

Stony Brook University



OFFICIAL COPY

The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.

© All Rights Reserved by Author.

Numerical Study of Periodic Adiabatic Rapid Passage Sequences

A Thesis Presented

by

He Zhang

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Master of Arts

in

Physics and Astronomy

Stony Brook University

February 2016

Stony Brook University

The Graduate School

He Zhang

We, the thesis committee for the above candidate for the Master of Arts degree, hereby recommend acceptance of this thesis.

Harold Metcalf – Thesis Advisor

Distinguished Teaching Professor, Department of Physics and Astronomy

Dominik Schneble – Chairperson of Defense

Associate Professor, Department of Physics and Astronomy

Neelima Sehgal

Assistant Professor, Department of Physics and Astronomy

This thesis is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Thesis

Numerical Study of Periodic Adiabatic Rapid Passage Sequences

by

He Zhang

Master of Arts

in

Physics and Astronomy

Stony Brook University

2016

Laser cooling refers to the techniques that cool atomic and molecular samples down to near absolute zero through the interaction with laser fields. Atoms absorb a photon and the transferred momentum generates force to lower the atom's velocity. The Doppler shift is associated with the motion of the atom, in which a moving atom always observes a slightly different frequency than the laser at rest.

Based on the two-level system theory, this thesis studied the process of multiple sequences of adiabatic rapid passage by simulating the process of the interaction between optical forces and helium atoms. By studying the force map in parameter space, trying to understand the relationship between the average force and

relevant laser parameters Ω_0 and δ_0 in the Rabi frequency. Furthermore, we studied the relationship between the average force and velocity by comparing the dragged atom model and moving atom model. We also studied the process of multiple sequences of adiabatic rapid passage by changing spontaneous emission, sweep range, pulse number for the interaction, exploring the distance between peaks that appear in the velocity distribution. The observational results will be presented in the thesis as well as the explanation.

For my family and friends

Contents

List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Overview	1
1.2 Two-Level Model	2
1.2.1 Rabi Oscillations	4
1.2.2 Optical Bloch Sphere Projection	6
1.3 Optical Forces and Laser Cooling	11
1.3.1 Radiative Force	11
1.3.2 Stimulated Emission and Dipole Force	16
1.3.3 Bichromatic Force	16
2 Theory of Adiabatic Rapid Passage	18
2.1 Adiabatic Rapid Passage	18
2.2 ARP Force	24
3 Numerical Study and Result	26

3.1	ARP Force Simulation	26
3.1.1	Force Maps	28
3.1.2	Force vs Velocity	30
3.2	Parameter Effects	33
3.2.1	Spontaneous Emission	33
3.2.2	Sweeping Direction	36
3.2.3	Number of Pulses	38
3.2.4	Distribution of Peak Distance	40
3.2.5	Pulse distortion	41
3.2.6	Sensitivity of parameters δ_0 & Ω_0	43
4	Conclusions	45
	Bibliography	47
A	Python Code for Position-velocity Plot Numerical Calculations	48
B	Python Code for Force Map Numerical Calculations	55
C	Python Code for Parallel Calculations	63
D	R Code for Data Visualization	71

List of Tables

3.1	Fit Values	40
-----	----------------------	----

List of Figures

1.1	Two level model	3
1.2	Rabi Oscillations	6
1.3	Bloch Sphere Representation	9
1.4	Bloch sphere projections of the Bloch and torque vectors with torque vector	10
1.5	Spontaneous emission	12
1.6	Velocity Dependence of Optical damping force in one dimensional picture . .	15
1.7	The bichromatic optical force with two frequencies ω_r and ω_b equally detuned above and below atomic resonance.	16
1.8	Bichromatic force profile	17
2.1	Eigenenergy of dressed picture	20
2.2	Bloch Sphere Trajectories for ARP	22
2.3	Timing Sequence for ARP	24
3.1	The contour plot for the average force over different velocity	29
3.2	Trajectory of the atom movement	30
3.3	The force vs velocity plot of dragged atom and moving atom	33
3.4	Longitudinal Velocity Distribution	35
3.5	The experiment result comparison	36

3.6	Longitudinal Velocity Distribution	37
3.7	Calculated velocity dependence of the ARP force	38
3.8	Force changing based on different pulses number	39
3.9	Peak splitting fitting	41
3.10	Distortion of the pulse	42
3.11	Force vs velocity without distortion	43
3.12	Resolution detection	44

Chapter 1

Introduction

In this chapter brief introductions to laser cooling and Rabi oscillations are given, as well as an optical forces analysis. Chapter 2 is the description of Adiabatic Rapid Passage theory. Chapter 3 describes the numerical simulations of the ARP force interpretation and simulation results. Chapter 4 has a conclusion of these topics.

1.1 Overview

Laser cooling refers to the techniques that cool atomic and molecular samples down to near absolute zero through the interaction with laser fields. The key idea is that atoms and molecules are constantly in motion, so cooling them is indeed to reduce their motion or speed. Atoms absorb light and the transferred momentum generates force to lower the atom's velocity. The atom emits spontaneous emission with average velocity change of zero since the emission direction is random.

The Doppler shift is associated with the motion of the atom. A moving atom always observes a slightly different frequency than the laser. When the atom is in two laser beams,

it experiences that the laser light Doppler shifted to a slightly higher frequency toward which it moving and likewise, at a slightly lower frequency from which it moving away.

Laser cooling provides a starting point for a number of experiments. Bose-Einstein condensation in quantum information and computing research is one of them, that is a state of matter of a dilute gas of bosons cooled to temperatures very close to absolute zero. Laser cooling also contributes to high-resolution spectroscopic measurements, like frequency standards in optical clocks based on ultracold ions or atoms. Ultraprecise measurement of gravitational fields based on the Doppler shifts of free-falling cooled atoms is another application, and lithography with cold atomic beams to form very accurately controlled structures.

1.2 Two-Level Model

Atomic gases at low density show sharp energy eigenspectra. Usually there are infinitely many energy eigenstates, but in quantum optics it is enough to take into account only two energy levels of an atom, in the interaction with the laser light, which is related to the laser transition. This simplified model is called the two-level atom [1].

To describe the semi-classical behaviour of a quantum mechanical two-level system (shown in Fig. 1.1) driven by resonant radiation, first we need to make some definitions.

- $|g\rangle$ and $|e\rangle$ form the ground and excited state for the system
- The detuning is $\delta = \omega_l - \omega_0$, here ω_l is laser frequency, ω_0 atomic resonance frequency

The corresponding energies of the eigenstates are E_g and E_e , satisfying the bare Hamiltonian H_0 :

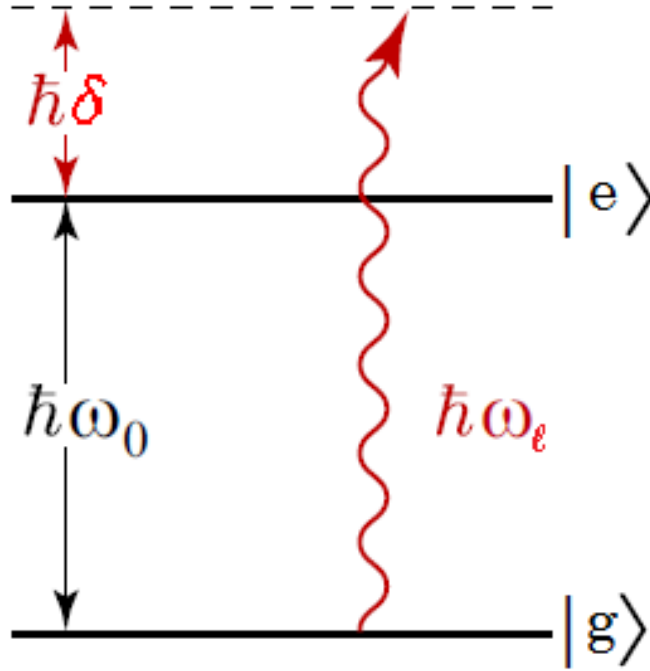


Figure 1.1: Two level model[1], $|g\rangle$ and $|e\rangle$ represent the ground and excited state for the system[1]

$$H_0|g\rangle = E_g|g\rangle \quad (1.1)$$

$$H_0|e\rangle = E_e|e\rangle \quad (1.2)$$

where $|g\rangle$ and $|e\rangle$ in the presence of a monochromatic light field are governed by the time-dependent Schrödinger equation:

$$\frac{\partial|\psi(t)\rangle}{\partial t} = -\frac{i}{\hbar} \sum_j c_j(t) e^{-iE_j t/\hbar} E_j |j\rangle \quad (1.3)$$

the probability of the state $|j\rangle$ satisfies $\sum_j |c_j(t)|^2 = 1$ all the time, so the two-level quantum

state can be expressed as $|\psi\rangle = c_g|g\rangle + c_e|e\rangle$, here c_g and c_e are complex state amplitudes, $|c_g|^2 + |c_e|^2 = 1$. The state can be expressed as vector

$$\psi = \begin{pmatrix} \langle g|\psi\rangle \\ \langle e|\psi\rangle \end{pmatrix} = \begin{pmatrix} c_g \\ c_e \end{pmatrix} \quad (1.4)$$

1.2.1 Rabi Oscillations

Now we consider the Rabi frequency, the frequency of oscillation for flopping between the states of the two-level system caused by the resonant light. The coefficients measure the population between of each level. The total Hamiltonian is the linear combination of bare Hamiltonian and interaction $H = H_0 + H_{int}$. The matrix corresponding to the Hamiltonian is [1]

$$H = \frac{\hbar}{2} \begin{pmatrix} 0 & \Omega(e^{i\omega t} + e^{-i\omega t}) \\ \Omega^*(e^{i\omega t} + e^{-i\omega t}) & -2\delta \end{pmatrix} \quad (1.5)$$

with the rotating wave approximation it becomes,

$$H = \frac{\hbar}{2} \begin{pmatrix} 0 & \Omega \\ \Omega^* & -2\delta \end{pmatrix} \quad (1.6)$$

Applied 1.4 and 2.12 to Schrödinger equation:

$$i\dot{c}_g = c_e(t) \frac{\Omega}{2} \quad (1.7a)$$

$$i\dot{c}_e = c_g(t) \frac{\Omega^*}{2} - c_e(\omega_l - \omega_0) \quad (1.7b)$$

$$c_g(0) = 1 \quad (1.7c)$$

$$c_e(0) = 0 \quad (1.7d)$$

The solution to 1.7 is

$$\begin{pmatrix} c_g(t) \\ c_e(t) \end{pmatrix} = \begin{pmatrix} \left(\cos \frac{\Omega' t}{2} + i \frac{\delta}{\Omega'} \sin \frac{\Omega' t}{2} \right) e^{i\delta t/2} \\ \left(i \frac{\Omega}{\Omega'} \sin \frac{\Omega' t}{2} \right) e^{i\delta t/2} \end{pmatrix} \quad (1.8)$$

here we have defined the generalized Rabi frequency $\Omega' \equiv (\Omega^2 + \delta^2)^{1/2}$. The corresponding evolution of the state populations, $\rho_{gg}(t) = |c_g(t)|^2$ and $\rho_{ee}(t) = |c_e(t)|^2$ are

$$\rho_{gg}(t) = \frac{1}{2} \left[1 + \left(\frac{\delta}{\Omega'} \right)^2 \right] + \frac{1}{2} \left(\frac{\Omega}{\Omega'} \right)^2 \cos \Omega' t \quad (1.9a)$$

$$\rho_{ee}(t) = \frac{1}{2} \left(\frac{\Omega}{\Omega'} \right)^2 [1 - \cos \Omega' t] \quad (1.9b)$$

From the above equations, the excited state population oscillates at an angular frequency of Ω' and has an amplitude of $(\Omega/\Omega')^2$. According to Eq. 1.9, when the detuning is not zero, the population oscillation can not reach the full oscillation. Fig. 1.2 shows how the probability of the excited state varies in time with various values of Ω and δ .

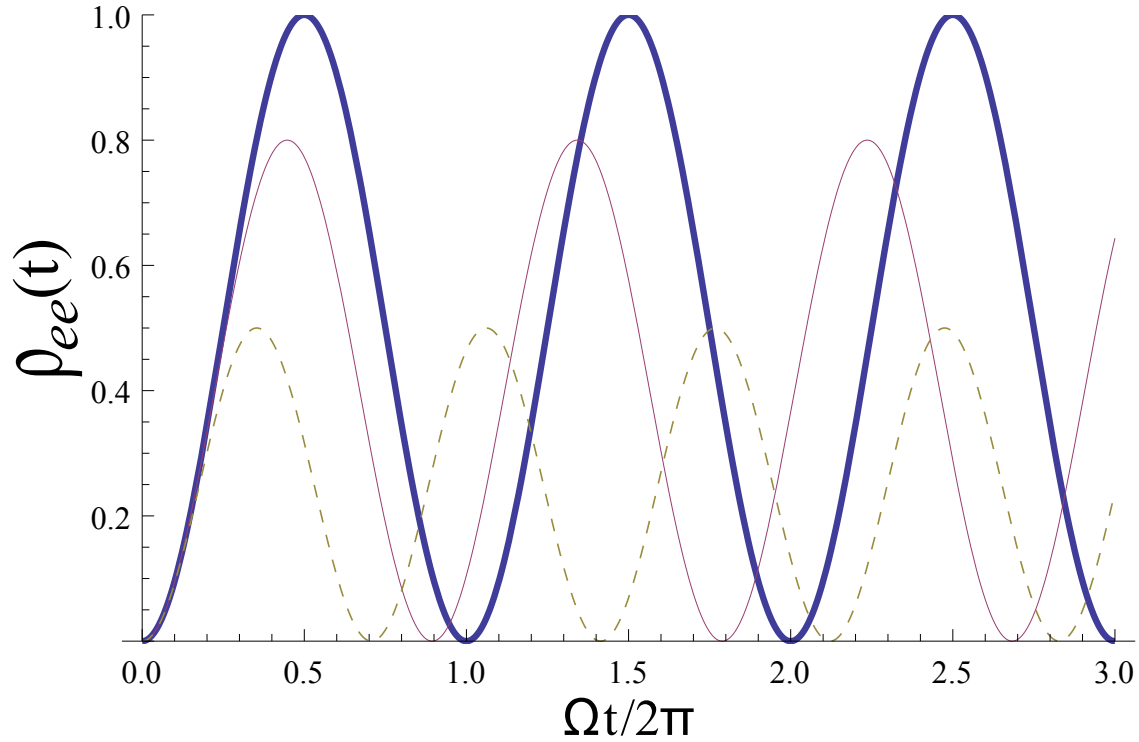


Figure 1.2: Oscillations of the excited state population for fixed Ω and varying δ . The thick, thin, and dashed lines correspond to $\delta = 0$, $\Omega/2$, and Ω respectively. The excited state population oscillates at an angular frequency of Ω' with an amplitude of $(\Omega/\Omega')^2$ [2].

1.2.2 Optical Bloch Sphere Projection

The Bloch sphere is a three dimensional projection of the atom interaction with laser light, geometrically representing the pure state space of a two-level quantum mechanical model. The pure state $|\psi\rangle$ of the two-level model can be represented as a superposition of the basis vectors $|g\rangle$ and $|e\rangle$. Furthermore the connection between the Bloch vector and wave function written in spherical coordinates is the following representation:

$$|\psi\rangle = \cos\left(\frac{\alpha}{2}\right) |g\rangle + e^{i\phi} \sin\left(\frac{\alpha}{2}\right) |e\rangle \quad (1.10)$$

where α is angle between z-axis and the Bloch vector. The phase of the wave function does not have physical meaning, so we are only left with three free parameters in the solutions to 1.8. The arbitrary state is given in Cartesian coordinate is

$$\mathbf{u} = 2\text{Re}(c_g \tilde{c}_e^*) \quad \mathbf{v} = 2\text{Im}(c_g \tilde{c}_e^*) \quad \mathbf{w} = |\tilde{c}_e|^2 - |\tilde{c}_g|^2 \quad (1.11)$$

where

$$\frac{dc_g}{dt} = -i \frac{\Omega^*}{2} \tilde{c}_e \quad (1.12)$$

$$\frac{d\tilde{c}_e}{dt} = -i \frac{\Omega}{2} c_g + i\delta \tilde{c}_e \quad (1.13)$$

Then we calculate time dependence of the parameters $\mathbf{u}, \mathbf{v}, \mathbf{w}$, and find

$$\frac{d\mathbf{u}}{dt} = \delta \mathbf{v} - \Omega_i \mathbf{w} \quad (1.14)$$

$$\frac{d\mathbf{v}}{dt} = \delta \mathbf{u} - \Omega_r \mathbf{w} \quad (1.15)$$

$$\frac{d\mathbf{w}}{dt} = \Omega_i \mathbf{u} + \Omega_r \mathbf{v} \quad (1.16)$$

here $\Omega = \Omega_r + i\Omega_i$ which is the Rabi frequency. The result is similar to vector cross product, and we define two vector: the torque vector $\mathbf{\Omega} = (\Omega_r, \Omega_i, -\delta)$ and $\mathbf{R} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$. The evolution for \mathbf{R} [3](shown in Fig. 1.3):

$$\frac{d\mathbf{R}}{dt} = \mathbf{\Omega} \times \mathbf{R} \quad (1.17)$$

Spontaneous emission is considered as inelastic collision where atom may change internal energy state. Such collision leads to dephasing or decreasing the coherence of interaction.

When we consider spontaneous emission, the Optical Bloch equations need to be modified to present the dynamic system [2].

$$\dot{u} = \Omega_i w - \delta v - (\gamma/2)u \quad (1.18a)$$

$$\dot{v} = -\Omega_r w + \delta u - (\gamma/2)v \quad (1.18b)$$

$$\dot{w} = \Omega_r v - \Omega_i u - \gamma(w + 1) \quad (1.18c)$$

here γ is the spontaneous emission rate for the excited state.

From eq. 1.17, we know the time derivative of \mathbf{R} is always perpendicular to \mathbf{R} , so the vector magnitude of \mathbf{R} is constant, so it moves along the surface of the Bloch sphere. If \mathbf{R} is in the equatorial plane with $w = 0$ and the mixed state refers to the eq. 1.10.

Fig. 1.3 shows the Bloch Sphere representation. The south pole of the Bloch sphere represents the ground state of the atom, the north pole corresponds to the excited state. All other points on the sphere represent various superposition states. Assume Bloch vector at the beginning is located at the south pole (ground state) with the initial condition $\mathbf{R}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = (0, 0, -1)$. Assume the torque vector $\mathbf{\Omega}$, choose $\delta = 0$. Because the time derivative of the Bloch vector is always perpendicular to torque vector and stays in the surface of Bloch sphere, The torque vector lies along the u axis and the Bloch vector follows a circular path in the v, w plane like Fig. 1.4 (a). Choose $\delta \neq 0$, the torque vector begin with south pole and rotate to north pole, the Bloch vector will always revolves round the torque vector until they reach north pole show in fig.1.4 (b).

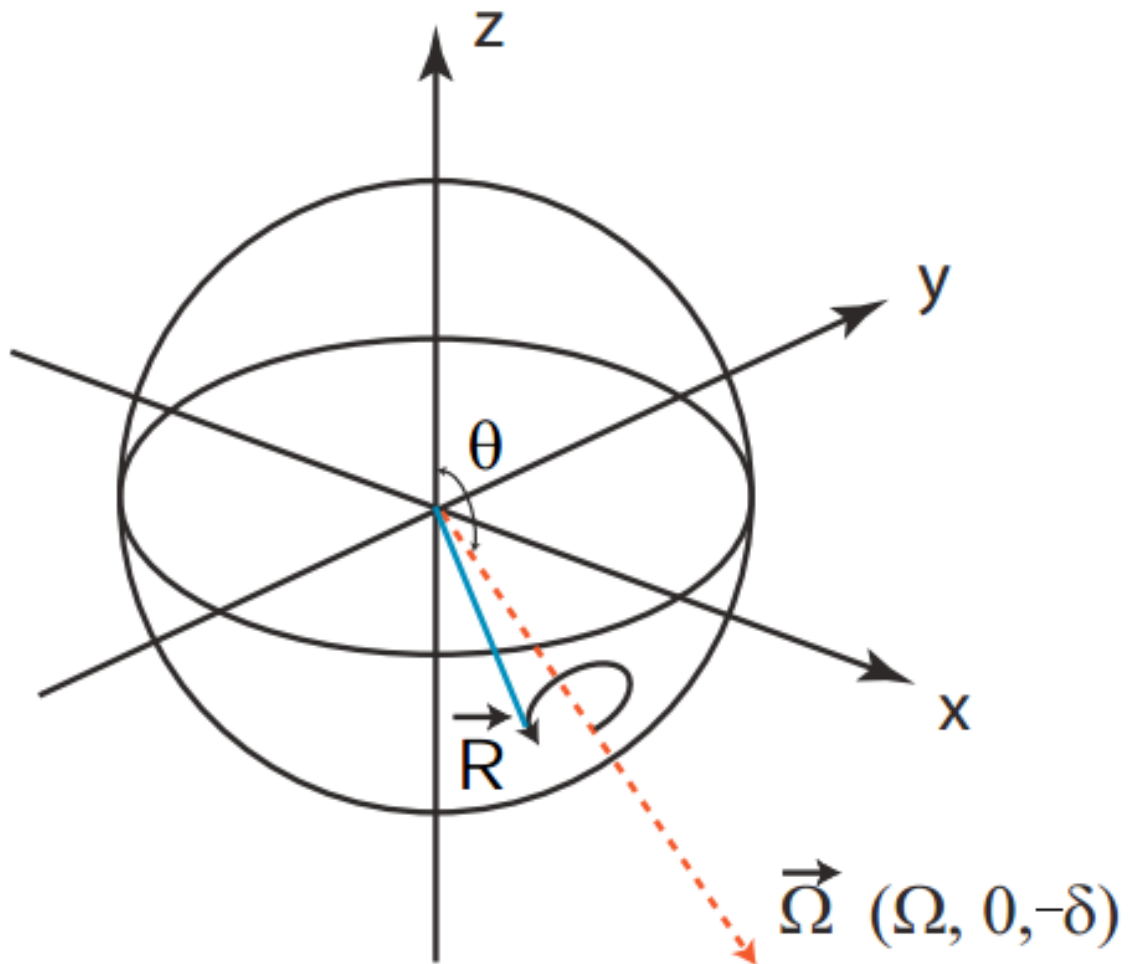


Figure 1.3: The Bloch sphere projections taken from[4]. The solid and dash line represent the Bloch vector and torque vector respectively.

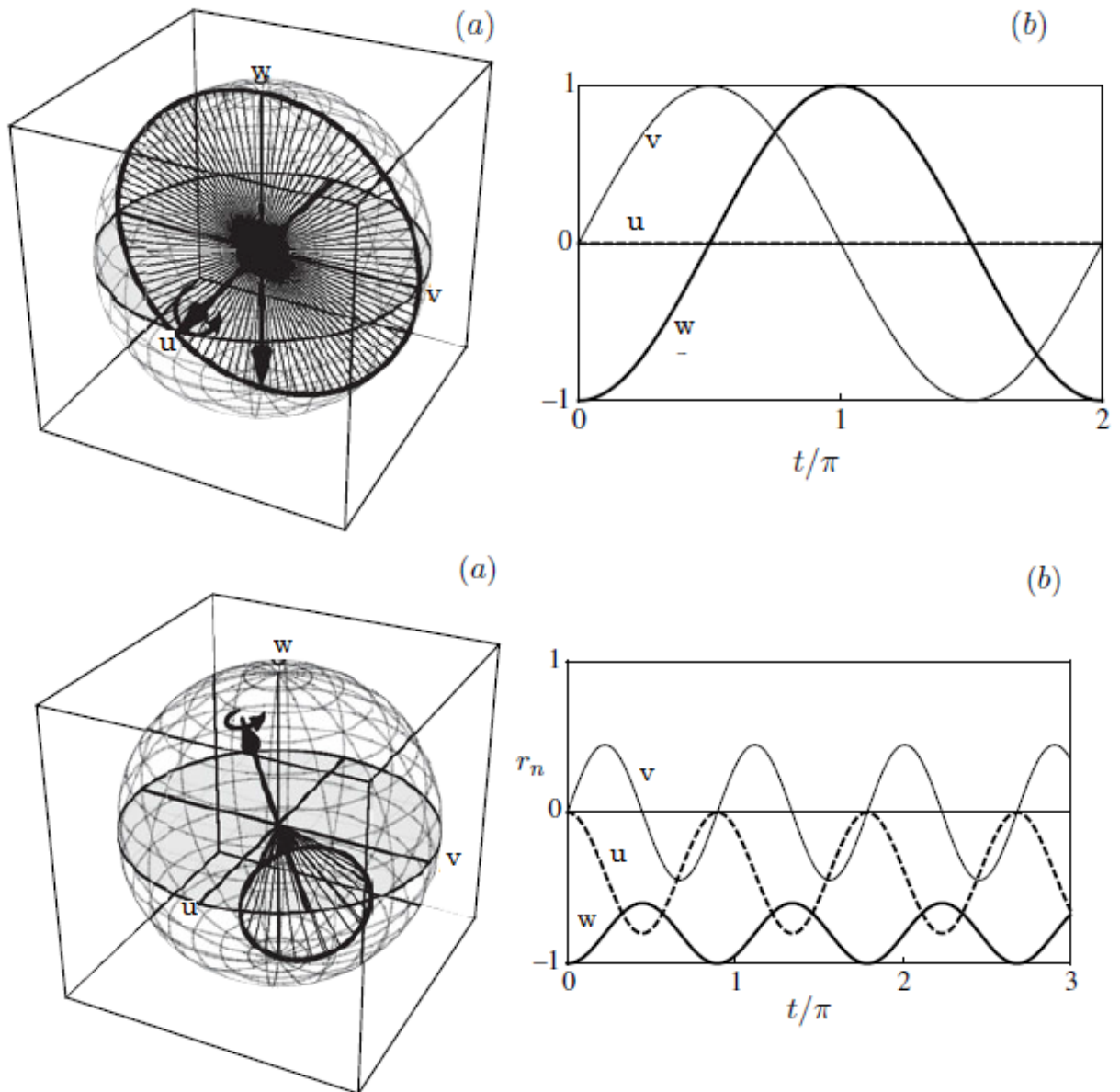


Figure 1.4: Bloch sphere projections of the Bloch and torque vectors with torque vector beginning on the south pole[5]. Fig. 1.4 (a) shows that when $\delta = 0$, the time derivative of the Bloch vector is always perpendicular to torque vector and stays in the surface of Bloch sphere, The torque vector lies along the u axis and the Bloch vector follows a circular path in the v, w plane. Fig.1.4 (b) shows that when $\delta \neq 0$, the torque vector begin with south pole and rotate to north pole, the Bloch vector will always revolves round the torque vector until they reach north pole.

1.3 Optical Forces and Laser Cooling

When the atom interacts with laser light with frequency close to the resonance frequency, it makes a transition from the ground state to the excited state, then subsequently comes back down in two ways: spontaneous emission or stimulated emission. When the intensity of absorbed light which left the atom still in excited state is very low, the condition is that the atom returns back to ground state results from spontaneous emission rather than stimulated emission. One photon comes out during this process, with momentum $\hbar k$ with random direction. Such momentum exchange is named the light pressure force. However, when the detuning is much larger than γ , which indicates spontaneous emission may be much less frequent than stimulated emission, the atom returns to the ground state more by stimulated emission. In this process, the exciting light and stimulated light have the same momentum so there is zero momentum exchange. But if there are beams with two different \mathbf{k} -vectors, absorbing from one beam followed by stimulated emission into the other would produce non-zero momentum exchange, which is named the dipole force. We particularly wish to point out that these two fundamental forces are quite different. Dipole forces can be made very large by high intensity light interaction with atom. By contrast, light pressure force can not be made strong. It is limited by the spontaneous emission rate [3].

1.3.1 Radiative Force

Spontaneous emission is a process that an atom in an excited state makes a transition to a lower energy state E_1 level with quanta emission of energy. For example, one atom interacting with the light source absorbs the light energy $\hbar\omega$ reach to energy E_2 level. It may spontaneously decay to a lower level with one photon emitted with energy $\hbar\omega$ emission

(shown in fig. 1.5).

$$E_2 - E_1 = \hbar\omega \quad (1.19)$$

here \hbar is the reduced Planck constant. The phase of the light in spontaneous emission is random as is the direction in which the light propagates. So the average overall force is zero.

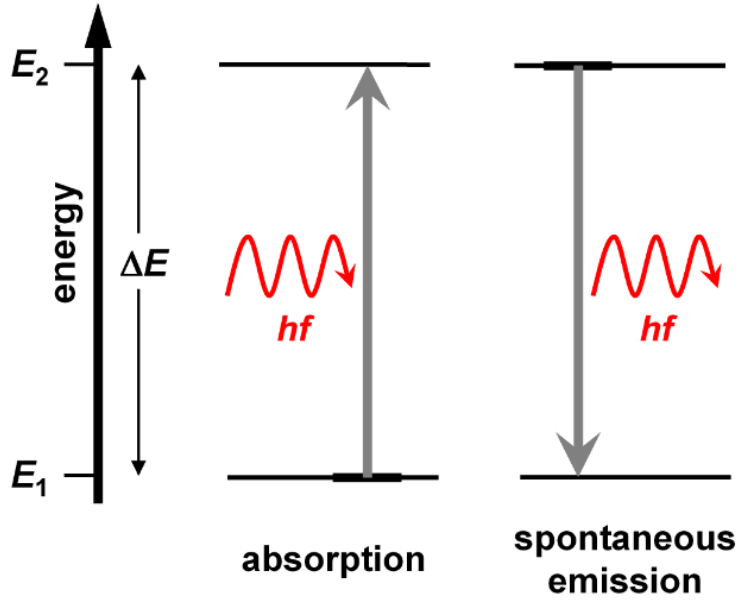


Figure 1.5: Spontaneous emission process[6] when one photon is absorbed the energy level goes to the excited state with energy E_2 , it may spontaneously decay to a lower lying level (the ground state) with energy E_1 , releasing the difference in energy between the two states as a photon.

The momentum changed is

$$\hbar k = \hbar\omega/c \quad (1.20)$$

The net force is proportional to the momentum absorbed and the scattering rate of the process

$$\mathbf{F} = \gamma_p \hbar \mathbf{k} \quad (1.21)$$

here $\gamma_p = \gamma\rho_{ee}$ is the scattering rate. From the semiclassical description of the light interaction, the force is defined as the expectation value of force operator \mathcal{F} , $F = \langle \mathcal{F} \rangle = d\langle p \rangle / dt$. Consider operator p can be replaced by $-i\hbar(\partial/\partial z)$. So the force is operator thus given by $\mathbf{F} = -\partial\mathcal{H}/\partial z$. [6] Using the dipole approximation, we have the result

$$\mathbf{F} = \hbar\left(\frac{\partial\Omega}{\partial z}\rho_{eg}^* + \frac{\partial\Omega^*}{\partial z}\rho_{eg}\right) \quad (1.22)$$

where ρ is the density matrix,

$$\rho = |\psi\rangle\langle\psi| = \begin{pmatrix} |c_g|^2 & c_g c_e^* \\ c_e c_g^* & |c_e|^2 \end{pmatrix} \quad (1.23)$$

Consider that the total scattering rate γ_p of light from laser field is

$$\gamma_p = \frac{s_0\gamma/2}{1 + s_0 + (2\delta/\gamma)^2} \quad (1.24)$$

Here s_0 is ratio of the laser field intensity to the saturation intensity defined by

$$s_0 \equiv 2\frac{|\Omega|^2}{\gamma^2} = \frac{I}{I_S} \quad (1.25)$$

with $I_S \equiv \pi\hbar c/(3\lambda^3\tau)$ (λ and τ are the wavelength and lifetime of the atomic transition respectively). Then we have the force from absorption followed by spontaneous emission,

$$F_{sp} = \frac{\hbar k s_0 \gamma / 2}{1 + s_0 + (2\delta/\gamma)^2} = \hbar k \gamma_p \quad (1.26)$$

For the previous case, we just consider the stationary atom, and now the Doppler shift will be considered. For the Doppler shift when the observer moving towards the source,

the frequency of wave increases. Especially when observer faces the wave, the time to reach the observer is slightly less; when the observer is moving along the direction of wave propagation, it will observe a little late arrival. Therefore, when considering one laser, assume the frequency shift of the light field is $\mathbf{k} \cdot \mathbf{v}$, and the effective detuning is $\delta - \mathbf{k} \cdot \mathbf{v}$. The corresponding force on a moving atom is

$$\mathbf{F}_{rad} = \hbar \mathbf{k} \frac{s_0 \gamma / 2}{1 + s_0 + \left(2(\delta - \mathbf{k} \cdot \mathbf{v}) / \gamma\right)^2} \quad (1.27)$$

In a common experimental setup, there are two incident light beams tuned below atomic resonance shining in opposite directions. The atom is slowed down for either direction of motion because both lasers can cause the atom to lose momentum in the same direction as the velocity.

$$\begin{aligned} \mathbf{F} &= \hbar \mathbf{k} \frac{\gamma s_0 / 2}{1 + s + (2(\delta - \mathbf{k} \cdot \mathbf{v})) / \gamma)^2} - \hbar \mathbf{k} \frac{\gamma s_0 / 2}{1 + s + (2(\delta + \mathbf{k} \cdot \mathbf{v})) / \gamma)^2} \\ &\approx \frac{8 \hbar k^2 \delta s_0 \mathbf{v}}{\gamma (1 + s_0 + (2\delta / \gamma)^2)^2} \equiv -\beta \mathbf{v} \end{aligned} \quad (1.28)$$

Here terms of the order $(kv/\gamma)^4$ and higher have been neglected in the approximation. In the second line, we may see a very interesting phenomenon called optical molasses, which means the Doppler shift induces the atom to see different detuning, so the atoms with different velocities will have different resonance frequency. In red detuned ($\delta < 0$) optical molasses, the total profile of force is shown in Fig. 1.6, atoms in the middle region with small range of velocity would be pushed into the middle where the force is close to zero.

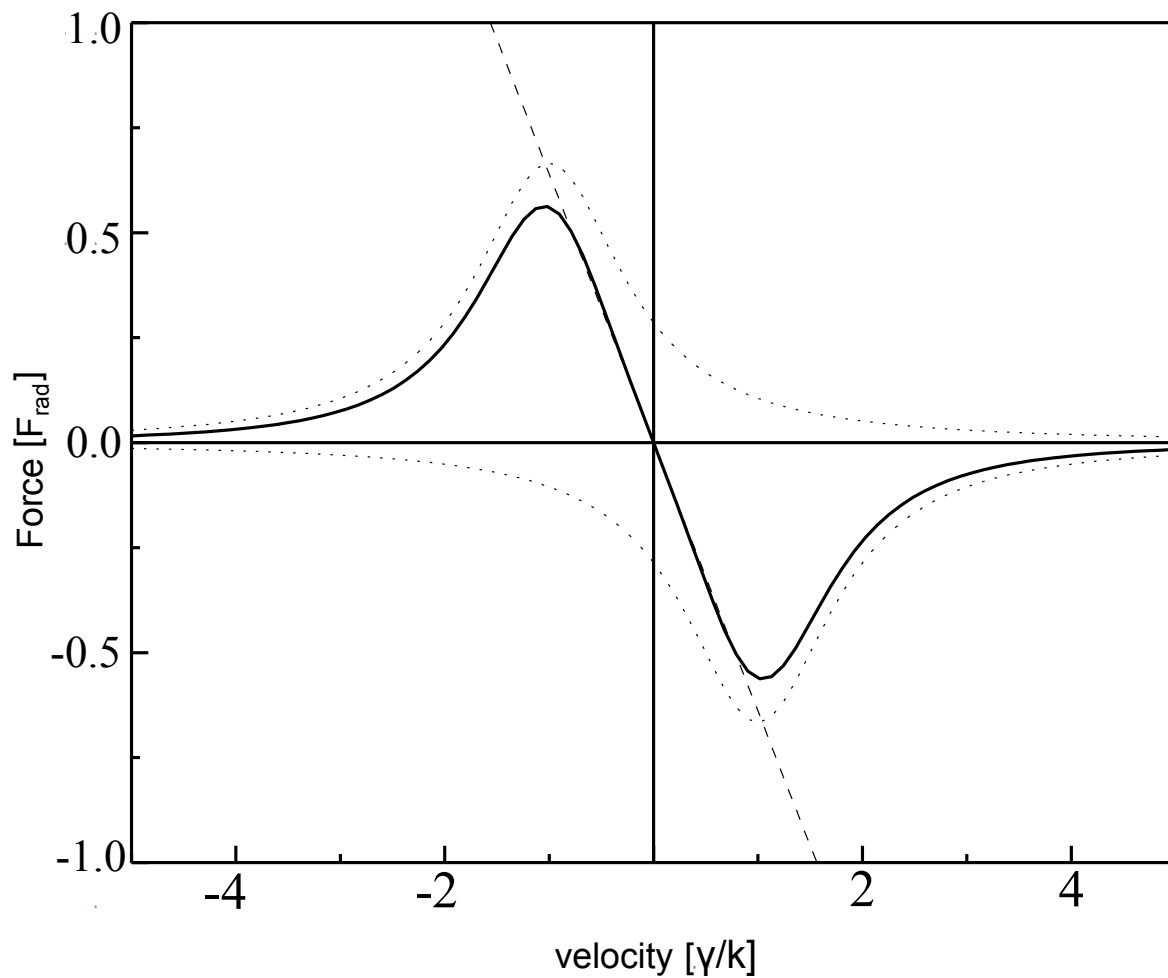


Figure 1.6: Velocity Dependence of Optical damping force in one dimensional picture with $s_0 = 2$ and $\delta = -\gamma$. The dotted lines show the force from each beam while the solid line is the sum of them. The dashed line represents Eq. 1.28 after the approximation.[6]

1.3.2 Stimulated Emission and Dipole Force

When the detuning δ is much greater than γ , the force derived from the absorption followed by stimulated emission is not limited by the properties of the atomic transition as in the radiative force, and can be much higher than the force from spontaneous emission. The example of a force from stimulated emission to excite an atom is the dipole force. Under two opposite laser setup, $2\hbar k$ momentum is transferred in each cycle. However, two oppositely light traveling beams forms standing wave due to the symmetry. The resulting dipole force can trap the atoms but not cool or slow down the atoms. When averaging over the wavelength, the dipole force would vanish.

1.3.3 Bichromatic Force

The bichromatic force is induced by two standing waves with the same detuning but completely different sign, in other words, the corresponding frequency is $\omega_{1,2} = \omega_l \pm \delta$ (shown in fig 1.7).

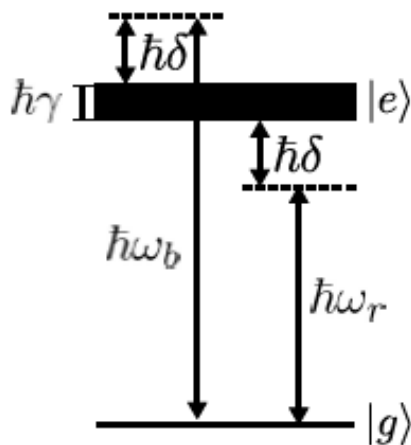


Figure 1.7: The bichromatic optical force with two frequencies ω_r and ω_b equally detuned above and below atomic resonance[7]

The resulting electric field is

$$E(t) = E_0[\cos(\omega_1 t) + \cos(\omega_2 t)] = 2E_0 \cos(\delta t) \cos(\omega_1 t) \quad (1.29)$$

So these two single frequency field can be rewritten as one amplitude modulated field with modulation period π/δ . To satisfy the π pulse model, the time integral should have the Rabi frequency $\Omega = \pi\delta/4$. It indicates that a single pulse could excite the atom. So once with proper relative phase setup, one beam can excite the atom by pulsed absorption, and another deexcite atom by pulsed stimulated emission. During the entire process, $\Delta p = 2\hbar k$ momentum is transferred to the atom (shown in fig 1.8). Here the Bichromatic force is $\mathbf{F} = 2\hbar\mathbf{k}\delta/\pi$. When the detuning is much larger than the decay rate $\delta \gg \gamma$, the bichromatic force turns out to be much larger than the radiative force. Increasing δ leads to larger force as long as we keep the π -pulse condition satisfied.

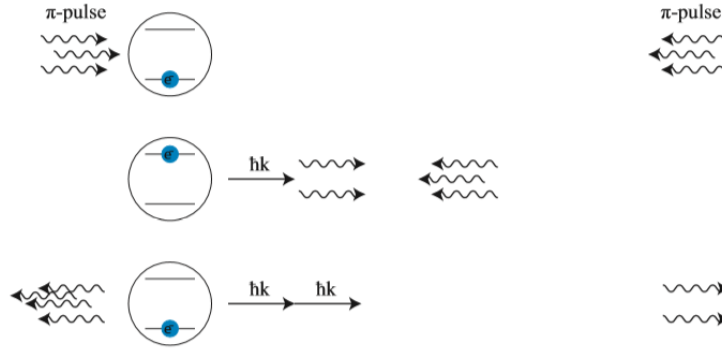


Figure 1.8: Bichromatic force profile: the two frequencies in the bichromatic laser beam create π -pulses and each excites or de-excites the atom. [2]

Chapter 2

Theory of Adiabatic Rapid Passage

The absorption of radiative forces is already discussed in the chapter 1. And it relies on spontaneous emission to push the atom in random direction, which limits the optical force affecting the atoms. Adiabatic Rapid Passage is the coherent process that a sequence of counterpropagating pulse pairs of laser light shine on the atom inducing absorption followed by stimulated emission. A momentum $2\hbar k$ is transferred in one cycle of duration T . This process relies only on stimulated emission rather than spontaneous emission, and the sweeping time through resonance is much less than life time of spontaneous emission.

2.1 Adiabatic Rapid Passage

The description of Adiabatic Rapid Passage is best done in term of the motion of the Bloch vector on the Bloch sphere under the influence of a changing torque vector. It is driven by chirped optical pulses. The optical frequency is swept through the resonance frequency of a atomic transition through $\pm\delta$. The amplitude of the light is pulsed and the detuning reaches resonance at the maximum value of amplitude-envelope.

The Hamiltonian for two level system is described as

$$\hat{H}_0 = \frac{1}{2}\hbar\omega_a\{|e\rangle\langle e| - |g\rangle\langle g|\} \quad (2.1)$$

We describe the interaction picture between two-level system and a classical oscillating electric field by Hamiltonian:

$$\hat{H} = \hat{H}_0 + \hat{V}_0\cos\omega t \quad (2.2)$$

$\hat{V}_0 = -\mu E_c(\hat{\mathbf{r}})$ defines the electric dipole. Knowing that $|e\rangle\langle e| + |g\rangle\langle g| = 1$, we could rewrite the Hamiltonian:

$$\hat{V}_0 = -\{|e\rangle\langle e| + |g\rangle\langle g|\}\mu \cdot E_c(\hat{\mathbf{r}})\{|e\rangle\langle e| + |g\rangle\langle g|\} \quad (2.3)$$

$$= d \cdot E_c(\hat{\mathbf{r}})\{|e\rangle\langle g| + |g\rangle\langle e|\} \quad (2.4)$$

where $d = -\langle e|\hat{\mu}|g\rangle$ is the transition dipole moment matrix element. So the Hamiltonian can be expressed as

$$\hat{H} = \frac{1}{2}\hbar\omega_a\{|e\rangle\langle e| - |g\rangle\langle g|\} + d \cdot E_c(\hat{\mathbf{r}})\{|e\rangle\langle g| + |g\rangle\langle e|\}\cos\omega t \quad (2.5)$$

$$= \frac{1}{2}\hbar\omega_a\{|e\rangle\langle e| - |g\rangle\langle g|\} + \frac{1}{2}\hbar\delta\{|e\rangle\langle e| - |g\rangle\langle g|\} + \hbar\Omega\{|e\rangle\langle g| + |g\rangle\langle e|\}\cos\omega t \quad (2.6)$$

where $\Omega = (d \cdot E_c)/\hbar$ and δ is the detuning of the driving field from resonance [8][9]. The second and third terms can be used to describe the system in terms of the matrix:

$$H = \frac{\hbar}{2} \begin{pmatrix} 0 & \Omega(t) \\ \Omega^*(t) & 2\delta(t) \end{pmatrix} \quad (2.7)$$

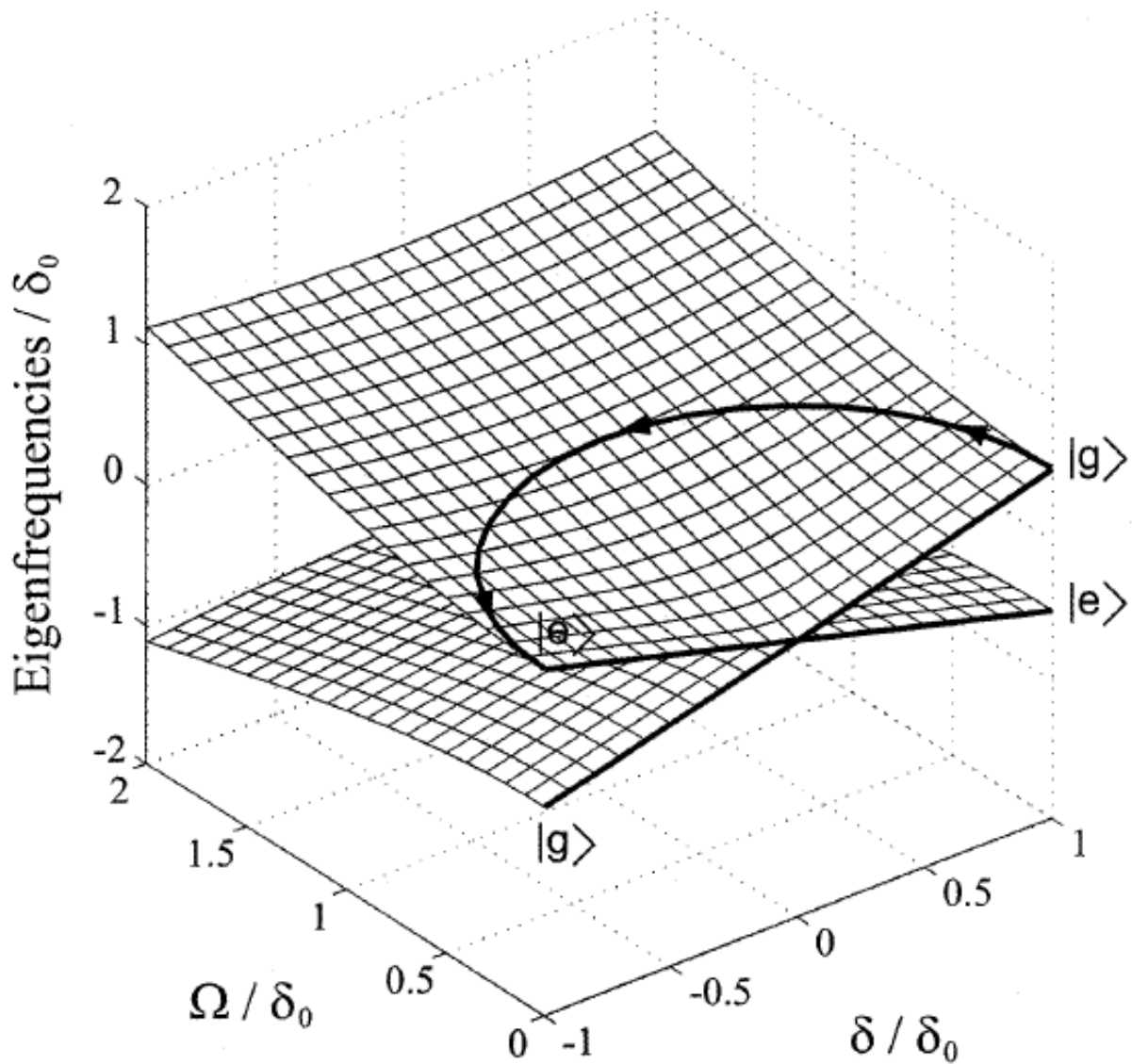


Figure 2.1: Two sheets represent the dressed energy states and cross at the original point. [4]

Where $\Omega(t)$ is time-dependent Rabi frequency with the same definition as in Chapter 1. The associate eigenenergies are $E(t) = \pm(\hbar/2)\sqrt{(\delta(t))^2 + (\Omega(t))^2}$.

The alternative explanation is that we could consider the interaction between light field

and atom is mapping the rotation of the Bloch vector and torque vector.

$$\frac{d\mathbf{R}(t)}{dt} = \boldsymbol{\Omega}(t) \times \mathbf{R}(t) \quad (2.8)$$

the detuning is $\delta(t) = \omega_l - \omega_a$, the difference between the frequency of laser and atom. The sweeping process is described as follows. The torque vector $[\Omega_r, \Omega_i, -\delta(t)]$ begins on the south pole where the Rabi frequency is 0, it moves slowly and continuously to the north pole. While going through the sweep, the Rabi frequency reaches its maximum value and then decreases to zero when it arrives at the north pole. In this process,

$$\begin{aligned} \delta(t) &= \delta_0 \cos(\omega_m t) \\ \Omega(t) &= \Omega_0 |\sin(\omega_m t)| \end{aligned} \quad (2.9)$$

where the sweeping time is π/ω_m and sweeping range is $2\delta_0$. Here we need to guarantee $\delta_0 \sim \Omega_0$ to ensure the best shape of trajectory on the Bloch sphere. And they should both be far larger than ω_m , to ensure that the torque vector is moving slowly enough. A quantitative description of how slow is it is given by,

$$|\boldsymbol{\Omega}| \gg \frac{d\theta(t)}{dt} \quad (2.10)$$

where $2\theta(t) \equiv \arctan(\Omega(t)/\delta(t))$ [10], is called the mixing angle, which is the mixing angle of the eigenstates in the dressed-atom picture. ARP is typically used in the parameter space given by

$$\delta_0 \sim \Omega_0 \gg \omega_m \gg \gamma, \quad (2.11)$$

where δ_0 is the amplitude of the frequency sweep, Ω_0 is the peak Rabi frequency, ω_m is the

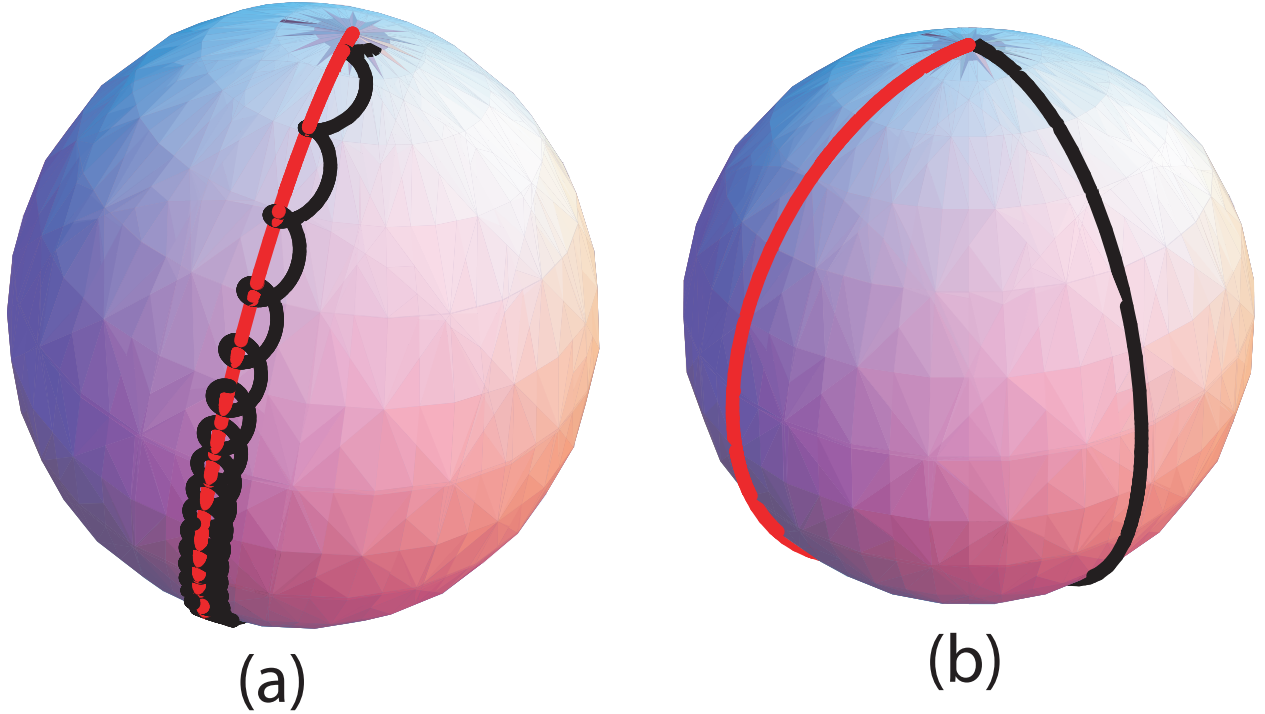


Figure 2.2: Plot of two trajectories. In Part (a), $\delta_0 = 30 \omega_m$ and $\Omega_0 = 50 \omega_m$, \mathbf{R} rotated along $\mathbf{\Omega}$ with high frequency. In Part (b), $\delta_0 = 1.10 \omega_m$ and $\Omega_0 = 1.61 \omega_m$. \mathbf{R} is $\sim 90^\circ$ away from the path of $\mathbf{\Omega}$. [2]

modulation frequency, and γ is the spontaneous decay rate. Once all these conditions are satisfied, the Bloch vector will track the vector vector adiabatically from the ground state to the excited state, and back to excited state again. These process is back and forth with the laser pulse shining.

We change the the interaction picture in the equivalent form that

$$H = \frac{\hbar}{2} \begin{pmatrix} 0 & \Omega(t)e^{-i\delta(t)} \\ \Omega(t)e^{-i\delta(t)} & 0 \end{pmatrix} \quad (2.12)$$

where the Torque vector can be expressed as $[\Omega_r e^{-i\delta(t)}, \Omega_i e^{-i\delta(t)}, 0]$ and the real and imaginary

part of Rabi frequency is expressed as

$$\Omega_r = \Omega_0 |\sin(\omega_m t)| \cos \alpha(t) \quad (2.13a)$$

$$\Omega_i = \Omega_0 |\sin(\omega_m t)| \sin \alpha(t) \quad (2.13b)$$

where $\alpha(t)$ is

$$\alpha(t) = -(\delta_0/\omega_m) |\sin(\omega_m t)| + (\mathbf{k} \cdot \mathbf{v})t + \phi \quad (2.14)$$

where the first term comes from the detuning $\delta(t)$, the second term is because of the Doppler shifts, and the last one is referring to the phase difference between pulse pairs. The sign in front of determines the sweeping directions. Positive means sweeping up while negative means sweeping down.

Here the frequency detuning $\delta(t)$ is included in the complex phase $\alpha(t)$ as $-(\delta_0/\omega_m) |\sin(\omega_m t)|$ along with contributions that come from Doppler shifts, $(\mathbf{k} \cdot \mathbf{v})t$, and unavoidable phase differences between pulses in a pair, ϕ_{rel} , as well as between pairs, ϕ_{pp} (see Fig. 2.3). The sign of δ_0 determines the direction of the frequency sweep for a single pulse. A positive δ_0 signifies a sweep up in frequency and a negative δ_0 signifies a sweep down. In the experimental setup, we have counter-propagating light fields that interact with the atom. The interaction duration for each is π/ω_m . As in Fig. 2.3, the pulse shape is shown in the top, and the sweeping frequency with upward direction is represented in the bottom. After a pulse pair, there is $2\pi/\omega_m$ dead time. All of the above is one period, and we will repeated such period for many times.

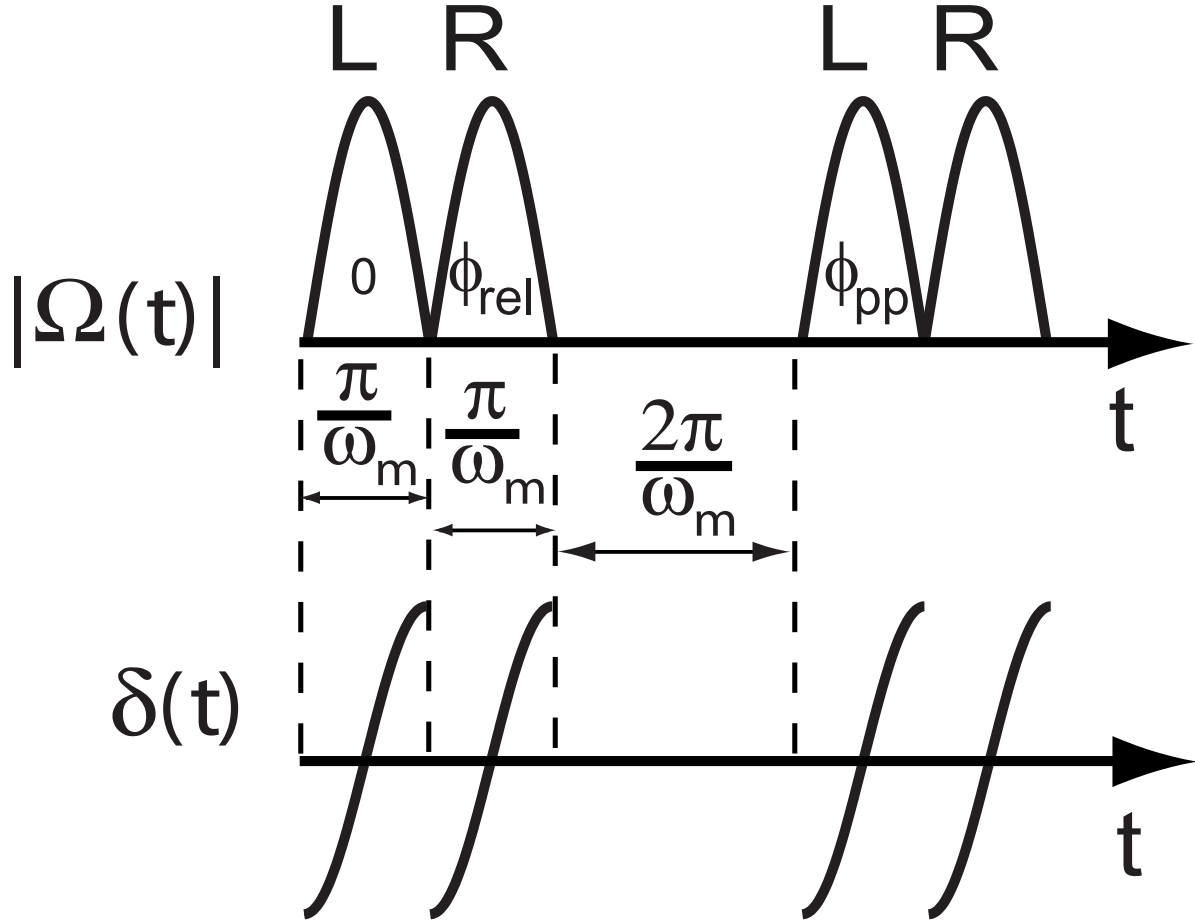


Figure 2.3: The plot shows how Rabi frequency $\Omega(t)$ and the frequency sweep $\delta(t)$ change in one period, two pulse followed by the dead time. [4]

2.2 ARP Force

Under periodic ARP pulses, the optical force, including Doppler shifts, can be calculated from the Optical Bloch Equation according to the Ehrenfest theorem.

$$\mathbf{F} = \langle -\nabla\mathcal{H} \rangle = \text{Tr}[\nabla\mathcal{H}\rho] = \frac{\hbar}{2}(\mathbf{u}\nabla\Omega_r + \mathbf{v}\nabla\Omega_i), \quad (2.15)$$

where $\Omega_{r,i}$ and \mathbf{u} and \mathbf{v} are the corresponding components of torque and Bloch vectors. For two counter-propagating light fields whose wavenumber is k , the resulting force on the atom

is

$$\mathbf{F} = \frac{\hbar \mathbf{k}}{2} ((\boldsymbol{\Omega}_+ - \boldsymbol{\Omega}_-) \times \mathbf{R})_3 \quad (2.16)$$

where $\boldsymbol{\Omega}_+$ and $\boldsymbol{\Omega}_-$ refer to the right and left propagating fields respectively. In the experimental setup shown in Fig. 2.3, we do not consider the case of temporally overlapping ARP pulses, so the force can be rewritten as follows

$$\mathbf{F} = \frac{\hbar \mathbf{k}}{2} (\boldsymbol{\Omega}_\pm \times \mathbf{R})_3 = \frac{\hbar \mathbf{k}}{2} \dot{w}. \quad (2.17)$$

A momentum of $\hbar k$ is exchanged between the light field and the atom when the Bloch vector is driven from one pole to the other. The force of a single pulse expressed through discretization of \dot{w} is

$$\mathbf{F} = \frac{\hbar \mathbf{k}}{2} \frac{\omega_m}{\pi} \Delta w \quad (2.18)$$

Chapter 3

Numerical Study and Result

In this chapter, we will present the nature of the program which simulates the interaction between the optical field and helium atoms. The process of multiple sequences of adiabatic rapid passage is simulated by changing sweep range, sweep direction, number of pulses, making pulse distortion and exploring sensitivity of parameters. It will start from various initial conditions for an atom and then propagate through the Optical Bloch Equations (OBEs) to a final landing position on the detector. The influence of the parameters will be given in the following. A sample program written in Python and a data analysis program in the R language will be shown in the appendix.

3.1 ARP Force Simulation

The simulation serves for the comparison with the experimental result, in which some of the parameters cannot be adjusted. We try to vary the parameters to explore the nature behind the experiment. For example, in the experiment, we cannot measure the exact value of Ω_0 and δ_0 , but in the simulation, we could fix them according to the force map to find

the possible strongest force area. And also, for the spontaneous emission, the decay rate is fixed for the specific atomic transition, but for the simulation, it is easy to vary, in order to understand how it influences the interaction process. We begin with equation 3.1, the Optical Bloch equations with spontaneous emission written as:

$$\dot{u} = \Omega_i w - \delta v - (\gamma/2)u \quad (3.1a)$$

$$\dot{v} = -\Omega_r w + \delta u - (\gamma/2)v \quad (3.1b)$$

$$\dot{w} = \Omega_r v - \Omega_i u - \gamma(w + 1) \quad (3.1c)$$

where u, v, w are the component of $\vec{R} = \vec{R}(u, v, w)$. From the 2.13b and 2.14, we know in the interaction picture that the torque vector is given as follows.

$$\mathbf{\Omega} e^{-i\alpha(\mathbf{t})} = (\Omega_{\text{real}}, \Omega_{\text{imag}}, 0)$$

$$\Omega_{\text{imag}} = |\Omega_0 \sin(\omega_m t)| \sin(-\delta_0/\omega_m \sin(\omega_m t) \pm \mathbf{k} \cdot \mathbf{v}t \pm \phi/2)$$

$$\Omega_{\text{real}} = |\Omega_0 \sin(\omega_m t)| \cos(-\delta_0/\omega_m \sin(\omega_m t) \pm \mathbf{k} \cdot \mathbf{v}t \pm \phi/2)$$

we use the torque vector below in the OBE's. The parameters for the calculations are as follows: n is the number of pulse pairs which interact with the atom and they are assumed to be perfectly coherent. Ω_0 is the peak Rabi frequency of the pulses, δ_0 is the amplitude of the frequency sweep, γ is the spontaneous emission rate, and ϕ is the phase of the optical field, which we average over 25 values equal spaced from 0 to 2π . v is the initial transverse velocity. Finally, independent of the other parameters, the atom always starts in the ground state $\mathbf{R}(u, v, w) = (0, 0, -1)$.

3.1.1 Force Maps

From Fig. 2.3, we know each time period is divided into four time units, with two pulses in opposite directions and two units of dead time. In the simulation, the time units are set to be π/ω_m . After each time unit, we calculate the force. Only in the first two time units with the pulse pair interacting with the atom, the force changes as

$$d\mathbf{F} = \pm(w_i - w_{(i-1)})\hbar k/(2t_{\text{pulse}}) \quad (3.2)$$

$$(3.3)$$

where we assume that during the dead time, the decay on the atom is negligible and does not influence the average force. The sign of $d\mathbf{F}$ indicate during the pulse pair interaction,if the atom gets perfectly inversion from ground state to excited state, we have $w = w_i - w_{i-1} = 1 - (-1) = 2$, otherwise, the atom experience gets perfectly inversion from excited state, we have $w = w_i - w_{i-1} = -1 - 1 = -2$.

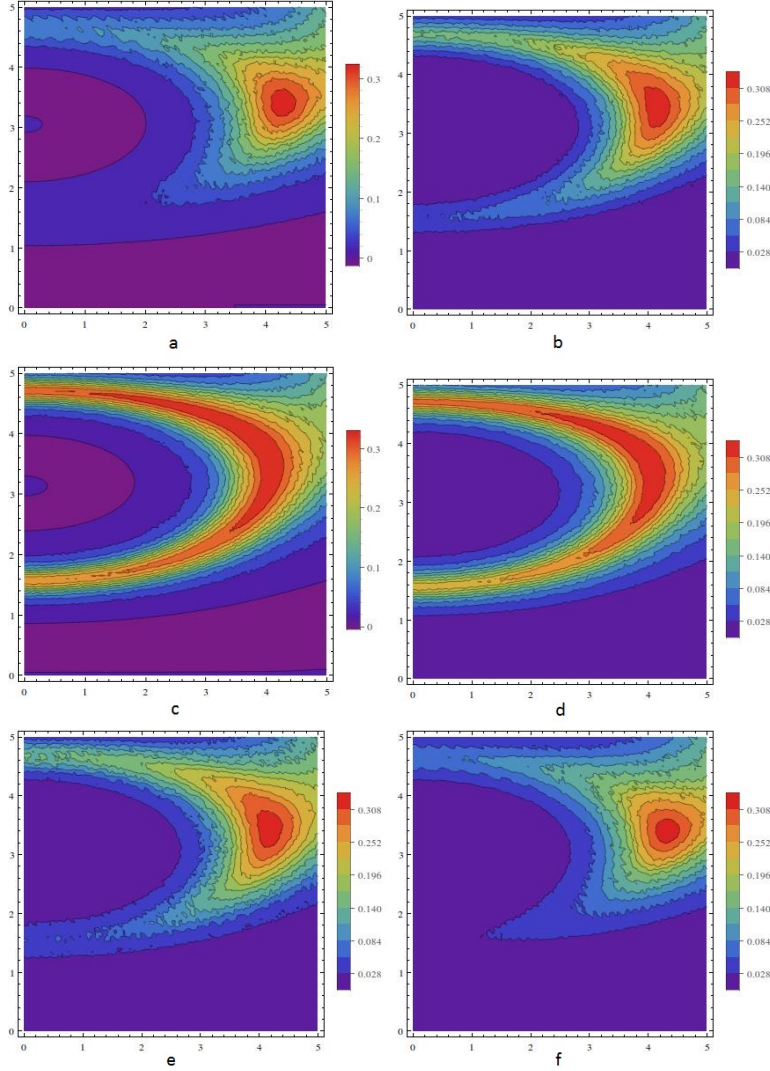


Figure 3.1: The contour plot for the average force over different velocity equal to -120,-80,-20,20,80,120 m/s corresponding to plots a,b,c,d,e,f with $\Omega_0 = 5.0\omega_m$ and $\delta_0 = 5.0\omega_m$

For the force maps, we vary the relative phase between pulses, and then average these results for the plots. Fig.3.1 shows the contour plots of the average force calculated over a range of the parameters under 320 pulses, $\Omega_0 = 5.0\omega_m$ and $\delta_0 = 5.0\omega_m$. We take the velocity at -120, -80, -20, 20, 80, 120 m/s mapping to the plots a,b,c,d,e,f in Fig. 3.1. First, We can see the color representing from red to blue according to the strength of the

force. It is much more sensitive to the value of Ω_0 and δ_0 , compared to the force map with the phase difference between pulse and atom is zero [2]. Second, the velocity's absolute value determines the shape of the plots, in other words, whenever the velocity is positive or negative, as long as they have the same absolute value, the plots are quite similar. Third, we found for the lower velocity the region of strong force is larger than that with higher velocity.

3.1.2 Force vs Velocity

The simulation process can be performed in the following two regions, the interaction region and drift region (shown in fig.3.2).

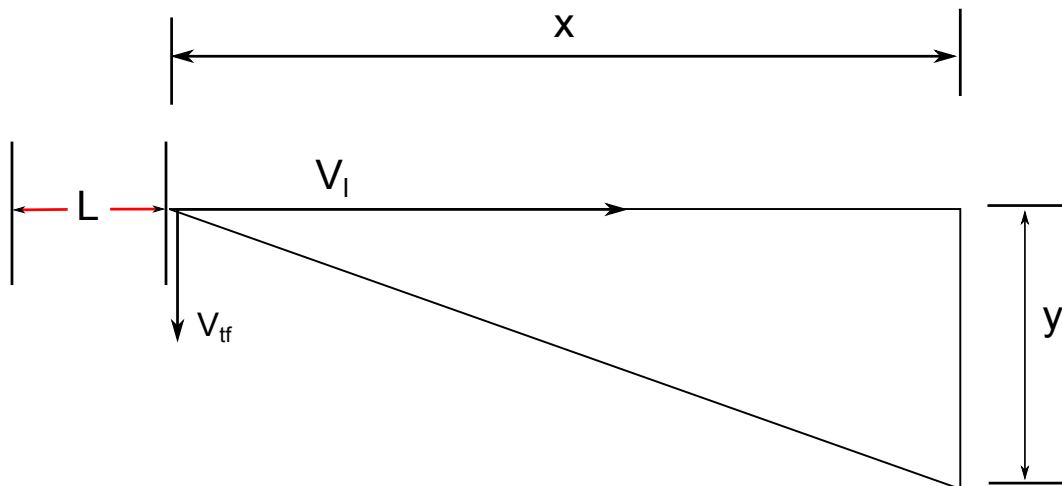


Figure 3.2: Trajectory of the atom movement[10]: after the atom leaves the interaction region(L), it will fly with longitudinal velocity(V_l) and transverse velocity (V_{tf})

In the drift region, there is no laser interaction, so we analyze this relative easy process first. In the experiment, there is longitudinal velocity distributes due to the atomic beam source. That can be measured from the equipment. However, the initial longitudinal velocity defined as the velocity the atom has before it enters the interaction region, determines the

number of pulses seen by the atoms because of the fixed interaction region size. This makes the atoms have different final transverse velocity changes, therefore, causing the atoms to land at different positions.

Therefore, for the drift region, the final transverse velocity and final longitudinal velocity depend on the time that the atom has for the interaction with the laser light and are v_t and v_l , the drift distance after interaction is x . The landing position is y , L is interaction length, n is the pulses number for interaction and t is the drift time.

$$y = v_t * t \quad (3.4)$$

$$x = v_l * t \quad (3.5)$$

$$t = T * n = L/v_l \quad (3.6)$$

where T is the period of a pulse pair interaction, so we found the landing position is $y = v_t * x * n * T/L$.

In the interaction region, the dragged atom model and moving atom model are compared in this part. The dragged atom picture (v is constant) can tell us about force vs velocity but cannot be used to simulate an experiment where the velocity change is important. In the dragged atom model, we ignore the the phase change contributed from the velocity change of the atom. Such an approximation is applied because the phase change by the velocity due to the Doppler shift is typically smaller than the natural linewidth of the atomic transition, but it does not apply for our case[10]. In the moving atom model, the transverse velocity change in each interaction period is calculated as

$$\Delta v = \frac{2dF * T}{M} \quad (3.7)$$

where M is the mass of the atom. The change of the velocity causes the change of the phase which can be calculated as:

$$\Delta\phi = 2\pi \frac{\pi}{\omega_m} \frac{\Delta v}{\lambda} \quad (3.8)$$

Fig.3.3 is the comparison between dragged atom and moving atom, the force vs velocity plot with the parameters $\Omega_0 = 3.37\omega_m$ and $\delta = 4.11\omega_m$. The x-axis and y-axis represent the initial transverse velocity and the average force after 320 pulses respectively, where we assume a single longitudinal velocity. We found there are two significant differences. One is that the starting velocity for the maximum force shifts from $v = 0$ to $v < 0$ in the moving atom model. The other is the presence of big resonances happening on higher absolute transverse velocity, disappearing in the moving atom model.

The reason for the shift of the peak in the moving atom picture is because the atom beginning with $v = 0$ sees a lower average force than atoms beginning with negative velocity. The atom beginning with $v = 0$ and $v < 0$ would experience the velocity increasing during the interaction. But with the velocity increasing, the force placed on the atom beginning with $v = 0$ would keep decreasing, while the force placed on the atom beginning with $v < 0$ would increase first to the maximum and then decrease. Therefore, the total average force of the latter is stronger than on the former one. But in the dragged atom picture, the atoms beginning with $v = 0$ are always under the maximum average force. Thus in the graph there is a shift of the entire curve from zero velocity to negative velocity in the moving atom model. For the vanishing of the resonances, one possible explanation is that in the moving atom model, for the velocity change, it is more like a moving average filter which smooths the curve and smears the resonances.

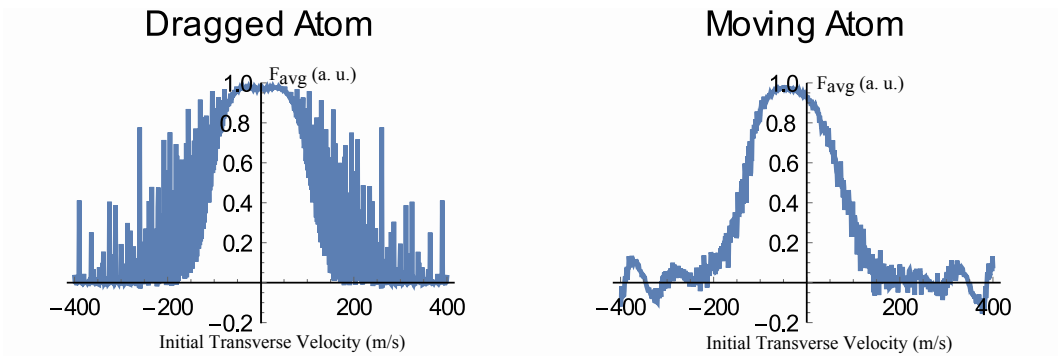


Figure 3.3: The force vs velocity plot of dragged atom and moving atom[10] with $\delta_0 = 4.11\omega_m$ and $\Omega_0 = 3.37\omega_m$ after 320 pulses interaction.

3.2 Parameter Effects

In this part, we try to vary the parameters to learn the effect on the results. For all the plots, δ_0 and Ω_0 are the same values used in the calculation always in units of ω_m .

3.2.1 Spontaneous Emission

The optical Bloch equations have the inherent capability to allow relaxation by spontaneous emission, and this requires that the frequency sweep rate is much larger than the atomic decay rate. Otherwise, it would preclude complete atomic inversion. From [11], we know that when the ratio of $\omega_m/\gamma \approx 1$, the spontaneous emission returns the atom to the ground state frequently so that the counterpropagating pulse pairs can not fully transfer momentum $2\hbar k$ to atoms, but with the ω_m/γ increasing, the coherence between atoms and light interaction can be preserved and the expected momentum can be transferred to the atoms before any spontaneous emission occurs. In our simulation, $\gamma = 1.62 \times 10^6 * 2\pi$ and $\omega_m = 160 \times 10^6 \pi$, so the ratio of $\omega_m/\gamma \approx 100$.

Fig.3.4 shows the atomic distribution after 540 pulses pairs with $\Omega_0 = 3.39\omega_m$ and $\delta_0 =$

$4.19\omega_m$. In the experiment the value of spontaneous emission (γ) can not be changed, but for the calculation we can change the value so that we can better understand its influence on the average force. We made plots with the ratio of the simulation spontaneous emission to real spontaneous emission equal to 1 and 0 respectively and with the dead time period, we found even for $\omega_m/\gamma \gg 1$, spontaneous emission still influence the atom landing positions. In the 8 plots in fig.3.4, we vary γ to see gradual change of the influence.

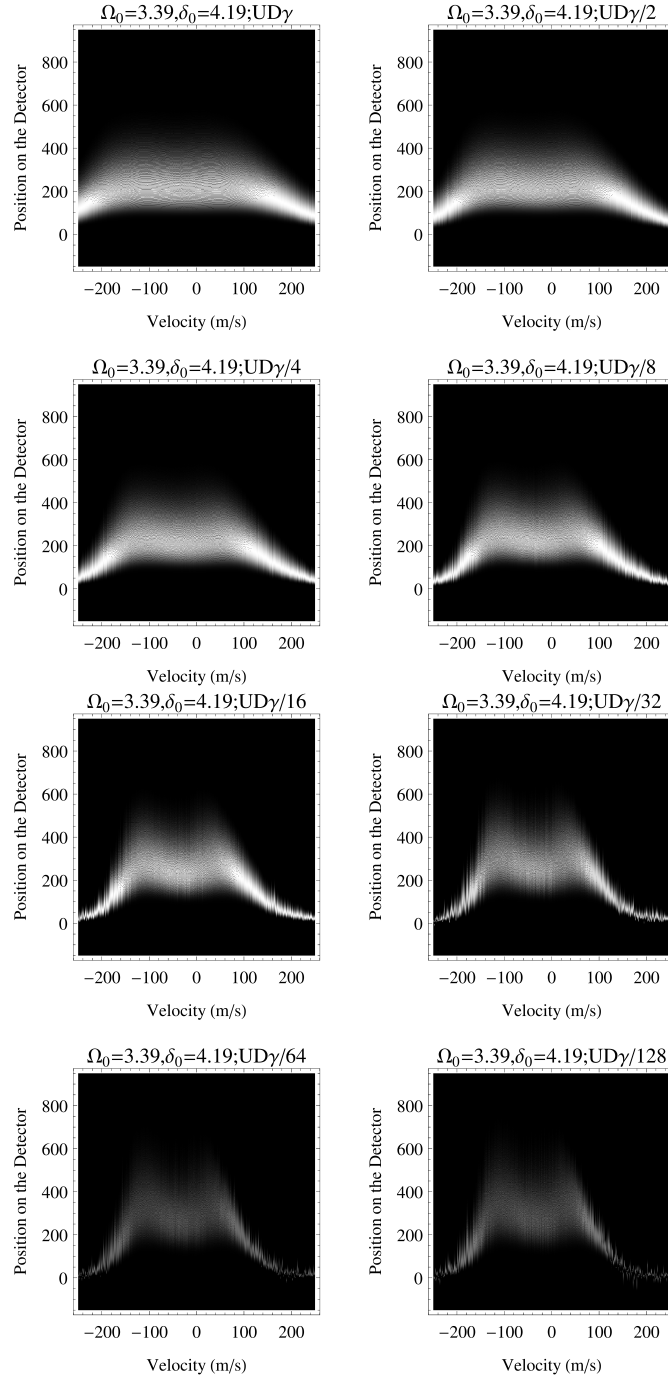


Figure 3.4: Force vs velocity with different γ with the sweeping direction from up to down(UD) in the moving atom simulation, where different γ means we manually set up the spontaneous emission rate into $\gamma, \gamma/2, \gamma/4$ and so on, where vertical axis is arbitrary units

Although we cannot directly compare the simulation result and the experimental result, we still can compare the characters of the shape. The unexpected result after we compare the simulation to the experiment result of fig.3.5, we found the experimental result is more like the simulation whose spontaneous emission is close to zero.

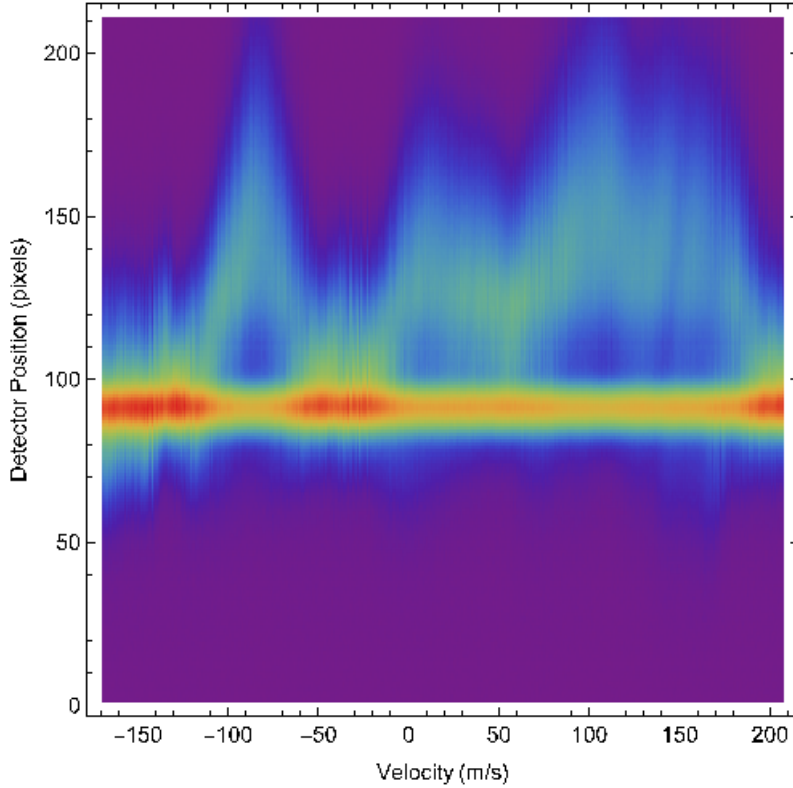


Figure 3.5: The experiment result with $\Omega_0 = 3.70\omega_m$ and $\delta_0 = 3.25\omega_m$

3.2.2 Sweeping Direction

Reversing the sweep direction leads to different paths for Bloch vector but does not change the end points of the trajectory. For example, if an atom starts near the north pole, such a frequency-swept pulse leaves it near the south pole, and vice versa. In fig.3.6, we change the sweeping direction of the first pulse and second pulse by down-down, down-up,

up-down and up-up without spontaneous emission. The laser parameters are kept the same for all four plots. The results show that there is no big change in the position by changing the sweeping direction.

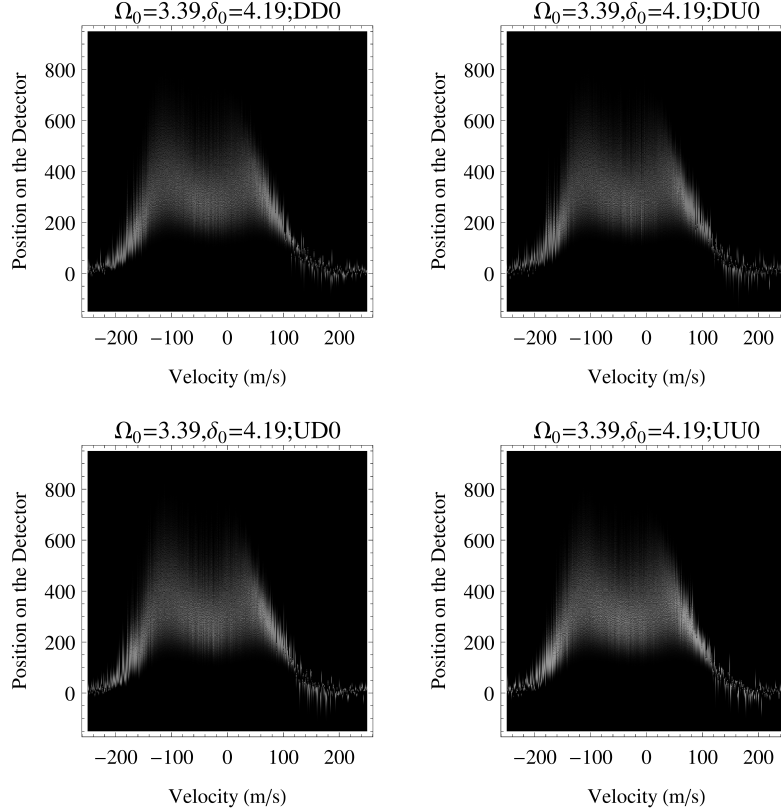


Figure 3.6: Sweeping direction in down-down(DD), down-up(DU), up-down(UD) and up-up(UU) direction without spontaneous emission.

Fig.3.7 shown in the thesis[2], which is asymmetric for the dragged atom model with twelve and sixty pulse pairs. The explanation for the unexpected asymmetric force is due to the time difference the atom spends in the excited state. Fig. 3.6 shows symmetric peak for the moving atom model. With more pulse pairs(320 pulses pairs), the asymmetric force disappears.

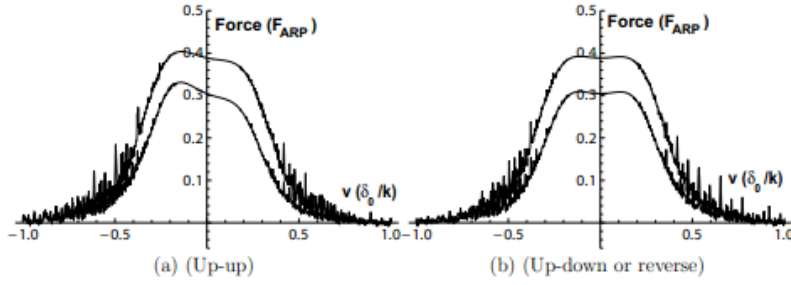


Figure 3.7: Calculated velocity dependence of the ARP force for two sweep schemes with $\delta_0 = 4.19\omega_m$ and $\Omega_0 = 3.39\omega_m$ [2]

3.2.3 Number of Pulses

Fig 3.8 shows that the force changes with different initial transverse velocities for the case of $\delta_0 = 4.55\omega_m$ and $\Omega_0 = 3.37\omega_m$ under the pulses number of 20, 160, 320, 540. When more and more pulses interact with the atoms, the curve begin to split at some specific velocity. The forces of two selected transverse velocities become stronger and stronger with increasing the number of pulses compared to fewer. Also we could find out the curve with a little shift to the left due to the velocity change. Third, the average force acting on the moving atoms is weaker than dragged atom case in that with the increasing number of the pulses, it is just like a signal taken through a moving average filter with large enough window. It would be more smooth with fewer resonances.

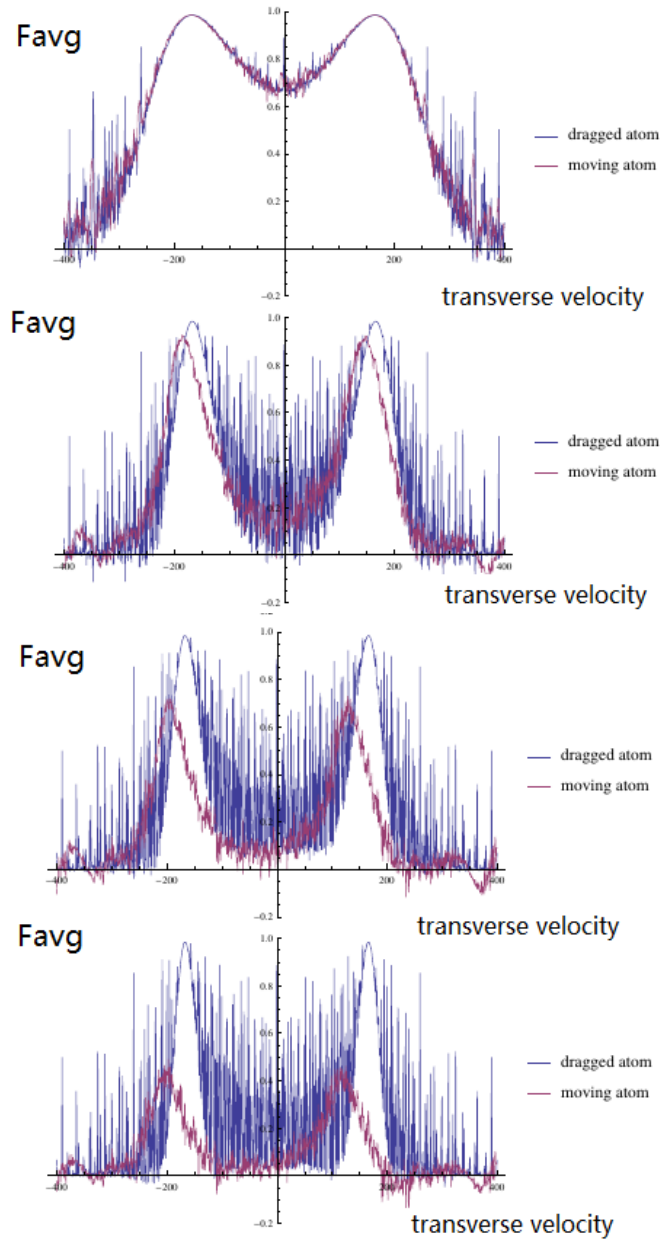


Figure 3.8: Force changing based on the pulses number of 20, 160, 320, 540 with $\delta_0 = 4.19\omega_m$ and $\Omega_0 = 3.39\omega_m$ sweeping from up to down.

3.2.4 Distribution of Peak Distance

As is stated in the previous part that the result split at the specific δ_0 , and we found the peak distance increasing with δ_0 . Therefore, we try to explore the relationship between the peak separation and δ_0 . To eliminate personal judgment, the following Lorentzian function is used to identify the peak position, which is show as following:

$$f(\mathbf{v}, w_f) = \frac{w_f}{(\mathbf{v} + \mathbf{v}_0)^2 + w_f^2} \quad \text{for } (\mathbf{v} + \mathbf{v}_0) < 0 \quad (3.9)$$

$$g(\mathbf{x}, w_g) = \frac{w_g}{(\mathbf{x} + \mathbf{x}_0)^2 + w_g^2} \quad \text{for } (\mathbf{v} + \mathbf{v}_0) \geq 0 \quad (3.10)$$

where \mathbf{v}_0 and \mathbf{x}_0 are the peak center, and w_f and w_g are fitting parameters related to the widths of the peaks. The distance between two peaks can be expressed as Δv . After trying different functions to fit the computed results, finally, we found the following function has the best fitting,

$$k\Delta v = \mu (\delta_0 - \eta)^{1/\xi} \quad (3.11)$$

where μ , η , and ξ are the fitting parameters. k is the magnitude of the wavevector $2\pi/\lambda$. Fig 3.9 shows the fitting for the following parameters[10].

Ω_0 (ω_m)	η (rad/s)	ξ (dimensionless)	μ (rad/s ^(1/ξ))
3.27	4.11	2.66	435.1
3.37	4.09	2.51	422.5
3.47	4.08	2.42	431.1

Table 3.1: Table of values for the fits shown in Fig. 3.9.

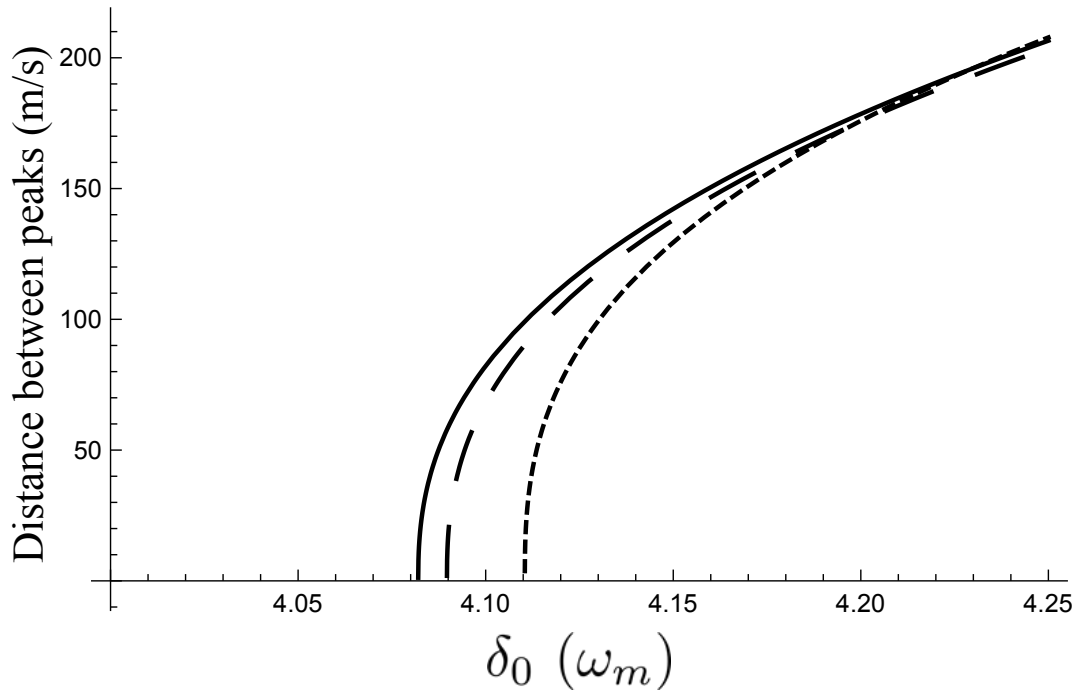


Figure 3.9: Plot of the peak splitting in the force vs. velocity calculation results. The curve from left to right follows the parameters in the table 3.1 from top to down. [10]

3.2.5 Pulse distortion

In this part, because the experimental pulse shape is not always perfect sin wave, we try to simulate the result. We try pulse distortion that is as close as possible to the source shape. We use the function: $\Omega = |\Omega_0 \sin(\omega_m t)|(1 - \varepsilon(\omega_m t - \frac{\pi}{2}))$. Fig.3.10 shows that the force is very sensitive to the distortion of the pulse shape.

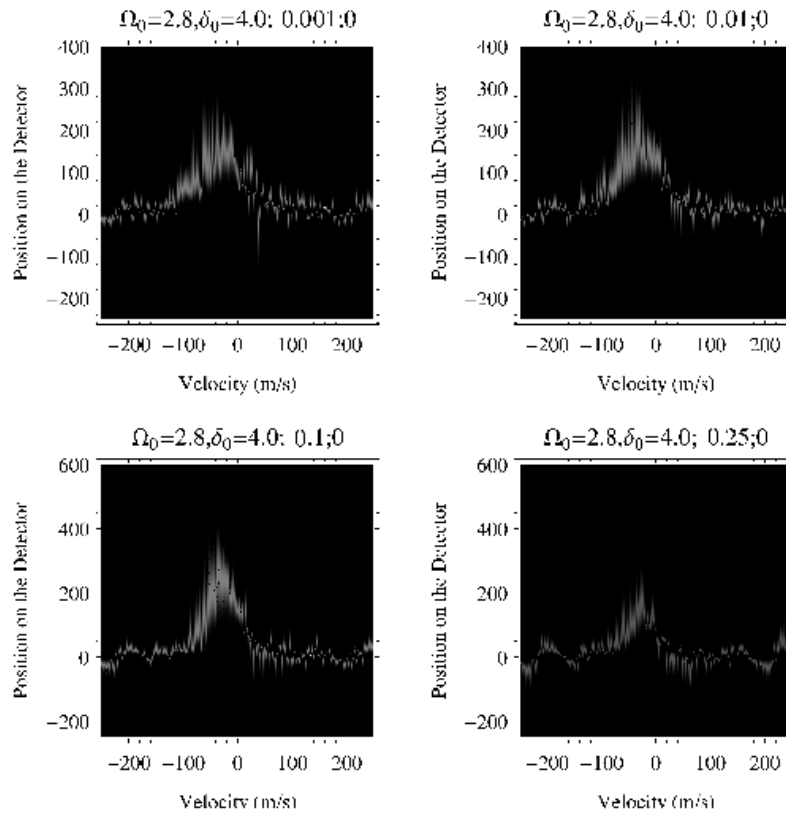


Figure 3.10: Force vs velocity with distortion on $\epsilon = 0.001, 0.01, 0.1, 0.25$ with $\delta_0 = 4.0\omega_m$ and $\Omega_0 = 2.8\omega_m$

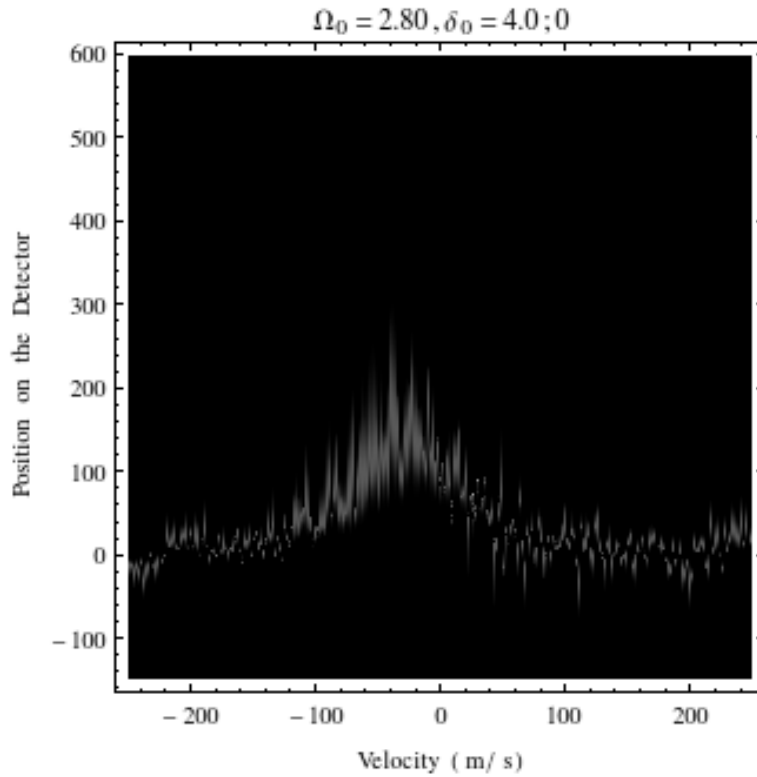


Figure 3.11: Force vs velocity without distortion with $\delta_0 = 4.0\omega_m$ and $\Omega_0 = 2.8\omega_m$

3.2.6 Sensitivity of parameters δ_0 & Ω_0

In this part, we try to figure out the resolution of the results, in other words, how sensitive the results are to the parameter changes. From Fig. 3.12, we could find that when δ_0 changes 0.04, there is significant shape change in the force vs velocity plot, but there is no big change when the Ω_0 varies.

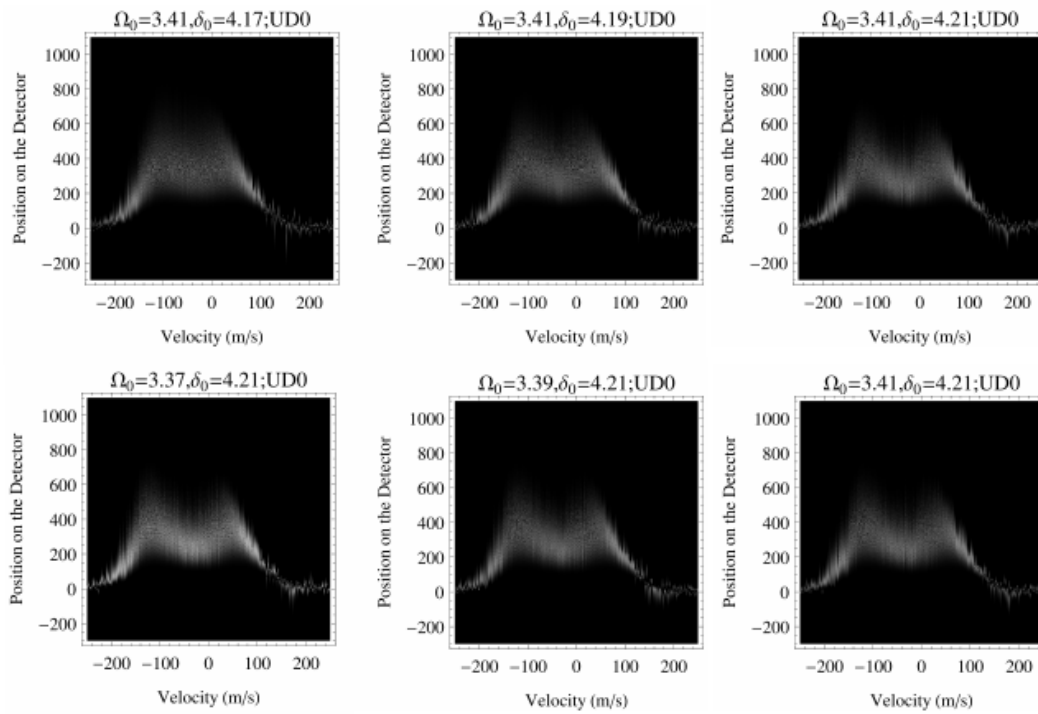


Figure 3.12: Resolution detection on force vs velocity plot by varying δ_0 and Ω_0 , the sweeping direction is from up to down

Chapter 4

Conclusions

Based on the two-level system theory, this thesis presents a study of the process of multiple sequence of adiabatic rapid passage by simulating the nature of the process for the interaction between optical forces and helium atoms. We try to understand the relationship between the average force and relevant parameters Ω_0 and δ_0 in the Rabi frequency. Furthermore, we study the relationship between the average force and single velocity by comparing the dragged atom model and moving atom model.

We also studied the process of multiple sequences of adiabatic rapid passage by changing spontaneous emission, sweep range, pulse number for interaction, exploring the peak distance, making pulse distortion, and resolution, trying to compare with the experimental result. We found several interesting phenomena when spontaneous emission rate is close to zero. This is more like the experiment result, also with the number of pulses increasing, the force is split into two peaks and the distance between the two peaks can be fitted into a specific formula. We could give explanations to some of them, but still cannot fully understand the cause for others.

In conclusion, the simulation result offers a very good description of the experiment, which

helps us to understand the nature of the ARP force but also poses post many questions that need to be solve.

Bibliography

- [1] Russell Anderson. Two-level system : Rabi oscillations. 2008. www.ph.unimelb.edu.au/part3/notes/Rabi.pdf.
- [2] Daniel Thomas Stack. *Ph.D. thesis, Optical Forces from Periodic Adiabatic Rapid Passage Sequences on Metastable Helium Atoms*. 2012.
- [3] Harold J Metcalf and Peter Van der Straten. *Laser cooling and trapping*. Springer Science & Business Media, 2012.
- [4] Xiyue Miao. Ph.d. thesis, optical force on atoms with periodic adiabatic rapid passage sequences. 2006.
- [5] Bruce W Shore. *Manipulating quantum structures using laser pulses*. Cambridge University Press, 2011.
- [6] George Horace Brooke Thompson. Physics of semiconductor laser devices. *Chichester, Sussex, England and New York, Wiley-Interscience, 1980. 572 p.*, 1980.
- [7] Christopher Scott Corder. Ph.d. thesis, optical forces on metastable helium. 2014.
- [8] T. Fischer, P. Maunz, P. W. H. Pinkse, T. Puppe, and G. Rempe. Feedback on the motion of a single atom in an optical cavity. *Phys. Rev. Lett.*, 88:163002, 2002.
- [9] B.C. Buchler N.P.Robins J.D.Close, R. Poldy. A single atom in an opticalcavity: An open-quantum system. 2012.
- [10] John David Elgin. Ph.d. thesis, study of the velocity dependence of the adiabatic rapid passage (arp) optical force in helium. 2015.
- [11] Daniel Stack, John Elgin, Petr M Anisimov, and Harold Metcalf. Numerical studies of optical forces from adiabatic rapid passage. *Physical Review A*, 84(1):013420, 2011.

Appendix A

Python Code for Position-velocity Plot Numerical Calculations

```
# -*- coding: utf-8 -*-

#Created on Mon Jan 19 18:27:00 2015

#@author: He Zhang , John Elgin

from numpy import *
from scipy.integrate import odeint
from pylab import *

#####

f = open("data20150121withgamma.txt", 'wb')
```

```
#f. write('phi'+','+ 'vstart/velfact'+','+ 'FF/(2.0*(i/4)*forfact)'+','+ 'n'+ '\n')
```

```
# define parameters
```

```
lambda_ = 1083.33e-9
```

```
hbar = 1.054571e-34
```

```
k = 2.0*pi/lambda_
```

```
gamma = 1.62e6*2*pi
```

```
mass = 6.64648e-27
```

```
wm = 160e6*2*pi
```

```
velfact = gamma/k
```

```
forfact = hbar*k*gamma
```

```
tpulse = 1.0*pi/wm
```

```
p = 4.0*pi/wm
```

```
#####
```

```
# solver parameters
```

```
ode_rtol = 1e-5
```

```
ode_atol = 1e-5
```

```

#####
# user parameters
phasepoints = 5
print 'Phasepoints:', phasepoints

npulses = 20
print 'pulse pairs:', npulses

d = 4.19
print 'delta_0:', d
d0 = d*wm

w = 3.39
print 'Omega_0:', w
w0 = w*wm

vmin = -500
print 'Minimim value of velocity (m/s):', vmin

vmax = 500
print 'Maximum value of velocity (m/s):', vmax

dkv = 10
print 'Steps in velocity:', dkv

```

```

vstep = (vmax-vmin)/dkv
print 'The step size in velocity is:', vstep, 'm/s'

#####

#
dopoff = 0
rpar = array([gamma, -gamma/2, w0, d0/wm, 0.0, 0.0, wm, dopoff,
              0.0])

#####

# system function
def feval(u, t0, p):
    if (int(p[8])%4 == 1):
        a = abs(p[2]*sin(p[6]*t0)) * sin(-p[3]*sin(p[6]*t0) + (p
            [4]-p[7])*t0 + p[5]/2)
        b = abs(p[2]*sin(p[6]*t0)) * cos(-p[3]*sin(p[6]*t0) + (p
            [4]-p[7])*t0 + p[5]/2)
    elif (int(p[8])%4 == 2):
        a = abs(p[2]*sin(p[6]*t0)) * sin(-p[3]*sin(p[6]*t0) - (p
            [4]-p[7])*t0 - p[5]/2)
        b = abs(p[2]*sin(p[6]*t0)) * cos(-p[3]*sin(p[6]*t0) - (p
            [4]-p[7])*t0 - p[5]/2)

```

```

else :
    a = 0.0
    b = 0.0
    du0 = p[1]*u[0] - a*u[2]
    du1 = p[1]*u[1] - b*u[2]
    du2 = a*u[0] + b*u[1] - p[0]*(u[2]+1.0)
    return array([du0, du1, du2])

#####
# Main Loop
npairs = 4*npulses
term0 = -1.0
phistep = 2.0*pi/phasepoints

for l in range(1,dkv+2,1):
    for j in range(1,phasepoints+1,1):
        phi = phistep*j
        rpar[5] = phi
        v = vmin + (l-1)*vstep
        vstart = v
        dF = 0.0
        tstart = 0.0
        FF = 0.0
        h0 = tpulse/100

```

```

term0 = -1.0
u = array([0.0, 0.0, -1.0]) # initial Bloch Vector
for i in range(1,npairs+1,1):
    tstop = tstart + tpulse
    rpar[8] = i
    rpar[4] = k*v
    y = odeint(feval, u, [tstart, tstop], args=(rpar,),
               rtol=ode_rtol, atol=ode_atol)
    u=y[1,:]
    tstart = tstop
    #print '[u,v,w] = ', y[1,:]
    if (i%4==1):
        dF = (u[2]-term0)*hbar*k/(2.0*tpulse)
    elif (i%4==2):
        dF = -(u[2]-term0)*hbar*k/(2.0*tpulse)
    else:
        dF = 0.0
    v = ((dF*2.0*tpulse)/mass + v)
    #print 'dF=',dF, 'v=',v
    FF = FF+dF
    term0 = u[2]
    if (i%4==0):

```



```
print '>> ', phi, vstart/velfact, FF/(2.0*(i/4)*
    forfact), i/4
#result=phi, vstart/velfact, FF/(2.0*(i/4)*forfact
    ), i/4
f.write(str(phi)+' ', '+str(vstart/velfact)+' ', '+str(
    FF/(2.0*(float(i)/4)*forfact)+' ', '+str(i/4)+'\n
    ')
```

```
f.close()
```

Appendix B

Python Code for Force Map

Numerical Calculations

```
# -*- coding: utf-8 -*-  
#Created on Mon Jan 19 18:27:00 2015  
  
#@author: He Zhang , John Elgin  
  
from numpy import array , sin , cos , pi  
from scipy.integrate import odeint  
#from pylab import *  
from time import clock  
  
clock_start = clock()  
  
#####
```

```

f = open("data20150122contour.txt", 'wb')
#f.write('phi'+', '+ 'vstart/velfact'+', '+ 'FF/(2.0*(i/4)*forfact)
      '+', '+ 'n'+ '\n')

# define parameters
lambda_ = 1083.33e-9
hbar = 1.054571e-34
k = 2.0*pi/lambda_
gamma = 1.62e6*2*pi
mass = 6.64648e-27
wm = 160e6*2*pi
velfact = gamma/k
forfact = hbar*k*gamma
tpulse = 1.0*pi/wm
p = 4.0*pi/wm

#####

# solver parameters
ode_rtol = 1e-5
ode_atol = 1e-5

```

```

#####
# user parameters
phasepoints = 5
print 'Phasepoints:', phasepoints

npulses = 20
print 'pulse pairs:', npulses

wd = 5
print 'wd', wd
maxwd = wd*wm
print maxwd

step = 0.25
print 'Omega_0:', step
wdstep = step*wm
print wdstep
print wdstep/wm

vmin = -500
print 'Minimim value of velocity (m/s):', vmin

vmax = 500
print 'Maximum value of velocity (m/s):', vmax

```

```

dkv = 2
print 'Steps in velocity:', dkv
vstep = (vmax-vmin)/dkv
print 'The step size in velocity is:', vstep, 'm/s'

#####

#
dopoff = 0
rpar = array([0, 0, 0, 0, 0.0, 0.0, wm, dopoff, 0.0])

#####

# system function
def feval(u, t0, p):
    if (int(p[8])%4 == 1):
        a = abs(p[2]*sin(p[6]*t0)) * sin(-p[3]*sin(p[6]*t0) + (p
            [4]-p[7])*t0 + p[5]/2)
        b = abs(p[2]*sin(p[6]*t0)) * cos(-p[3]*sin(p[6]*t0) + (p
            [4]-p[7])*t0 + p[5]/2)
    elif (int(p[8])%4 == 2):
        a = abs(p[2]*sin(p[6]*t0)) * sin(-p[3]*sin(p[6]*t0) - (p
            [4]-p[7])*t0 - p[5]/2)

```

```

        b = abs(p[2]*sin(p[6]*t0)) * cos(-p[3]*sin(p[6]*t0) - (p
            [4]-p[7])*t0 - p[5]/2)
else :
    a = 0.0
    b = 0.0
    du0 = p[1]*u[0] - a*u[2]
    du1 = p[1]*u[1] - b*u[2]
    du2 = a*u[0] + b*u[1] - p[0]*(u[2]+1.0)
    return array([du0, du1, du2])

#####
# Main Loop
npairs = 4*npulses
term0 = -1.0
phistep = 2.0*pi/phasepoints

square = wd/step
print square

for l in range(1,dkv+2,1):
    v = vmin + (l-1)*vstep
    for m in range (0,int(square)+1):
        rpar[2] = m*wdstep
        for s in range(0,int(square)+1):

```

```

rpar[3] = s*wstep/wm
for j in range(1,phasepoints+1,1):
    phi = phistep*j
    rpar[5] = phi
    v = vmin + (l-1)*vstep
    vstart = v
    dF = 0.0
    tstart = 0.0
    FF = 0.0
    h0 = tpulse/100
    term0 = -1.0
    u = array([0.0, 0.0, -1.0]) # initial
        Bloch Vector
    for i in range(1,npairs+1,1):
        tstop = tstart + tpulse
        rpar[8] = i
        rpar[4] = k*v
        y = odeint(feval, u, [tstart, tstop],
            args=(rpar,), rtol=ode_rtol, atol
            =ode_atol)
        u=y[1,:]
        tstart = tstop
        #print '[u,v,w] = ', y[1,:]
        if (i%4==1):

```

```

        dF = (u[2] - term0)*hbar*k/(2.0*
            tpulse)
    elif (i%4==2):
        dF = -(u[2] - term0)*hbar*k/(2.0*
            tpulse)
    else:
        dF = 0.0
    v = ((dF*2.0*tpulse)/mass + v)
    #print 'dF=',dF, 'v=',v
    FF = FF+dF
    term0 = u[2]
    dF = 0
    if (i%4==0):

        #print '>> ',phi, vstart/velfact,
            FF/(2.0*(i/4)*forfact), i/4
        #result=phi, vstart/velfact, FF
            /(2.0*(i/4)*forfact), i/4
        f.write(str(phi)+' ',str(vstart)+
            ', ',str(m*wdstep/wm)+' ',str(s
            *wdstep/wm)+' ',str(FF/(2.0*(i
            /4)*forfact))+' ',str(i/4)+'\n
            ')
        #f.write(str(y[1,:])+'\n')

```



```
f.close()

clock_end = clock()
print("execution time:", clock_end-clock_start)
```

Appendix C

Python Code for Parallel Calculations

```
# -*- coding: utf-8 -*-

#Created on Mon Jan 19 18:27:00 2015

#@author: He Zhang , John Elgin

import sys, time
import pp

from scipy.integrate import odeint
from numpy import sin, cos, abs, array

# define parameters
```

```

pi = 3.1415926
lambda_ = 1083.33e-9
hbar = 1.054571e-34
k = 2.0*pi/lambda_
gamma = 1.62e6*2*pi
mass = 6.64648e-27
wm = 160e6*2*pi
velfact = gamma/k
forfact = hbar*k*gamma

tpulse = 1.0*pi/wm
p = 4.0*pi/wm

#####

# solver parameters
ode_rtol = 1e-5
ode_atol = 1e-5

#####

# user parameters
phasepoints = 5
npulses = 5
d = 4.19

```

```

d0 = d*wm
w = 3.39
w0 = w*wm
vmin = -500
vmax = 500
dkv = 5
vstep = (vmax-vmin)/dkv

#####

#
dopoff = 0

#####

# system function
def feval(u, t0, p):
    sin = numpy.sin
    cos = numpy.cos
    if (int(p[8])%4 == 1):
        a = abs(p[2]*sin(p[6]*t0)) * sin(-p[3]*sin(p[6]*t0) + (p
            [4]-p[7])*t0 + p[5]/2)
        b = abs(p[2]*sin(p[6]*t0)) * cos(-p[3]*sin(p[6]*t0) + (p
            [4]-p[7])*t0 + p[5]/2)

```

```

elif (int(p[8])%4 == 2):
    a = abs(p[2]*sin(p[6]*t0)) * sin(-p[3]*sin(p[6]*t0) - (p
        [4]-p[7])*t0 - p[5]/2)
    b = abs(p[2]*sin(p[6]*t0)) * cos(-p[3]*sin(p[6]*t0) - (p
        [4]-p[7])*t0 - p[5]/2)
else :
    a = 0.0
    b = 0.0
    du0 = p[1]*u[0] - a*u[2]
    du1 = p[1]*u[1] - b*u[2]
    du2 = a*u[0] + b*u[1] - p[0]*(u[2]+1.0)
return [du0, du1, du2]

```

```
#####
```

```
# Main Loop
```

```
npairs = 4*npulses
```

```
term0 = -1.0
```

```
phistep = 2.0*pi/phasepoints
```

```

def job_func(l, j, gamma, w0, d0, wm, dopoff, phistep, vmin, vmax, vstep,
    tpulse, npairs, k, ode_rtol, ode_atol, term0, hbar, mass, velfact,
    forfact):

```

```

    from scipy.integrate import odeint

```

```

    #for l in range(1, dkv+2, 1):

```

```

#for j in range(1, phasepoints+1, 1):
rpar = [gamma, -gamma/2, w0, d0/wm, 0.0, 0.0, wm, dopoff, 0.0]
phi = phistep*j
rpar[5] = phi
v = vmin + (l-1)*vstep
vstart = v
dF = 0.0
tstart = 0.0
FF = 0.0
h0 = tpulse/100
term0 = -1.0
u = [0.0, 0.0, -1.0] # initial Bloch Vector
result = []
for i in range(1, npairs+1, 1):
    tstop = tstart + tpulse
    rpar[8] = i
    rpar[4] = k*v
    y = odeint(feval, u, [tstart, tstop], args=(rpar, ), rtol=
        ode_rtol, atol=ode_atol)
    u=y[1, :]
    tstart = tstop
    if (i%4==1):
        dF = (u[2] - term0)*hbar*k/(2.0*tpulse)
    elif (i%4==2):

```

```

        dF = -(u[2] - term0)*hbar*k/(2.0*tpulse)
    else:
        dF = 0.0
    v = ((dF*2.0*tpulse)/mass + v)
    #print 'dF=',dF, 'v=',v
    FF = FF+dF
    term0 = u[2]
    if (i%4==0):
        result.append([phi, vstart/velfact, FF/(2.0*(float(i)
            /4)*forfact), i/4])
    return result

# tuple of all parallel python servers to connect with
ppservers = ()
#ppservers = ("10.0.0.1",)

#####

if __name__=='__main__':
    print 'Phasepoints:', phasepoints
    print 'pulse pairs:', npulses
    print 'delta_0:', d
    print 'Omega_0:', w

```

```

print 'Minimim value of velocity (m/s):', vmin
print 'Maximum value of velocity (m/s):', vmax
print 'Steps in velocity:', dkv
print 'The step size in velocity is:', vstep, 'm/s'

# Construct Job Server
if len(sys.argv) > 1:
    ncpus = int(sys.argv[1])
    # Creates jobserver with ncpus workers
    job_server = pp.Server(ncpus, ppservers=ppservers)
else:
    # Creates jobserver with automatically detected number of
    # workers
    job_server = pp.Server(ppservers=ppservers)
print "Starting pp with", job_server.get_ncpus(), "workers"

# Construct Jobs
arg_list = [(1,j, gamma,w0,d0,wm,dopoff,phistep,vmin,vmax,
    vstep,tpulse,npairs,k,ode_rtol,ode_atol,term0,hbar,mass,
    velfact,forfact)
    for l in range(1,dkv+2,1) for j in range(1,
    phasepoints+1,1)]
print "Number of Job: ", len(arg_list)

```



```

# Submit Jobs
deffuncs = (odeint, feval)
modules = ('numpy', 'scipy.integrate', )
start_time = time.time()
jobs = [job_server.submit(job_func, a, modules=modules,
    deffuncs=deffuncs) for a in arg_list]
results = [j() for j in jobs]
f = open("data20150121withgamma.txt", 'wb')
for i in range(len(arg_list)):
    f.write('#####\n')
    f.write('## l='+str(arg_list[i][0])+', j='+str(arg_list[i]
        ][0])+'\n')
    for r in results[i]:
        f.write(str(r[0])+', '+str(r[1])+', '+str(r[2])+', '+str(
            r[3])+'\n')
f.close()

print "Time elapsed: ", time.time() - start_time, "s"
job_server.print_stats()

```

Appendix D

R Code for Data Visualization

```
#@author: He Zhang

setwd("/home/arp/hezhang/SBU/data")
data = read.table("data15.dat", sep="")
#View(data)

#define parameters

npulse<-320
dkv<-0 #when only one velocity, we need to set dkv=0.
phasepoint<-25
wd = 5
step = 0.1

#####
```

```

nmat<-(wd/step+1)

#####

spl1<- dim(data)[1]/npulse #number of each group by npulse

data_sort_n <- data[order(data[,6]),]

spl_data=list()

for (j in 1:npulse) {
  m=(j-1)*spl1+1
  data_sort_n_pha<- data_sort_n[m:(m+spl1-1),]
  data_sort_n_pha<- data_sort_n_pha[order(data_sort_n_pha[,1]),]
  data_sort<-data_sort_n_pha[,3:5]
  spl_data<-append(spl_data, data_sort)}

#View(spl_data)

ob<-length(spl_data[[1]])
matpoint <- ob/(dkv+1)/phasepoint #6615/3/5=441

f=as.vector(1:matpoint)
force=as.vector(1:matpoint)

```

```

#T_force=0
Force=NULL
i=1
for (i in 1:npulse) {
  for (m in 1:(matpoint*(dkv+1))) {
    j<-c(1:ob) #we have 3 velocity
    f[m]<-sum(spl_data[[3*i]][ which (j%%(matpoint*(dkv+1))==m)])
    f[matpoint*(dkv+1)]<-sum(spl_data[[3*i]][ which (j%%(matpoint
      *(dkv+1))==0)])
  }
  force<-f/phasepoint
  #Table<-data.frame(spl_data[[3*(i-1)+2]], spl_data[[3*(i-1)+1]],
    force)
  Force<-append(Force, force)
}

save.image("~/hezhang/SBU/data/data15.RData")

#####

#plot
i=1
matrix<-matrix(Force[(1+matpoint):(matpoint+matpoint)], nrow=nmat,
  ncol=nmat, byrow = TRUE, dimnames = NULL)

```

```

rownames(matrix) <- spl_data[[3*(i-1)+1]][1:nmat]
colnames(matrix) <- spl_data[[3*(i-1)+2]][1:nmat]

heatmap(matrix, Rowv=NA, Colv=NA, col = heat.colors(200), scale="
  none", margins=c(8,10))

#####

name<-paste(c("pic15_"), 1:npulse, sep="")

# extract plot
for (i in 1:npulse) {
png(name[i], w=2000, h=2000)
matrix2<-matrix(Force[(1+(i-1)*matpoint):(i*matpoint)], nrow=nmat,
  ncol=nmat, byrow = TRUE, dimnames = NULL)
rownames(matrix2) <- spl_data[[3*(i-1)+1]][((i-1)*nmat+1):(i*nmat)
  ]
colnames(matrix2) <- spl_data[[3*(i-1)+2]][((i-1)*nmat+1):(i*nmat)
  ]
heatmap(matrix2, Rowv=NA, Colv=NA, col = heat.colors(200), scale="
  none", margins=c(8,10))
dev.off()
}

```

```
#####
#dynamic frame
install.packages("grDevices")
library(animation)
library(grDevices)
opt = ani.options(interval = 0.2, nmax = npulse, ani.type = "png"
)

#save gif
install.packages("animation")
library(animation)

saveGIF(
for (i in 1:npulse){
matrix1<-matrix(Force[(1+(i-1)*matpoint):(i*matpoint)],nrow=
nmat,ncol=nmat,byrow = TRUE,dimnames = NULL)
rownames(matrix1) <- spl_data [[3*(i-1)+1]][((i-1)*nmat+1):(i*
nmat)]
colnames(matrix1) <- spl_data [[3*(i-1)+2]][((i-1)*nmat+1):(i*
nmat)]
heatmap(matrix1, Rowv=NA, Colv=NA, col = heat.colors(200),
scale="none", margins=c(8,10))
ani.pause() ## pause for a while ('interval')
```

```

},
movie.name = "pic11.gif"
)

#generate html with button
saveHTML(
  for (i in 1:npulse){
    matrix1<-matrix(Force[(1+(i-1)*matpoint):(i*matpoint)],nrow=
      nmat,ncol=nmat,byrow = TRUE,dimnames = NULL)
    rownames(matrix1) <- spl_data[[3*(i-1)+1]][((i-1)*nmat+1):(i*
      nmat)]
    colnames(matrix1) <- spl_data[[3*(i-1)+2]][((i-1)*nmat+1):(i*
      nmat)]
    heatmap(matrix1, Rowv=NA, Colv=NA, col = heat.colors(200),
      scale="none", margins=c(8,10))
    ani.pause() ## pause for a while ('interval')
  }
, single.opts = "'controls': ['first', 'previous', 'play', 'next',
  'last', 'loop', 'speed'], 'delayMin': 0", autoplay = TRUE,
  interval = 0.5,
,htmlfile = "data13.html", ani.height = 800, ani.width = 800,
title = "force_map")

#change 4 name: initial datafile, name, savegif, saveHTML

```