SQR(rdist2(x1,y1))))
#define wz2 (wzo2*wz2c*sqrt(1.-3.*c4*rdist2(x2,y2)+45./8*c6*SQR(rdist2(x2,y2))))

```

```

/* The functions describing the derivatives of the variables */
inline double dx1(double x1,double y1,double x2,double y2,
                 double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
  return ( dw1*(-y1) /* i*dw1*x_complex */
          + kqm/m1/ww1/CUBE(sep)*(y2-y1)
          /* Electrostatic repulsion piece */
          /* = i*F_electric/m1/ww1 */
          *(1-3./2*(z21+z22)/sep2/(1+3./2*(z21+z22)/sep2))
          /* Correct for ax ampl */
          + SQR(wzo1)/ww1*(-y1)*( /* Anharmonic electric field */
          +3./2.*z21/SQR(d)*c4 /* Effect of ax. ampl */
          -3./4.*c4*rdist2(x1,y1) /* c4 piece */
          +15./16.*c6*SQR(rdist2(x1,y1)) /* c6 piece */
          )
          +csign*bigomega/2.*z1i /* Cooling = -i*omega/2*z */
          );
}

inline double dy1(double x1,double y1,double x2,double y2,
                 double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
  return ( dw1*(x1) + kqm/m1/ww1/CUBE(sep)*(x1-x2)
          *(1-3./2*(z21+z22)/sep2/(1+3./2*(z21+z22)/sep2))
          +SQR(wzo1)/ww1*(x1)*(+3./2.*z21/SQR(d)*c4-3./4.*c4*rdist2(x1,y1)
          +15./16.*c6*SQR(rdist2(x1,y1))) -csign*bigomega/2.*z1r);
}

inline double dx2(double x1,double y1,double x2,double y2,
                 double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
  return ( dw2*(-y2) + kqm/m2/ww2/CUBE(sep)*(y1-y2)
          *(1-3./2*(z21+z22)/sep2/(1+3./2*(z21+z22)/sep2))
          +SQR(wzo2)/ww2*(-y2)*(+3./2.*z22/SQR(d)*c4-3./4.*c4*rdist2(x2,y2)
          +15./16.*c6*SQR(rdist2(x2,y2))) +csign*bigomega/2.*z2i);
}

inline double dy2(double x1,double y1,double x2,double y2,
                 double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
  return ( dw2*(x2) + kqm/m2/ww2/CUBE(sep)*(x2-x1)
          *(1-3./2*(z21+z22)/sep2/(1+3./2*(z21+z22)/sep2))
          +SQR(wzo2)/ww2*(x2)*(+3./2.*z22/SQR(d)*c4-3./4.*c4*rdist2(x2,y2)
          +15./16.*c6*SQR(rdist2(x2,y2))) -csign*bigomega/2.*z2r );
}

inline double dz1r(double x1,double y1,double x2,double y2,
                 double z1r,double z1i,double z2r,double z2i,double er,double ei)

```

```

{
return ( -delta1*z1i          /* i*delta*z */
        + (q/2./m1/amu/wd)*ei /* -i*stuff*E */
        + bigomega/2.*y1     /* -i*omega/2*x */
        );
}
140

inline double dz1i(double x1,double y1,double x2,double y2,
                  double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
return ( delta1*z1r-(q/2./m1/amu/wd)*er-bigomega/2.*x1 );
}

150

inline double dz2r(double x1,double y1,double x2,double y2,
                  double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
return ( -delta2*z2i+(q/2./m2/amu/wd)*ei+bigomega/2.*y2 );
}

inline double dz2i(double x1,double y1,double x2,double y2,
                  double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
return ( delta2*z2r-(q/2./m2/amu/wd)*er-bigomega/2.*x2 );
}
160

inline double der(double x1,double y1,double x2,double y2,
                  double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
return ( -deltae*ei-gmma/2.*er          /* (i delta_e - gamma/2) e */
        +(q/2./mcoil/wd)*(z1i+z2i)     /* (stuff)*-i(z1+z2) */
        );
}
170

inline double dei(double x1,double y1,double x2,double y2,
                  double z1r,double z1i,double z2r,double z2i,double er,double ei)
{
return (deltae*er-gmma/2.*ei-(q/2./mcoil/wd)*(z1r+z2r)
        );
}

main(int argc, char *argv[])
{
180

/* Get options and parameters */
extern char *optarg; extern int optind; char c;
while ((c = getopt(argc, argv, "p:r:t:v:s:o:R:")) != EOF)
switch (c) {

```

```

        case 'p': printevery=atoi(optarg); break;
        case 'r': runfor=atoi(optarg); break;
        case 't': sscanf(optarg, "%lf", &dt); break;
        case 'v': sscanf(optarg, "%lf", &vguard); break;
        case 's': sscanf(optarg, "%lf", &c6); break;
        case 'o': sscanf(optarg, "%lf", &bigomega); break;
        case 'R': sscanf(optarg, "%lf", &addrot); break;
    }
    switch (argc-optind) {
        case 4: sscanf(argv[optind+3], "%lf", &y2);
        case 3: sscanf(argv[optind+2], "%lf", &x2);
        case 2: sscanf(argv[optind+1], "%lf", &y1);
        case 1: sscanf(argv[optind ], "%lf", &x1);
    }

    /* Adjust the time step */
    N=(int)(2.*PI/wmod/dt);
    dt=2*PI/wmod/N;
    /* Fill in actable */
    actable=(double *)malloc(N*sizeof(double));
    for(t=0;t<N;t++) {
        actable[t]=(1.+modamp*sin(wmod*dt*t));
    }
    t=0;
    /* Set freq of rotating frame */
    wrot=(wm1+wm2)/2+addrot;

#ifdef DEBUG
    printf("%f %f %f %f\n", wo1, wo2, wz1, wz2);
    printf("%f %f %f %f %f %d %d %f\n", wm1, wm2, ww1, ww2, dt, printevery, runfor, wrot);
    printf("%f %f %f %f\n", x1, y1, x2, y2);
    printf("%f %f %f\n", delta1, delta2, deltae);
    printf("%f %f\n", wcool, wmod);
    fflush((void *)0);
#endif

    do
    {
        if ((t % printevery) == 0) {
            if (t==50000) { z1r = 2e-3; }
            /* If it's time, print out current stuff */
            printf("%g %g %g %g %g %g %g %g %g %g\n", t*dt,
                x1, y1, x2, y2, z1r, z1i, z2r, z2i, wz1-wzo1, wz2-wzo2);
        }
        /* Update everything */
        rk(dt, &x1, dx1, &y1, dy1, &x2, dx2, &y2, dy2,
            &z1r, dz1r, &z1i, dz1i, &z2r, dz2r, &z2i, dz2i,
            &er, der, &ei, dei);
        t++;

```

```

    } while (t < runfor);
}



---




---


/* rk.c: Solve DEQ by RK method */

void rk(double dt,
        double* X1, double (*dx1)(), double* Y1, double (*dy1)(),
        double* X2, double (*dx2)(), double* Y2, double (*dy2)(),
        double *Z1R, double (*dz1r)(), double *Z1I, double (*dz1i)(),
        double *Z2R, double (*dz2r)(), double *Z2I, double (*dz2i)(),
        double *ER, double (*der)(), double *EI, double (*dei)() )

#define x1 (*X1)
#define y1 (*Y1)
#define x2 (*X2)
#define y2 (*Y2)
#define z1r (*Z1R)
#define z1i (*Z1I)
#define z2r (*Z2R)
#define z2i (*Z2I)
#define er (*ER)
#define ei (*EI)

{
    double hdt=dt/2.0,
    ax1,ay1,ax2,ay2,az1r,az1i,az2r,az2i,aer,aei,
    bx1,by1,bx2,by2,bz1r,bz1i,bz2r,bz2i,ber,bei,
    cx1,cy1,cx2,cy2,cz1r,cz1i,cz2r,cz2i,cer,cei,
    ex1,ey1,ex2,ey2,ez1r,ez1i,ez2r,ez2i,eer,eei,
    fx1,fy1,fx2,fy2,fz1r,fz1i,fz2r,fz2i,fer,fei;

    ax1=(*dx1)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    ay1=(*dy1)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    ax2=(*dx2)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    ay2=(*dy2)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    az1r=(*dz1r)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    az1i=(*dz1i)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    az2r=(*dz2r)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    az2i=(*dz2i)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    aer=(*der)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);
    aei=(*dei)(x1,y1,x2,y2,z1r,z1i,z2r,z2i,er,ei);

    bx1=(*dx1)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
              z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
              er+hdt*aer,ei+hdt*aei);
    by1=(*dy1)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
              z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
              er+hdt*aer,ei+hdt*aei);
    bx2=(*dx2)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,

```

```

        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);
by2>(*dy2)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);
bz1r>(*dz1r)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);
bz1i>(*dz1i)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);
bz2r>(*dz2r)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);
bz2i>(*dz2i)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);
ber>(*der)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);
bei>(*dei)(x1+hdt*ax1,y1+hdt*ay1,x2+hdt*ax2,y2+hdt*ay2,
        z1r+hdt*az1r,z1i+hdt*az1i, z2r+hdt*az2r,z2i+hdt*az2i,
        er+hdt*er,ei+hdt*eci);

cx1>(*dx1)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cy1>(*dy1)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cx2>(*dx2)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cy2>(*dy2)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cz1r>(*dz1r)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cz1i>(*dz1i)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cz2r>(*dz2r)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cz2i>(*dz2i)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*er,ei+hdt*eci);
cer>(*der)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,

```

```

        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*ber,ei+hdt*bei);
cei>(*dei)(x1+hdt*bx1,y1+hdt*by1,x2+hdt*bx2,y2+hdt*by2,
        z1r+hdt*bz1r,z1i+hdt*bz1i, z2r+hdt*bz2r,z2i+hdt*bz2i,
        er+hdt*ber,ei+hdt*bei);
100
ex1>(*dx1)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
ey1>(*dy1)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
ex2>(*dx2)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
ey2>(*dy2)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
110
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
ez1r>(*dz1r)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
ez1i>(*dz1i)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
ez2r>(*dz2r)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
120
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
ez2i>(*dz2i)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
eer>(*der)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
        er+dt*cer,ei+dt*cei);
eei>(*dei)(x1+dt*cx1,y1+dt*cy1,x2+dt*cx2,y2+dt*cy2,
        z1r+dt*cz1r,z1i+dt*cz1i, z2r+dt*cz2r,z2i+dt*cz2i,
130
        er+dt*cer,ei+dt*cei);

fx1=(ax1+bx1+bx1+cx1+cx1+ex1)/6.0;
fy1=(ay1+by1+by1+cy1+cy1+ey1)/6.0;
fx2=(ax2+bx2+bx2+cx2+cx2+ex2)/6.0;
fy2=(ay2+by2+by2+cy2+cy2+ey2)/6.0;
fz1r=(az1r+bz1r+bz1r+cz1r+cz1r+ez1r)/6.0;
fz1i=(az1i+bz1i+bz1i+cz1i+cz1i+ez1i)/6.0;
fz2r=(az2r+bz2r+bz2r+cz2r+cz2r+ez2r)/6.0;
fz2i=(az2i+bz2i+bz2i+cz2i+cz2i+ez2i)/6.0;
fer=(aer+ber+ber+cer+cer+eer)/6.0;
140
fe1=(aei+bei+bei+cei+cei+eei)/6.0;

x1+=dt*fx1;

```

```
    y1+=dt*fy1;
    x2+=dt*fx2;
    y2+=dt*fy2;
    z1r+=dt*fz1r;
    z1i+=dt*fz1i;
    z2r+=dt*fz2r;
    z2i+=dt*fz2i;
    er+=dt*fer;
    ei+=dt*fei;
}
```

150

References

- [Brown & Gabrielse] Brown, L. S. and Gabrielse, G. *Rev. Mod. Phys.* 58:233, 1986.
- [Boyce] Boyce, K. R. *Improved Single Ion Cyclotron Resonance Mass Spectroscopy*, Ph. D. Dissertation. MIT, 1992.
- [Cornell 1990] Cornell, E. A. *Mass Spectroscopy Using Single Ion Cyclotron Resonance*, Ph. D. Dissertation. MIT, 1990.
- [Cornell 1992] Cornell, E. A., Boyce, K., Fyngenson, D., and Pritchard, D. E. “Two Ions in a Penning Trap: Implications for Precision Mass Spectroscopy,” *Phys. Rev. A*, Vol. 45, No. 5, Mar. 1, 1992.
- [DiFilippo 1992] DiFilippo, F., Natarjan, V., Boyce, K., and Pritchard, D. E. “Classical Amplitude Squeezing for Precision Measurements,” *Phys. Rev. Lett.*, Vol. 68, pp. 2859–2862. 1992.
- [DiFilippo 1994] DiFilippo, F., Natarjan, V., Boyce, K., and Pritchard, D. E. “Accurate Masses for Fundamental Metrology,” *Phys. Rev. Lett.*, Vol. 73, pp. 1481–1484, 1994.
- [Kuchnir] Kuchnir, D. L. *Trapping and Detecting Two Different Single Ions at Once: A Step Towards Ultra-High-Precision Mass Comparison Measurements*, S. B. Dissertation, 1989.