CHARACTERIZATION OF MICRO-CYLINDERS IN A LASER TRAP: A THEORETICAL AND EXPERIMENTAL INVESTIGATION

By

Mike Ashman

Department of Physics and Astronomy

This thesis is submitted in partial fulfillment of

the requirements of the Degree of

Master of Science in Physics

School of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

1998

0-612-46465-2

Abstract

In this thesis the characterization of micro-cylinders in a laser trap is presented. Using the modified ray optics theory a sophisticated computer program has been designed. This program models objects in and near the minimum waist region of a laser beam. The equations of torque, moment of inertia for a cylinder with flat or spherical end-caps, angular acceleration and the damping factor due to the medium are derived and implemented into the program. With the program the orientation of a stable trapped micro-cylinder is found for different lengths. The optimal design of a cylinder to produce the maximum amount of torque is determined. The continuous rotation of a micro-cylinder using multiple laser beams is also examined. Using three different experimental set-ups, the "Top-Down", "Bottom-Up" and the "Horizontal" designs, the orientation of the cylinders trapped in the laser beam is in good agreement with theoretical results. This information should aid in the design of micro-motors using cylindrical shaped objects as posts and micro-rotors based on a multiple cylinder design as the components. In the future, the design of micro-systems should be possible which will have applications in such fields as the medical, chemical, aerospace and any other industry where a decrease in the size of equipment is required.

Acknowledgements

I would like to express thanks to my supervisor Dr. Robert Gauthier for his extended patience while sharing his knowledge and expertise over the past few years. It is from his teachings that I have learned the dedication and professionalism required in the process of scientific experimentation. I would also like to extend my appreciation to my committee members Dr. Rizwan Haq and Dr. Marcel Leach for their time in reviewing this thesis. I would like to salute Ivano Renelli and Willard Desjardins for their technical expertise. The experiments in this thesis would not have been possible if it were not for NRC (The National Research Council) which allowed me to use their facilities. To John Theal, Dan Moulton, Jason Hewitt, Cindi Kilkenny and John Cowan, my friends, who have supported me through the best of times and the worst of times, thank-you for your support. I would like to thank my parents Dorothy Burt-Gerrans and Glenn Ashman for their love and continued support through out my life. Finally, I would like to dedicate this work to Valerie McShane who stood by me for the past two years and kept telling me that this day would finally come.

Contents

List of Figures

LIST OF TABLES

Chapter 1

Introduction

According to the Canadian Dictionary of the English Language the definition of trapping is "to catch in or as if in a trap; ensnare."[1] In the past 30 years a new type of trap has appeared called an "Optical Trap". An optical trap can be defined as an instrument that utilizes laser light to capture microscopic particles.

Johannes Kepler in 1619 [2], had suggested that light could have a mechanical effect to explain why the tails of comets entering our solar system always pointed away from the sun. The theory behind this mechanical effect was studied extensively. It was two well-known physicists, Maxwell and Einstien, who made major contributions to the explanation of this mechanical effect. A theory was in place and in 1901 the existence of field momentum was confirmed experimentally by P. N. Lebedev[3]. Lebedev was able to show that the field momentum existed in the form of the pressure of light which is now termed 'radiation pressure'. Radiation pressure is defined as "the momentum associated with a photon [which] can be transferred to objects of finite mass, giving rise to a force and causing mechanical motion."[4]

It is the radiation pressure, which is fundamental to the operation of the laser trap, which led A. Ashkin [5] to build the first levitation device. Radiation pressure is a small force in the range of 10^{-12} N for a laser of 100 mW, which is usually obscured, by other forces such as radiometric forces. When the particle to be levitated is relatively transparent and submerged into a relatively transparent medium the radiometric forces are overcome and the radiation forces become the dominate force which then allows the object to be levitated. Ashkin reported the first levitation in 1970. By using two opposing laser beams, it was possible to use the principles of levitation to capture particles in the central region of the two beams. By using four beams and a magnetic field it was now possible to look at ultra low temperatures of atoms. The work in this field produced the first observation of a Bose-Einstien condensate in a dilute atomic gas and the development of the first rudimentary atom laser [2]. The 1997 Nobel Prize was awarded to S. Chu, C. Cohen-Tannoudji, and W. D. Phillips for these achievements.

In 1986 Ashkin *et. al.* [6] found that it was possible to trap particles with just one laser beam, now called "optical tweezers". This great feat opened the door for many new possibilities in different fields of science. In biology the tweezers can manipulate cells [7] for such events as cell-to-cell adhesion. In physics and engineering, micro parts could now be built piece by piece instead of being etched out layer by layer in a substrate to create the micro-mechanical systems. These micro parts were shown to rotate due to the radiation pressure by E. Higurashi *et. al.*[8]. Micro-rotors could now be created.

In the development of micro-machines, posts are required to fix the rotor in place and still permit rotation. These posts, which are cylindrical in shape, also have to be placed before the rotors can be attached. The trapping and continuous rotation of cylinders is the basis of this thesis and has not been reported in previously published works. With the help of a new computer program and three different experimental set-ups, the cylinder's orientation properties in a focused laser beam will be discussed. The ability to translate and rotate the cylinder by a single beam will be experimentally demonstrated. Along with the cylinder's orientation, the continuous rotation of the cylinder by multiple beams and the rotation of a rotor by a single beam will be examined. Theoretically the rotor is designed from four crossing cylinders at 45° to its neighbour. The simulations will give a better understanding of how cylindrically shaped objects behave in the tweezers and help in the design of micro-rotors and micro-systems of the future. The results of this thesis can also be applied in other fields such as biology where several organisms have an overall cylindrical shape.

Chapter 2

Theory of Laser Trapping in the Ray (Mie) Regime

In this chapter, the modified ray optics theoretical approach of laser trapping is discussed. A sphere will be the object of discussion in the beam but any shape can be used provided the features of the object are larger than the wavelength, λ_o. The expressions are first developed in a 2-D space for the basic understanding of interactions between the beam and sphere. The expressions are then generalized to 3-D space for any object or beam. The computer modeling of the laser trapping experiments is built on the expressions for the 3-D space.

2.1 Two Dimensional Ray Theory of Laser Trapping of a Sphere

A sphere suspended in a liquid and illuminated by an off center continuous wave laser beam will be examined first. In 1970 A. Ashkin demonstrated that it was possible to use the forces produced from radiation pressure to levitate micro-particles once the thermal effects were overcome. One of the requirements for levitation is that the medium index of refraction must be lower than the index of refraction of the sphere. By examining the refracted or reflected photons at the sphere's interface, it will be seen that the sphere is drawn towards the laser's central axis. The sphere will also be drawn into the minimum waist region of the beam when the beam is highly focused.

Figure 2.1.1 depicts photons interacting at two different positions on the sphere. At position (a) the sphere is being struck with more photons than at position (b) due to the sphere's top being in a more intense part of the beam. The beam's intensity is indicated by the left graph of figure 2.1.1 and has a lowest order mode Gaussian profile.

It was A. H. Compton [9] who first argued that if photons carry energy then they must also carry momentum. This can be easily seen from the equation for energy below:

$$E^2 = (pc)^2 + (mc^2)^2 \tag{2.1.1}$$

where E is the energy and p, c, m are the momentum, speed of light and the mass respectfully. Since photons have no mass the above equation becomes:

$$E = pc \tag{2.1.2}$$

giving the well-known energy-momentum expression for a massless particle like the photon.

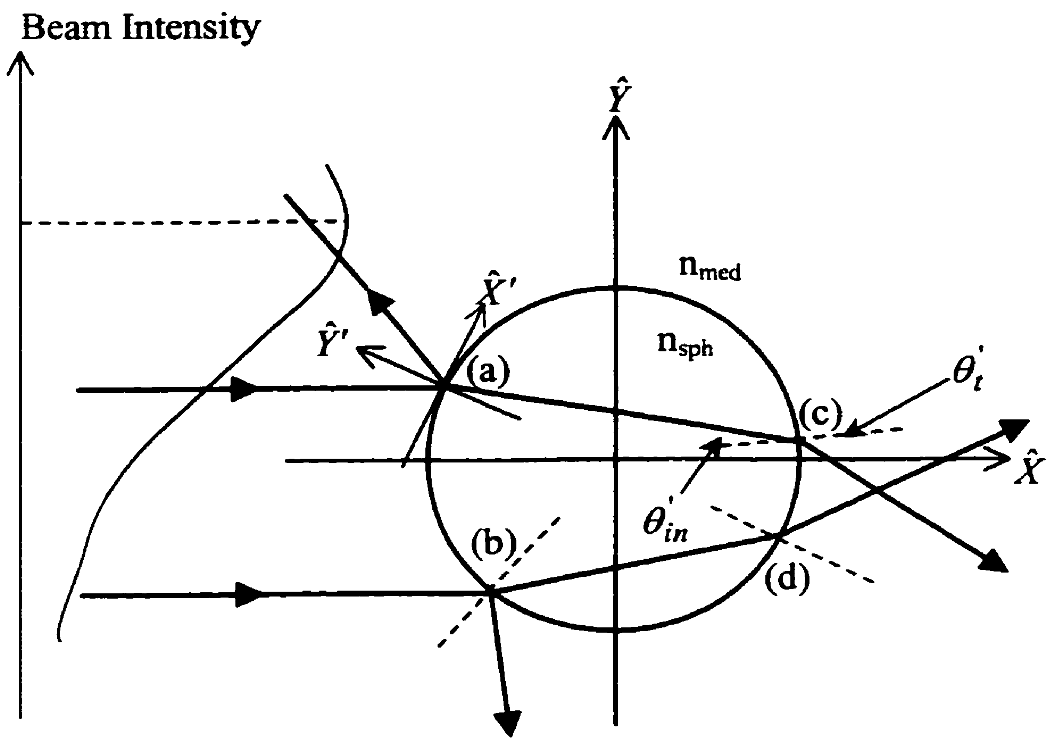


Figure 2.1.1: TWO PARALLEL INCIDENT RAYS UPON A MICRO-SPHERE.

From the change in momentum, $d\vec{P}$, of two parallel rays (photon streams) entering the sphere at points (a) and (b), it will be shown that the sphere will have a net change in momentum directed towards the more intense part of the beam.

Figure 2.1.2 is an enlarged, rotated, diagram of the interaction point (a) of figure 2.1.1.

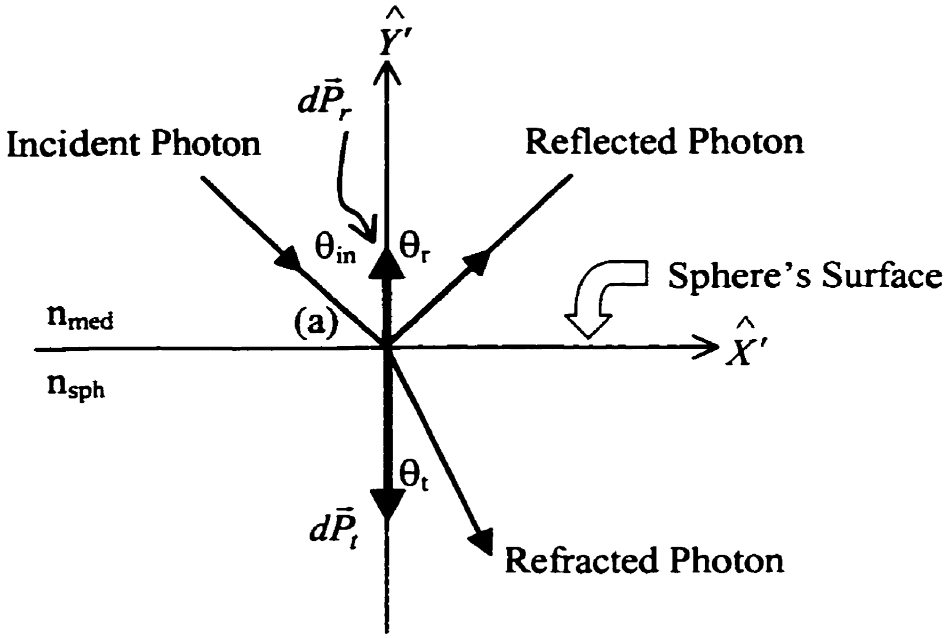


Figure 2.1.2: REFRACTING AND REFLECTING PHOTONS AT THE SURFACE OF THE SPHERE.

The index of refraction outside, and inside, the sphere are defined as n_{med} and n_{sph} respectively. The normal to the sphere is directed along the $\hat{Y}'$ axis. The angles of incidence, reflection and refraction are defined as θ_{in}, θ_r, and θ_t where they are related through the following: law of reflection $\theta_r=\theta_{in}$; law of refraction $n_{in}\sin\theta_{in}=n_t\sin\theta_t$; with $n_{in}=n_{med}$ and $n_t=n_{sph}$. The symbols θ_r, θ_{in}, n_t and n_{in} represent the angle of reflection, incidence angle, the transmitted index of refraction and the incident index of refraction. The momentum vector of the incident photon is defined as:

$$\vec{P}_{in} = \hbar \vec{k}_{in} = \hbar \frac{2\pi n_{in}}{\lambda_o} \qquad (2.1.3)$$

where $\hbar$ is Plank's constant, h, divided by 2π, $\vec{k}_{in} = (k_{X'}, k_{Y'})$, in the 2-D space, is the medium dependent propagation vector or wave vector and λ_o is the free space wavelength. Expressed in Cartesian coordinates the propagation vector becomes:

$$\vec{k}_{in} = \frac{2\pi n_{in}}{\lambda_o}\left(\sin(\theta_{in})\hat{X}' - \cos(\theta_{in})\hat{Y}'\right) \qquad (2.1.4)$$

The momentum of the incident photon is then defined as:

$$\vec{P}_{in} = \hbar \frac{2\pi n_{in}}{\lambda_o}\left(\sin(\theta_{in})\hat{X}' - \cos(\theta_{in})\hat{Y}'\right) \qquad (2.1.5)$$

The momentum of the transmitted photon is defined as:

$$\vec{P}_{t} = \hbar \frac{2\pi n_{t}}{\lambda_o}\left(\sin(\theta_{t})\hat{X}' - \cos(\theta_{t})\hat{Y}'\right) \qquad (2.1.6)$$

The momentum difference between the refracted and incident photon is:

$$d\vec{P}_{t} = \hbar \frac{2\pi}{\lambda_o}\begin{pmatrix} [n_t \sin(\theta_t) - n_{in}\sin(\theta_{in})]\,\hat{X}' - \\ [n_t \cos(\theta_t) - n_{in}\cos(\theta_{in})]\,\hat{Y}' \end{pmatrix} \qquad (2.1.7)$$

Using Snell's law: $n_t \sin(\theta_t) = n_{in}\sin(\theta_{in})$

the component along the $\hat{X}'$ direction is zero and equation (2.1.7) reduces to:

$$d\vec{P}_t = -\hbar\frac{2\pi}{\lambda_o}\left[n_t\cos(\theta_t) - n_{in}\cos(\theta_{in})\right]\hat{Y}' \qquad (2.1.8)$$

With the index of refraction of the sphere or any object greater than the index of refraction of the surrounding medium, the change in momentum of the photons is pointing in the negative $\hat{Y}'$ direction. This is opposite to the direction of the normal as indicated by the thick arrow of figure 2.1.2.

Applying a similar analysis for the reflected photons of momentum expressed as:

$$\vec{P}_r = \hbar\frac{2\pi\, n_r}{\lambda_o}\left(\sin(\theta_r)\hat{X}' + \cos(\theta_r)\hat{Y}'\right) \qquad (2.1.9)$$

the change in momentum of the reflected photon is:

$$d\vec{P}_r = \hbar\frac{2\pi\, n_{in}}{\lambda_o}\left(\left[\sin(\theta_r') - \sin(\theta_{in})\right]\hat{X}' + \left[\cos(\theta_r') - (-\cos(\theta_{in}))\right]\hat{Y}'\right) \qquad (2.1.10)$$

By applying the law of reflection, $\theta_r = \theta_{in}$, and $n_r = n_{in}$ the $\hat{X}'$ component cancels leaving only the $\hat{Y}'$ component. The simplified expression for change in momentum of the reflected photons is:

$$d\vec{P}_r = \hbar\frac{4\pi\, n_{in}}{\lambda_o}\cos(\theta_{in})\hat{Y}' \qquad (2.1.11)$$

which is pointing along the positive $\hat{Y}'$ direction as indicated by the thick arrow of figure 2.1.2.

When the photon exits the sphere at point (c) of figure 2.1.1 they will experience a change in momentum. This change in momentum is:

$$d\vec{P}_t = \frac{\hbar 2\pi}{\lambda_o}\left(n_{in}\cos\left(\theta_{in}'\right) - n_t\cos\left(\theta_t'\right)\right)\hat{Y}' \qquad (2.1.12)$$

where in this case n_t and n_{in} are now the indices of refraction for the medium and sphere respectively.

The force that is acting on the sphere can be found from the time rate of change of the momentum of the sphere (Newton's second law). The $d\vec{P}$ component is obtained using Newton's third law and is equal to, but opposite in sign, to the time rate of change of the photons momentum.

$$\vec{F} = \frac{d\vec{P}}{dt} \qquad (2.1.13)$$

The force directions at the point's (a) to (d) for a low reflecting sphere are as follows:

(a): Along the $-\hat{X}$ axis and along the $+\hat{Y}$ axis.

(b): Along the $-\hat{X}$ axis and along the $-\hat{Y}$ axis.

(c): Along the $+\hat{X}$ axis and along the $+\hat{Y}$ axis.

(d): Along the $+\hat{X}$ axis and along the $-\hat{Y}$ axis

The intensity of the laser beam is greater at the point (a) than at point (b) of figure 2.1.1, which implies that there are more photons per second passing point (a). Thus the force from the refraction and reflection will be greater at points (a) and (c) than at points (b) and (d). The sphere experiences a net force directed towards the more intense part of the beam.

In the next section the ray model is expanded to 3-D and the restriction of parallel incident rays on the object is removed. This examination of the 3-D case is also used in order to develop the complete expressions required for the computer model.

2.2 Three Dimensional Approach to Laser Trapping

A three-dimensional computer simulation program of a particle subjected to a focused laser beam requires a modified ray theory and was developed by R. C. Gauthier [10]. It is possible to use the conventional scattering theory [11] or the pseudo ray optics model [12] to calculate the forces of radiation pressure but each has its own limitations. The scattering theory is readily applicable to calculate the forces present on a spherical particle in the traps but is impractical to apply to irregularly shaped objects. The pseudo ray optics model can be used when the object dimensions are larger than the wavelength of light but neglects many important properties of trapping systems. The modified ray theory can account for the beam wavefront curvature, polarization, reflection-transmission probabilities and the conservation of momentum.

The model begins with a Gaussian mode profile laser beam, propagating along the z-axis, incident onto the lower surface of a micro-object. The intensity of the laser beam is defined as [13]:

$$I(x, y, z) = \frac{2P_{laser}}{\pi W_z^2} \exp\left[\frac{-2\left(x^2 + y^2\right)}{W_z^2}\right] \tag{2.2.1}$$

where P_{laser} is the total power of the laser, W_z is the beams waist measured a distance z from the minimum waist location and is related to the minimum waist W_o and Rayleigh range z_o through the equation:

$$W_z = W_o \sqrt{1 + \left(z/z_o\right)^2} \tag{2.2.2}$$

The photons from the beam passing through an arbitrary point (x_s, y_s, z_s) can be approximated as a point ray with direction cosines (l, m, n). This point ray has properties, which are different from the ray defined in the geometrical context of optics. These properties can be found in Appendix A1. The initial point (x_s, y_s, z_s) in this plane and direction cosine (l, m, n) for the photons ensures that no one photon is traced more than once when they interact with the object. In treating the photon's of the entire beam an initial (x, y) reference plane perpendicular to the z-axis is selected.

Two methods of obtaining the intercept point of the irregular object have been developed. The first method of locating the point of intersection, by the photons, with any surface of the micro-object is obtained by finding the parameter *t*, which satisfies the following set of equations:

$$\begin{aligned} x(t) &= x_S + lt \\ y(t) &= y_S + mt \\ z(t) &= z_S + nt \end{aligned} \tag{2.2.3}$$

and at the same time the point *(x(t), y(t), z(t))* must also be a point on the surface of the object. The initial direction cosine of the photons is chosen as (0, 0, 1), z directed photons. At the intercept point the direction cosines for the photons are calculated from the spherical nature of the wavefront. The wavefront radius of curvature, see figure 2.2.1, can be obtained by solving the two next equations for *R* and *Δz*:

$$\begin{aligned} R &= \sqrt{(R-\Delta z)^2 + x(t)^2 + y(t)^2} \\ R &= \left(z(t)+\Delta z\right)\left[1+\left(\frac{z_o}{z(t)+\Delta z}\right)^2\right] \end{aligned} \tag{2.2.4}$$

From these equations the location of the center of curvature for the wavefront, C=(c_x, c_y, c_z), can be calculated.

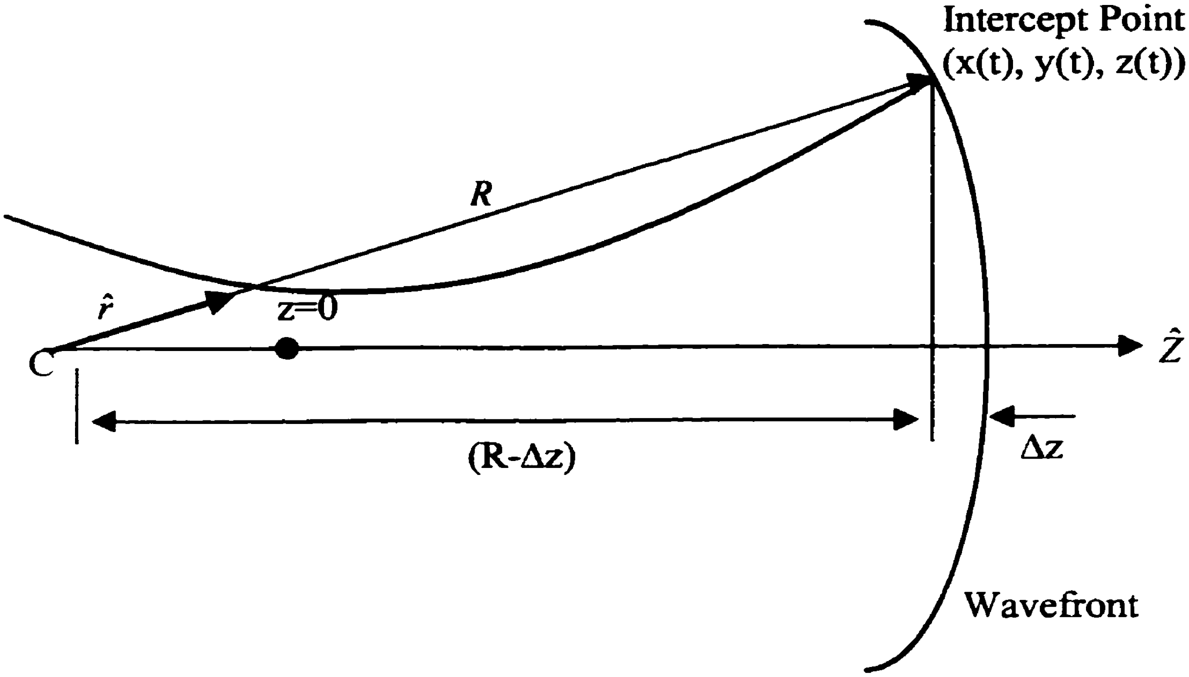


Figure 2.2.1: AT THE POINT OF INCIDENCE THE INCIDENT PHOTON DIRECTION COSINES ARE GIVEN BY THE DIRECTION COSINES OF THE RADIAL UNIT VECTOR $\hat{r}$, WHICH POINTS FROM THE CENTER OF CURVATURE OF THE WAVEFRONT TO THE POINT OF INCIDENCE.

The direction cosines (l(t), m(t), n(t)) of the photon's intercept point are taken as directed along the radius vector $\hat{r}$ and are computed from:

$$l = \pm (x(t) - c_x)/R$$
$$m = \pm (y(t) - c_y)/R \qquad (2.2.5)$$
$$n = (z(t) - c_z)/R$$

If the intercept point is located before the beam has propagated through the minimum waist then the negative sign is used.

It is still possible for photons to intersect the micro-object even if the stream of photons with initial direction cosines (0, 0, 1) does not. This is possible since the wavefront curvature can direct the photons onto the sidewalls of the object. If this occurs the intercept point and photon direction cosines are obtained by the process of projecting the initial photon trajectory and surfaces in the (x, y) plane then finding the intercept of these trajectories with these surfaces. The spherical wavefront for the photon direction cosines is then included. The last step is to reverse project the intercept point. For these direction cosines to be acceptable they must correspond to photons pointing towards the inside of the micro-object's body.

The second method also uses equation 2.2.1, 2.2.2 and 2.2.3 and the equation for the surface directly to find the intercept. The initial point (x_s, y_s, z_s) is selected in the reference plane. At this point the direction cosines are directly determined for the wavefront radius of curvature. The parameter t is incremented slightly giving a second point. The ray is taken as a straight line between these two points. The computer program can then determine if the ray (between these two points) intersects any object's surface. If no, the process is repeated as the ray is traced through the trap system. If yes, the parameter t is decremented and the process is repeated. The parameter t can be subdivided as many times as desired with each subdivision increasing the ray parameters accuracy at the point of intercept. This technique is somewhat simpler to implement that the first presented but requires longer computational time but, and most importantly, is applicable to all trap system designs. The computation routines now utilize the second technique exclusively.

After the intercept point has been obtained on a surface, a momentum vector can be determined which characterizes each individual photon:

$$\vec{P} = \hbar\vec{k} = \frac{2\pi n}{\lambda_o}\left(l(t)\hat{x} + m(t)\hat{y} + n(t)\hat{z}\right) \qquad (2.2.6)$$

where $(\hat{x}, \hat{y}, \hat{z})$ are unit vectors and n is the index of refraction of the medium. The element of force on the surface can be computed from the time rate of change of the momentum of the incident photons as they are reflected and refracted at the intercept point. The radiation pressure force on the micro-object is generated from the vector sum of all the force elements for all the interacting photons.

$$\vec{F} = \sum_{\substack{\text{all interacting} \\ \text{photons}}} \frac{d\vec{P}}{dt} \qquad (2.2.7)$$

The incident, reflected and refracted photon direction cosines will be defined as (l_i, m_i, n_i), (l_r, m_r, n_r), and (l_t, m_t, n_t). The momentum transferred **to the surface** from the reflected photons is defined as:

$$d\vec{P}_r = \frac{hn_r}{\lambda_o}\left[(l_i - l_r)\hat{x} + (m_i - m_r)\hat{y} + (n_i - n_r)\hat{z}\right] \tag{2.2.8}$$

From the refracted photons the change in momentum is:

$$d\vec{P}_t = \frac{hn_{in}}{\lambda_o}\left[(l_i - n_{rel}\, l_t)\hat{x} + (m_i - n_{rel}\, m_t)\hat{y} + (n_i - n_{rel}\, n_t)\hat{z}\right] \tag{2.2.9}$$

where n_{rel} is equal to the ratio of the output index after refraction to the input index before refraction $(n_{rel} = n_t / n_{in})$. Included in Appendix A.2 are the details for obtaining the incident, reflected and refracted direction cosines and the surface normal when the point of intercept is given. It is known that the refracted momentum contribution with respect to the surface will point from the region of high index of refraction to the region of lower index of refraction.

To calculate the net force it is considered that there are N_i photons with an arbitrary polarization incident per unit time, dt, at the point of intercept. There will be a fraction of photons that will refract, R_{ave}, and a fraction that will reflect, $(1-R_{ave})$. This implies that the net force of equation 2.2.7 can be rewritten as the sum over all points of intercept:

$$\vec{F} = \sum_{\substack{all\ points\\ of\ intercept}} d\vec{F}_i = \sum_{\substack{all\ points\\ of\ intercept}} N_i \left[R_{ave}\, d\vec{P}_r + (1 - R_{ave}) d\vec{P}_t \right] \qquad (2.2.10)$$

The number of photons incident per second at the point of intercept centered on an area dA is related to the optical intensity (equation 2.2.1) at the intercept point by:

$$N_i = \frac{I(x,y,z)}{hc} \lambda_o \; dA \qquad (2.2.11)$$

By taking the cross product of a radial vector and the force element vector the torque element generated by the interacting photons at a point of incidence is obtained. Usually the geometrical center of the object is selected as the radial vector's origin but other reference points may be chosen. By summing all of the torque elements for all points of intercept for the photons the total torque, with respect to the reference point, is obtained.

$$\vec{\tau} = \sum_{\substack{all\ points\\ of\ interception}} d\vec{\tau}_i = \sum_{\substack{all\ points\\ of\ interception}} \vec{r} \times d\vec{F}_i \qquad (2.2.12)$$

It is equations 2.2.10 and 2.2.12 that are the key equations used in the levitation and trapping modeling calculations.

This concludes the chapter on the pseudo ray optics theory of laser trapping. An understanding on laser trapping principles is required before a researcher can interpret the observations seen in the laboratory. The next two chapters discuss related concepts such as the moment of inertia and rotation of an object due to the presence of a torque on a micro-object.

Chapter 3

Torque

In everyday life the world population is constantly turning objects which range from door handles to car keys to screwdrivers. To make a bicycle move the chain must impart a force onto the rear sprockets which makes the tire rotate. This act of turning, or rotating, the object requires that a tangential force (torque) be applied to it. In this chapter the details of torque applied to an object will be discussed. In the first section the expressions required to compute the torque on an object are derived. They are incorporated into the computer model. In the second section the dynamics of rotating objects and in particular cylinders, are examined.

3.1 Derivation of the Torque Equation

One of the final pieces to be added to the working version of the computer simulation is an algorithm suitable for rotating an object when a torque is present. To help visualize the technique developed, figure 3.1.1 shows an irregular object with its center off set with respect to the coordinate origin.

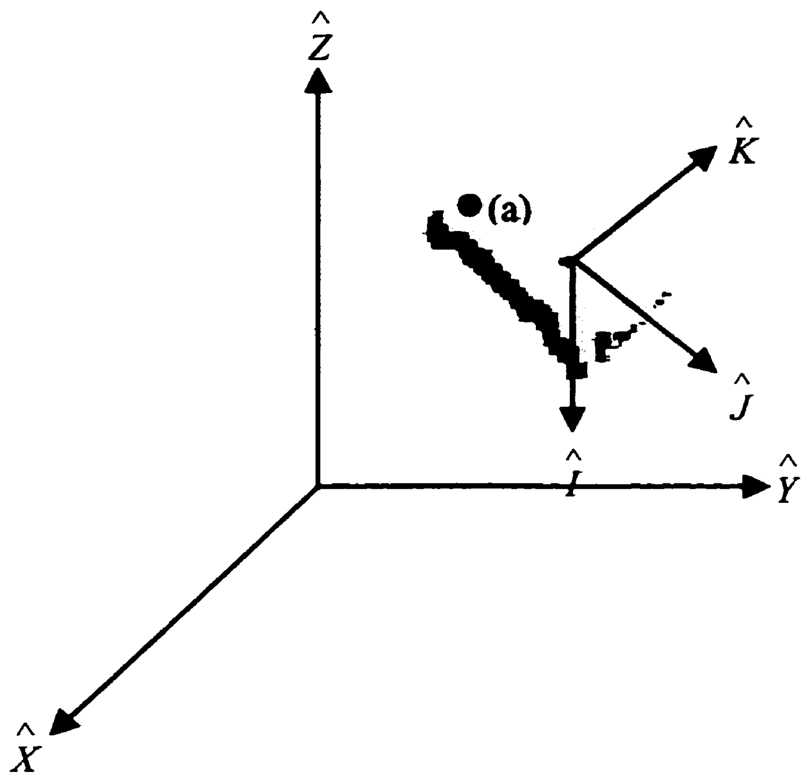


Figure 3.1.1: IRREGULAR SHAPED OBJECT WITH TORQUE AXIS OF $\left(\hat{I}, \hat{J}, \hat{K}\right)$ OFFSET FROM THE COORDINATE ORIGIN.

When there is a torque applied to a free object it will rotate. To obtain the new position of the point (a) after rotation requires that the coordinate axis be translated onto the torque vector axis. Using matrices to describe the mathematics, the initial translation is achieved by moving each coordinate point by a specific distance. This is written as [14]:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} \tag{3.1.1}$$

where X, Y, Z represent the original coordinate system, X', Y', Z' represent the final coordinate system position, and ΔX, ΔY, ΔZ is the distance the coordinate system translated. Figure 3.1.2 shows a picture of a translated axis.

Once the translation is completed the inital $\hat{Y}$ axis has to be aligned with the torque vector axis. This is accomplished by two rotation matrices [15]. The first matrix will rotate an angle of ϕ, around the $\hat{Z}'$ axis such that the $\hat{Y}'$ axis will be aligned properly with the torque vector. This is shown in figure 3.1.3. The calculation to achieve this is as follows:

$$\begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \tag{3.1.2}$$

To finish the alignment the axis will then be rotated an angle θ about the $\hat{Y}''$ axis (see figure 3.1.4):

$$\begin{bmatrix} X''' \\ Y''' \\ Z''' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} \qquad (3.1.3)$$

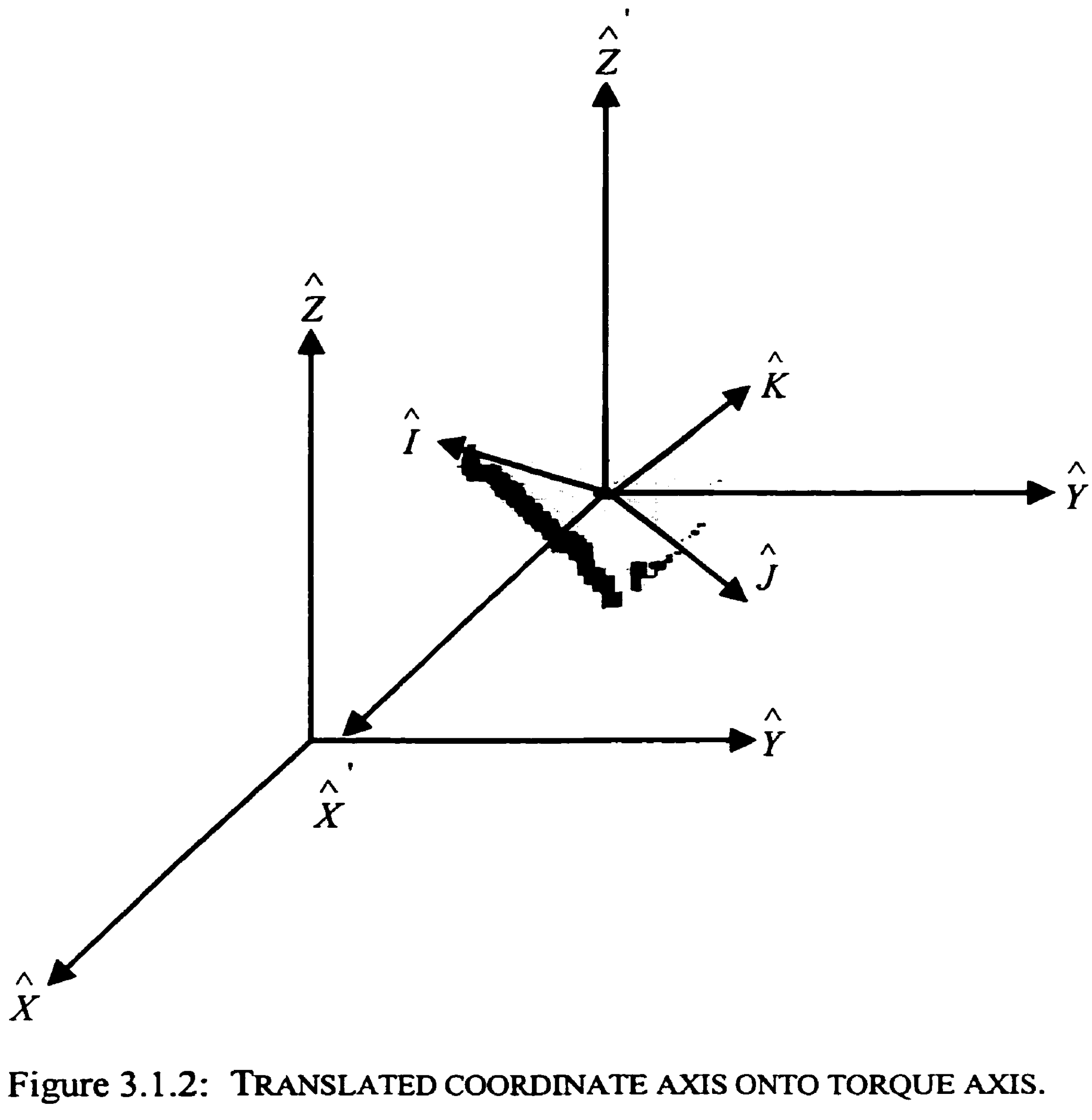


Figure 3.1.2: TRANSLATED COORDINATE AXIS ONTO TORQUE AXIS.

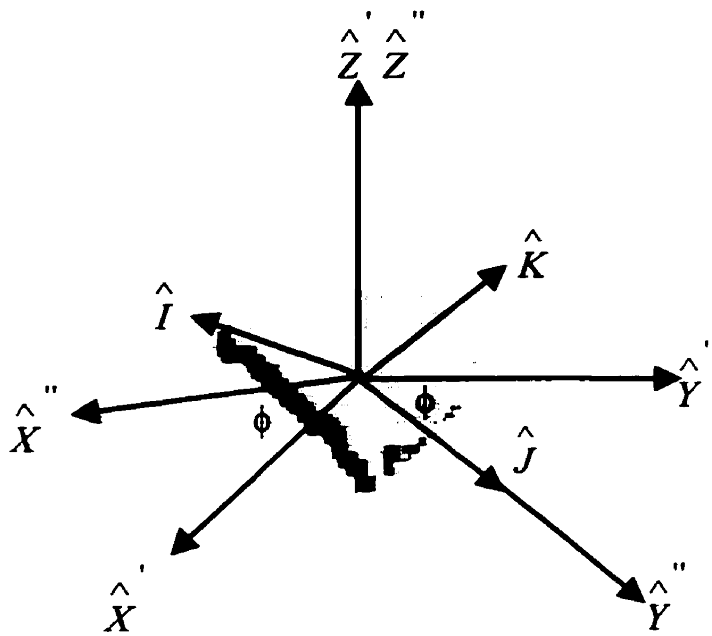


Figure 3.1.3: ROTATION ABOUT $\hat{Z}'$ AXIS.

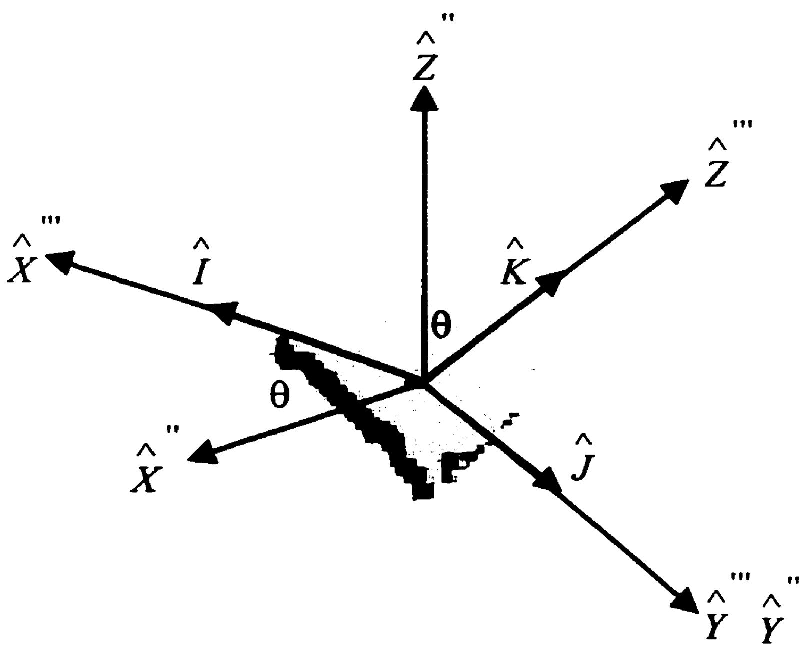


Figure 3.1.4: ROTATION ABOUT $\hat{Y}''$ AXIS.

When the matrices have been multiplied through the final product is known as the forward rotation matrix, which is given below.

$$FRM = \begin{bmatrix} \cos\theta\cos\phi(X-\Delta X)+\cos\theta\sin\phi(Y-\Delta Y)+\sin\theta(Z-\Delta Z) \\ \sin\phi(X-\Delta X)+\sin\phi(Y-\Delta Y) \\ \sin\theta\cos\phi(X-\Delta X)+\sin\theta\sin\phi(Y-\Delta Y)+\cos\theta\,(Z-\Delta Z) \end{bmatrix} \quad (3.1.4)$$

Once the $\hat{Y}$ axis is aligned with the torque vectors, the object can be rotated under the influence of the torque present. The torque matrix is defined as:

$$\tau = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1.5)$$

where the magnitude of the angle γ is related to the magnitude of the torque vector. After the object has been rotated by an angle γ under the influence of the torque, the object's new orientation must be referred back to the initial $\left(\hat{X},\hat{Y},\hat{Z}\right)$ coordinate system. The axis de-rotation matrix is the inverse of equation 3.1.2 multiplied by the inverse of equation 3.1.3 which yields:

$$RRM = \begin{bmatrix} \cos\phi\cos\theta & -\sin\phi & \cos\phi\sin\theta \\ \sin\phi\cos\theta & \cos\phi & \sin\phi\sin\theta \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (3.1.6)$$

To obtain the coordinate system matrix the forward, torque and de-rotation matrices must be multiplied together:

$$system = RRM \; \tau \; FRM \tag{3.1.7}$$

This produces:

$$system = \begin{bmatrix} \left\{ \begin{matrix} \left(c^2\phi\, c^2\theta\, c\gamma + c^2\phi\, s^2\theta + c\gamma\, s^2\phi\right)(X - \Delta X) + \\ \left(c\phi\, s\phi\left\{c^2\theta\, c\gamma + s^2\theta - c\gamma\right\} - s\gamma\, c\theta\left\{c^2\phi + s^2\phi\right\}\right)(Y - \Delta Y) \\ \left(c\phi\, c\theta\left\{s\theta - c\gamma\, s\phi\right\} + s\phi\, s\theta\, s\gamma\right)(Z - \Delta Z) \end{matrix} \right\} \\ \left\{ \begin{matrix} \left(c\phi\, s\phi\left(s^2\theta[1 - c\gamma]\right) + c\theta\, s\gamma\right)(X - \Delta X) \\ \left(s^2\phi\left\{c^2\theta\, c\gamma + s^2\theta\right\} + c\gamma\, c^2\phi\right)(Y - \Delta Y) \\ \left(s\theta\left\{s\phi\, c\theta[1 - c\gamma]\right\} - c\phi\, s\gamma\right)(Z - \Delta Z) \end{matrix} \right\} \\ \left\{ \begin{matrix} \left(s\theta\, c\phi\, c\theta[1 - c\gamma] + s\phi\, s\theta\, s\gamma\right)(X - \Delta X) \\ \left(s\theta\, s\phi\, c\theta[1 - c\gamma] + c\phi\, s\theta\, s\gamma\right)(Y - \Delta Y) \\ \left(c\gamma\, s^2\theta + c^2\theta\right)(Z - \Delta Z) \end{matrix} \right\} \end{bmatrix} \tag{3.1.8}$$

where c and s replace cos and sin. The last operation that is required is a coordinate translation by the addition of ΔX, ΔY, ΔZ to the $\left[\hat{X}, \hat{Y}, \hat{Z}\right]$ elements of the system matrix. The final results of this addition is:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \left\{ \begin{array}{l} \left(c^2\phi\, c^2\theta\, c\gamma + c^2\phi\, s^2\theta + c\gamma\, s^2\phi\right)(X - \Delta X) + \\ \left(c\phi\, s\phi\left\{c^2\theta\, c\gamma + s^2\theta - c\gamma\right\} - s\gamma\, c\theta\left\{c^2\phi + s^2\phi\right\}\right)(Y - \Delta Y) \\ \left(c\phi\, c\theta\{s\theta - c\gamma\, s\phi\} + s\phi\, s\theta\, s\gamma\right)(Z - \Delta Z) \end{array} \right\} + \Delta X \\ \left\{ \begin{array}{l} \left(c\phi\, s\phi\left(s^2\theta[1 - c\gamma]\right) + c\theta\, s\gamma\right)(X - \Delta X) \\ \left(s^2\phi\left\{c^2\theta\, c\gamma + s^2\theta\right\} + c\gamma\, c^2\phi\right)(Y - \Delta Y) \\ \left(s\theta\left\{s\phi\, c\theta[1 - c\gamma]\right\} - c\phi\, s\gamma\right)(Z - \Delta Z) \end{array} \right\} + \Delta Y \\ \left\{ \begin{array}{l} \left(s\theta\, c\phi\, c\theta[1 - c\gamma] + s\phi\, s\theta\, s\gamma\right)(X - \Delta X) \\ \left(s\theta\, s\phi\, c\theta[1 - c\gamma] + c\phi\, s\theta\, s\gamma\right)(Y - \Delta Y) \\ \left(c\gamma\, s^2\theta + c^2\theta\right)(Z - \Delta Z) \end{array} \right\} + \Delta Z \end{bmatrix} \tag{3.1.9}$$

Each major $\{\ \}$ set represents an element of the matrix. Also, X_2, Y_2, Z_2, represent the new coordinates of the point (a) after it has been rotated under the influence of a torque (see figure 3.1.5).

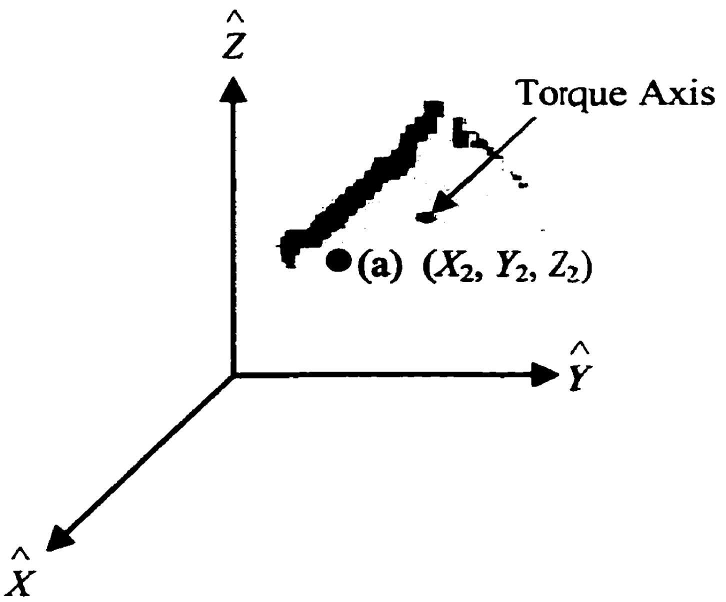


Figure 3.1.5: POSITION OF (a) AFTER TORQUE.

This was designed into a program in Visual Basic 5.0 for testing before it was slightly modified and implemented into the completed version of the simulation program. The code for the test program can be found in Appendix B.1.

3.2 Torque Required to Constantly Rotate an Object

When a tangential force is applied to a free object it will tend to rotate. This tangential force, as stated previously, is known as a torque. The simplest equation relating torque τ, inertia I and angular acceleration α for an object that is free to rotate about a fixed axis is [16]:

$$\tau = I\alpha \tag{3.2.1}$$

If the object is not allowed to move freely through its surrounding space then a damping term must be included in the above equation. An example of this would be a spinning cylinder in a viscous medium. It is also possible to have a restoring force on the object like a toy airplane's propeller wound up against an elastic band. When the propeller is manually rotated it winds up an elastic which, when released, acts as a restoring force that turns the propeller and allows the toy to fly. This energy storage and release technique can in principle be applied to micro-motors. Part of this thesis explores the concept of a continuously rotating cylinder in a slightly viscous medium. It is therefore, necessary to modify equation 3.2.1 to account for the other factors.

The modified torque equation then takes on the form of:

$$\tau = I\alpha + b\omega + k\gamma \tag{3.2.2}$$

where b is the damping factor and k the spring constant. The symbols ω and γ represent the angular velocity and the angular position. This equation can also be written as a second order differential equation:

$$\tau = I\frac{d\gamma^2(t)}{dt^2} + b\frac{d\gamma(t)}{dt} + k\gamma(t) \tag{3.2.3}$$

In the design of the cylinder-based micro-rotor, the cylinder is to be rotated at a constant velocity. The above equation (3.2.3) was first solved for $\gamma(t)$ then solved for $\frac{d\gamma(t)}{dt}$ which is the angular velocity.

To solve for $\gamma(t)$ the method of undetermined coefficients [17] was used. In the simulations there is no restoring force, "spring constant", required so k is equal to zero. Equation 3.2.3 is then written as:

$$I\frac{d^2\gamma(t)}{dt^2} + b\frac{d\gamma(t)}{dt} = \tau \tag{3.2.4}$$

$\frac{d^2\gamma(t)}{dt^2}$ and $\frac{d\gamma(t)}{dt}$ are then replaced with ζ^2 and ζ and all terms are divided by I:

$$\zeta^2 + \frac{b}{I}\zeta = \frac{\tau}{I} \qquad (3.2.5)$$

The homogeneous equation has roots of $\zeta=0$ and $\zeta = -\frac{b}{I}$ and produces a general solution:

$$\gamma(t)_g = C_1 e^0 + C_2 e^{-\frac{b}{I}t} \qquad (3.2.6)$$

where C_1 and C_2 are constants to be determined. The particular solution is assumed to be a polynomial of the same or higher degree and is assumed to be of the form:

$$\gamma(t)_p = At^2 + Bt + C \qquad (3.2.7)$$

where A, B, C are constants to be determined. The first and second derivative are:

$$\frac{d\gamma(t)_p}{dt} = 2At + B$$
$$\frac{d^2\gamma(t)_p}{dt^2} = 2A \qquad (3.2.8) \text{ and } (3.2.9)$$

and are substituting into equation 3.2.4.

$$2bAt + (2IA + bB) = \tau \qquad (3.2.10)$$

Equating the coefficients of like powers of t on both sides of the equation yields:

$$
\begin{aligned}
&t^0 : 2IA + bB = \tau \\
&t^1 : 2bA = 0 \\
&t^2 : C = 0
\end{aligned}
\tag{3.2.11}
$$

The t^1 expression gives A=0 since the damping factor is assumed not zero. Using this result in the t^0 expression gives $B=\tau/b$. The t^2 expression gives directly C=0. Substituting A, B, and C into equation 3.2.7 gives the particular solution.

$$\gamma(t)_p = \frac{\tau}{b}t \tag{3.2.12}$$

The final solution of $\gamma(t)$ is the addition of the general and particular solutions.

$$\gamma(t) = C_1 + C_2 e^{-\frac{b}{I}t} + \frac{\tau}{b}t \tag{3.2.13}$$

At time t=0 we will require that the cylinder be at rest and not rotating. This produces the two necessary initial conditions $\gamma(0)$=0 and $\frac{d\gamma(0)}{dt} = 0$ required to solve for the constants C_1 and C_2. The first derivative of 3.2.13 gives:

$$\frac{d\gamma(t)}{dt} = -C_2 \frac{b}{I} e^{-\frac{b}{I}t} + \frac{\tau}{b} \tag{3.2.14}$$

It is noted that the velocity, $\frac{d\gamma(t)}{dt} = 0$ at time t=0 implies that the object is considered to be at rotational rest before the laser is turned on. This implies that there is an initial start-up period before the object achieves a steady state velocity.

By applying the initial condition of the first derivative to equation 3.2.14 it is found that:

$$C_2 = \frac{I\tau}{b^2} \tag{3.2.15}$$

Substituting 3.2.15 into 3.2.13 and using the other initial condition the constant C_1 is found to be:

$$C_1 = -\frac{I\tau}{b^2} \tag{3.2.16}$$

Substituting 3.2.15 and 3.2.16 into 3.2.13 gives the complete solution of $\gamma(t)$.

$$\gamma(t) = \frac{I\tau}{b^2}\left(e^{-\frac{b}{I}t} - 1\right) + \frac{\tau}{b}t \tag{3.2.17}$$

The desired angular velocity equation is the derivative of equation 3.2.17:

$$\frac{d\gamma(t)}{dt} = \frac{\tau}{b}\left(1 - e^{-\frac{b}{I}t}\right) \tag{3.2.18}$$

Given the torque τ, the rotational rate can be determined provided the damping factor and the inertia are known. The determination of the cylinder's inertia about the pivot axis is the subject of the next chapter. The damping factor is obtained starting with the expression for a sphere as a guide. The solution for the case of $k \neq 0$ can be found in Appendix B.2.

For a sphere the damping is:

$$b = 6\pi \eta r \tag{3.2.19}$$

where η is the viscosity of the medium, r is the radius of the sphere. It was assumed that the rotating cylinder was made up of many smaller spheres placed in a line and rotated about the cylinder's pivot point. The sum of the damping values for all the spheres gives an estimate of the damping factor for the cylinder.

$$b = \sum_{i=1}^{N} 6\pi \eta r_i \tag{3.2.20}$$

where N is the number of spheres in the length. Instead of evaluating for discrete spheres, it is possible to integrate over the length.

$$b = \int_{-L}^{L} 6\pi \eta r l \; dl \tag{3.2.21}$$

where the limit of $-L$ to L is from one end of the cylinder to the other end of the cylinder passing through the center. The integration gives an expression, which enables the damping factor to be estimated:

$$b = 6\pi \eta r L^2 \tag{3.2.22}$$

This chapter presents the theory of the torque of a cylinder where the system matrix was determined to rotate an object under the influence of a torque. The angular velocity was then derived where the two constants b and I were introduced as the damping factor and the moment of inertia. The damping factor was discussed fully but the moment of inertia of a cylinder will be discussed in the next chapter due to its complexity.

Chapter 4

Moment of Inertia

To rotate an object such as a cylinder a torque must be applied to it. To calculate the degree of rotation (3.2.17) and the rotation rate (3.2.18) requires that the moment of inertia about the torque axis be known. In this chapter expressions for the moment of inertia of a cylinder with two types of ends will be derived.

4.1 Moment of Inertia

The general form of the moment of inertia for a discrete set of rotating masses is defined as [18]:

$$I = \sum m_i r_i^2 \tag{4.1.1}$$

where m_i is the mass of each element and r_i is the distance to the rotation axis for each element as shown in Figure 4.1.1. By replacing the sum with an integral,

$$I = \int r^2 dm \tag{4.1.2}$$

the inertia can be calculated for a continuous body as shown in Figure 4.1.2.

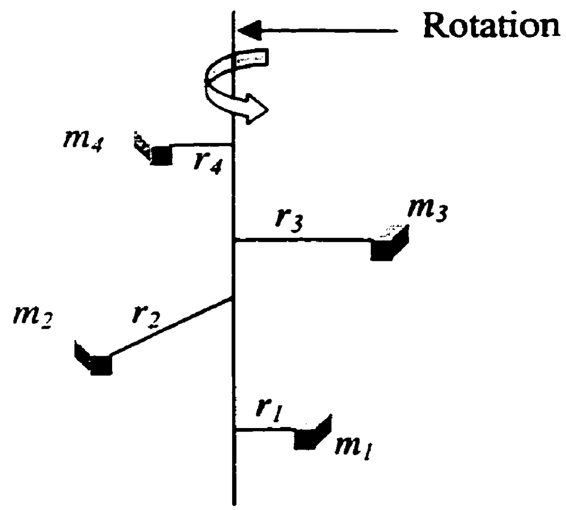


Figure 4.1.1: ROTATING ELEMENTS ABOUT A CENTRAL AXIS.

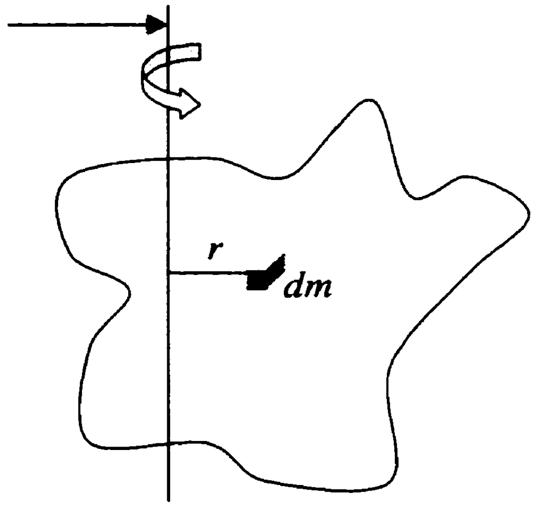


Figure 4.1.2: ROTATING BODY ABOUT A CENTRAL AXIS.

4.2 Moment of Inertia of a Cylinder

For a typical cylinder as shown in figure 4.2.1, the value of r was found to be:

$$r^2 = x^2 + \rho'^2 \sin^2 \phi \tag{4.2.1}$$

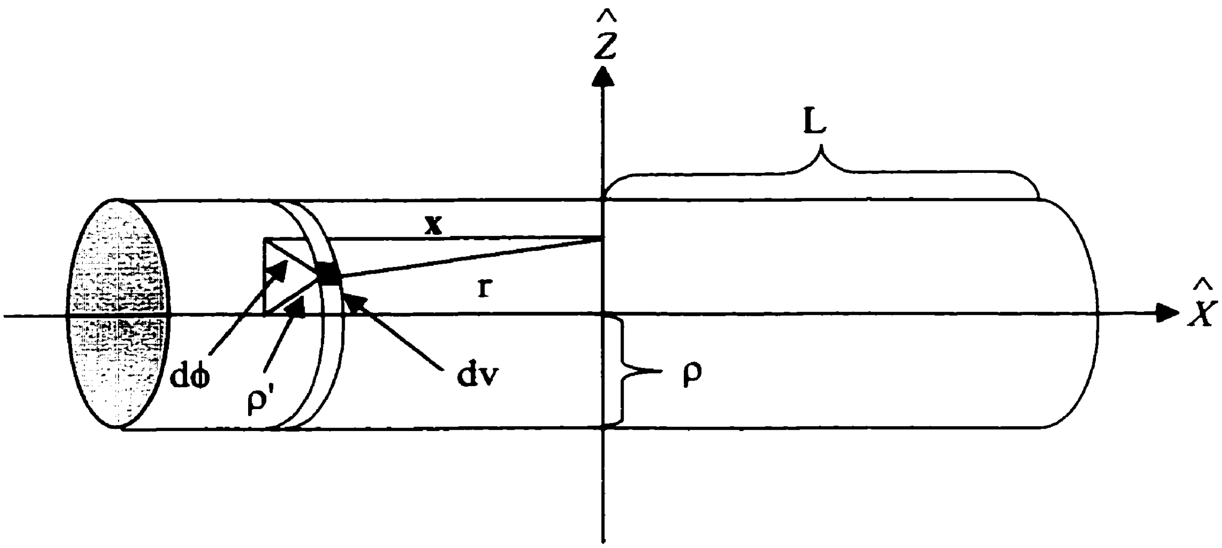


Figure 4.2.1: SIDE VIEW OF VOLUME ELEMENT IN CYLINDER.

where dv is the volume element, the $\hat{Z}$ axis is the rotation axis and the cylinder has flat ends. Figures 4.2.2 and 4.2.3 show an end-on and top-down view of the position of the volume element.

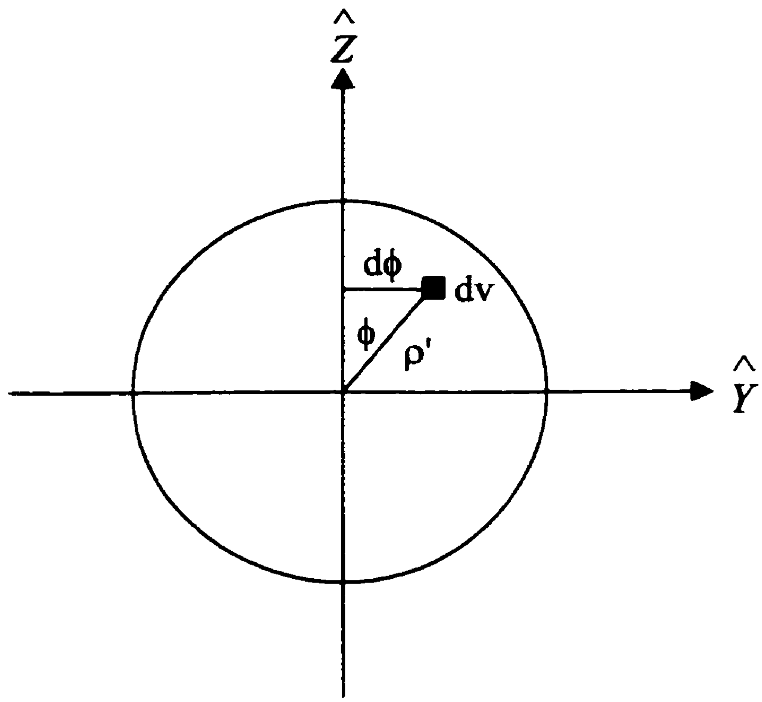


Figure 4.2.2: END VIEW OF VOLUME ELEMENT dv, IN CYLINDER.

The full equation for the moment of inertia then becomes:

$$I = \sigma \int_{-L}^{L} \int_{0}^{2\pi} \int_{0}^{\rho} \left(\rho' x^2 + \rho'^3 \sin^2 \phi \right) d\rho' \, d\phi \, dx \tag{4.2.2}$$

For convenience σ is used to represent the volume density. Integrating the above equation gives:

$$I = \sigma \pi \rho^2 \left[\frac{1}{12} L^3 + \frac{1}{4} \rho^2 L \right] \tag{4.2.3}$$

where $\sigma \pi \rho^2 L = M$ where M is the mass of the cylinder.

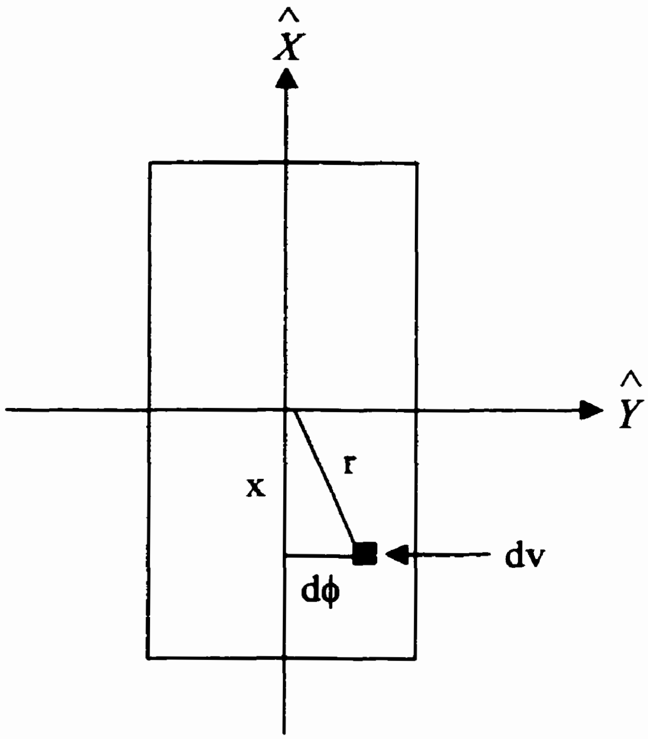


Figure 4.2.3: TOP-DOWN VIEW OF VOLUME ELEMENT IN CYLINDER.

The equation for the moment of inertia of a flat end cylinder is:

$$I = M\left[\frac{1}{12}L^2 + \frac{1}{4}\rho^2\right] \tag{4.2.4}$$

4.3 Moment of Inertia of End Caps (Case 1)

The moment of inertia about the rotation axis of the end-caps must be included. There are two cases to consider; the radius of the end-caps is less than half of the cylinder length or the radius is greater than half of the cylinder's length. This section deals with the first case of the radius being less than half the cylinder length. The end-caps that are used in the simulation are hemispheres or fractions of a hemisphere as shown in Figure 4.3.1.

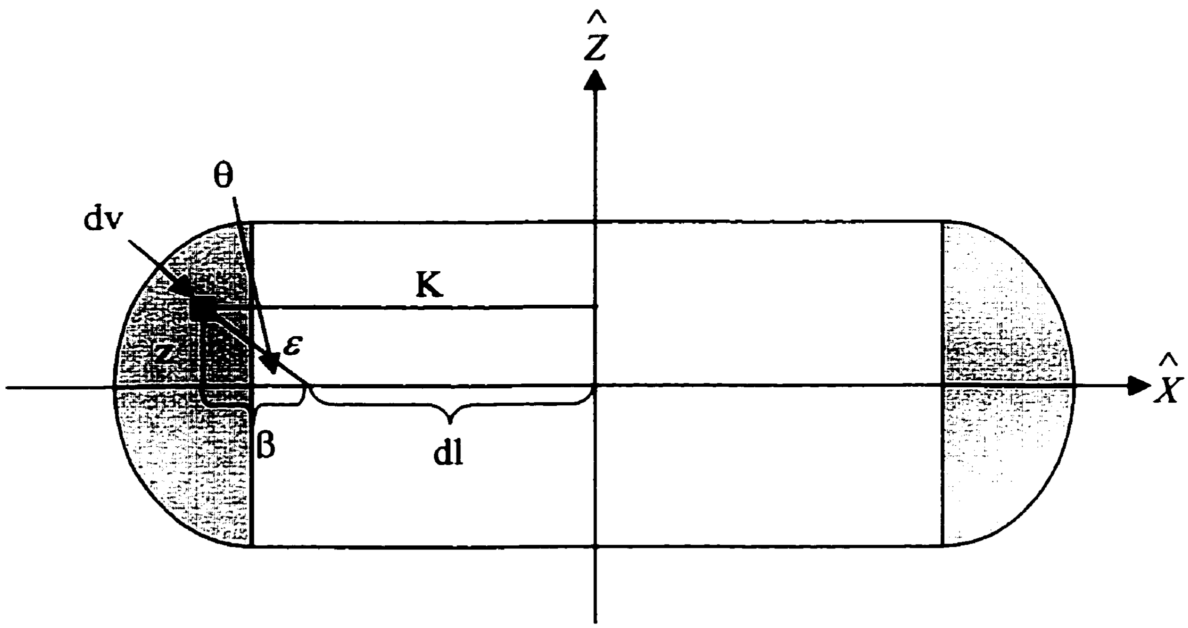
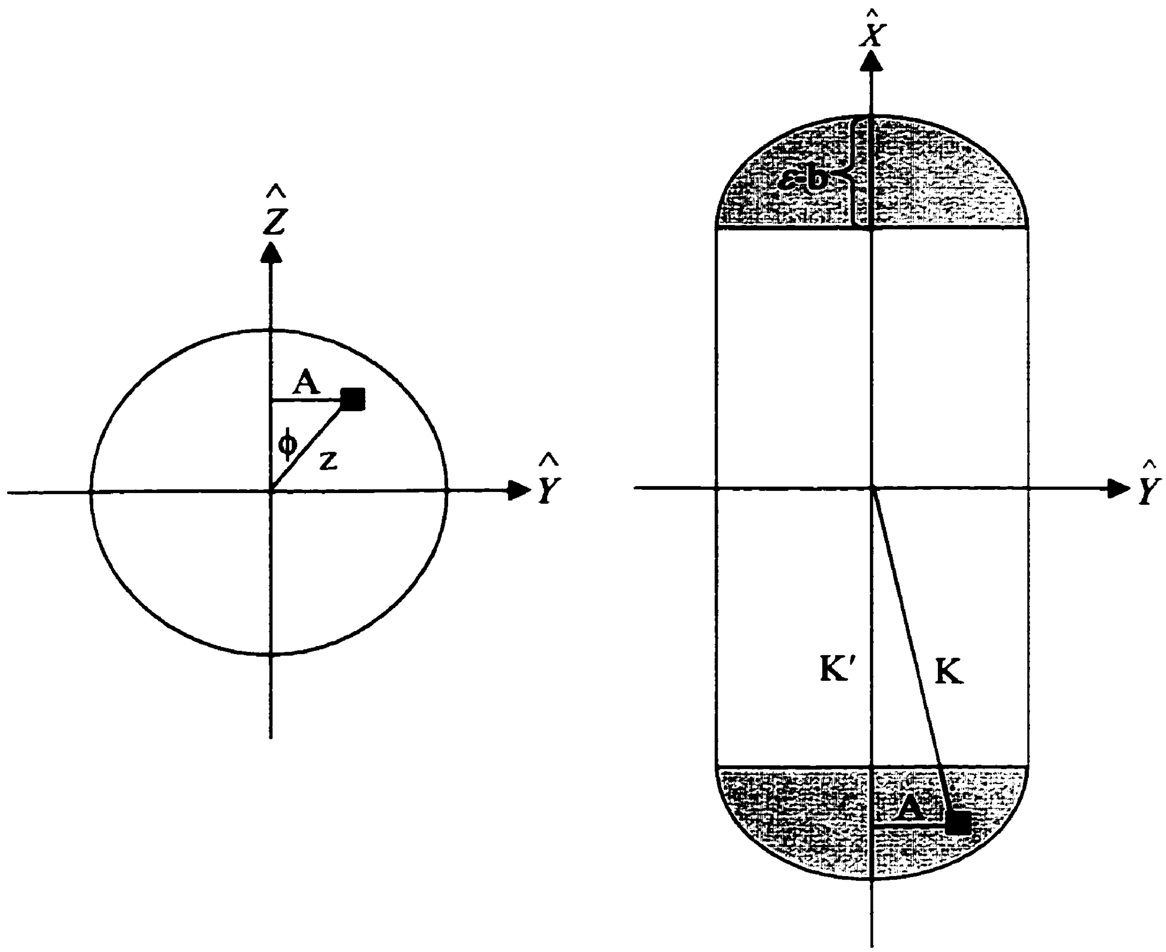


Figure 4.3.1: SIDE VIEW OF CYLINDER WITH SHADED END-CAPS.

Since the moment of inertia is the sum of the mass times the distance squared for each element, it is possible to calculate the moment of inertia for each hemisphere separately. The results can then be added to the cylinder body to get the total moment of inertia for the entire cylinder. Figures 4.3.2 and 4.3.3 help give a complete picture of the position of the volume element that is used to calculate the moment of inertia.



Figures 4.3.2 and 4.3.3 : END VIEW AND TOP VIEW OF THE POSITION OF THE VOLUME ELEMENT IN THE CYLINDER.

In this case the r from equation 4.1.2 that is to be integrated is K shown in figures 4.3.1 and 4.3.3. The length K' is the sum of dl and β from figure 4.3.1. This then implies that:

$$K^2 = A^2 + K'^2 \tag{4.3.1}$$

$$K^2 = A^2 + (dl + \beta)^2 \tag{4.3.2}$$

$$K^2 = A^2 + dl^2 + 2dl\beta + \beta^2 \tag{4.3.3}$$

where $dl^2 = dx^2 + dy^2 + dz^2$ and in this case $dy = dz = 0$. The constant β can be found from figure 4.3.1.

$$\beta = \varepsilon \cos\theta \tag{4.3.4}$$

where ε is the radius of the end-cap. From figure 4.3.2 it can be seen that the constant A is defined as:

$$A = z \sin\phi \tag{4.3.5}$$

where from figure 4.3.1

$$z = \varepsilon \sin\theta \tag{4.3.6}$$

Substituting equations 4.3.4-4.3.6 into 4.3.3 gives:

$$K^2 = dl(dl + 2\varepsilon \cos(\theta)) + \varepsilon^2\left(\cos^2(\theta) + \sin^2(\theta)\sin^2(\phi)\right) \tag{4.3.7}$$

By utilizing equation 4.1.2, where $dm = \sigma \varepsilon^2 \sin\theta \, d\theta \, d\phi \, d\varepsilon$, the equation for the moment of inertia takes on the form of:

$$I = \sigma \int_b^{\varepsilon} \int_0^{a} \int_0^{2\pi} \begin{bmatrix} dl\left(dl\,\varepsilon^2 \sin\theta + 2\varepsilon^3 \sin\theta \sin\phi\right) + \\ \varepsilon^4\left(\cos^2\theta \sin\theta + \sin^3\theta \sin^2\phi\right) \end{bmatrix} d\phi \, d\theta \, d\varepsilon \tag{4.3.8}$$

Solving this equation produces the following:

$$I = \sigma\pi \begin{Bmatrix} dl\left[\frac{2}{3} dl(1 - \cos(a))\left(\varepsilon^3 - b^3\right) + \frac{1}{2}\sin^2(a)\left(\varepsilon^4 - b^4\right)\right] + \\ \frac{1}{5}\left(\varepsilon^5 - b^5\right)\left[\frac{4}{3} - \cos(a) - \frac{1}{3}\cos^3(a)\right] \end{Bmatrix} \tag{4.3.9}$$

where a is the maximum angle of θ which is less than or equal to 90° and b is the minimum value of the hemisphere found from:

$$b = \varepsilon \cos(a) \tag{4.3.10}$$

The value of b is important as it helps define the thickness of the hemisphere. The value of ε-b is the thickness of the hemisphere as shown in figure 4.3.3. The volume density is defined by σ.

4.4 Moment of Inertia of End Caps (Case 2)

The second case examines the moment of inertia of the end-caps when the radius is greater than half the length of the cylinder as shown in figure 4.4.1. This is required since the expression for K' in equation 4.3.1 is different. Using figure 4.3.3 the expression is:

$$K' = \varepsilon \cos\theta - dl \tag{4.4.1}$$

The equation of K^2 then takes on the form of:

$$K^2 = A^2 + (\varepsilon \cos\theta - dl)^2 \tag{4.4.2}$$

The final form of K^2 will then be:

$$K^2 = dl(dl - 2\varepsilon\cos\theta) + \varepsilon^2\left(\cos^2\theta + \sin^2\theta\sin^2\phi\right) \tag{4.4.3}$$

where $A{=}z\ sin\phi$ and $z{=}\varepsilon\ sin\theta$ as can be seen from figures 4.3.2 and 4.4.1.

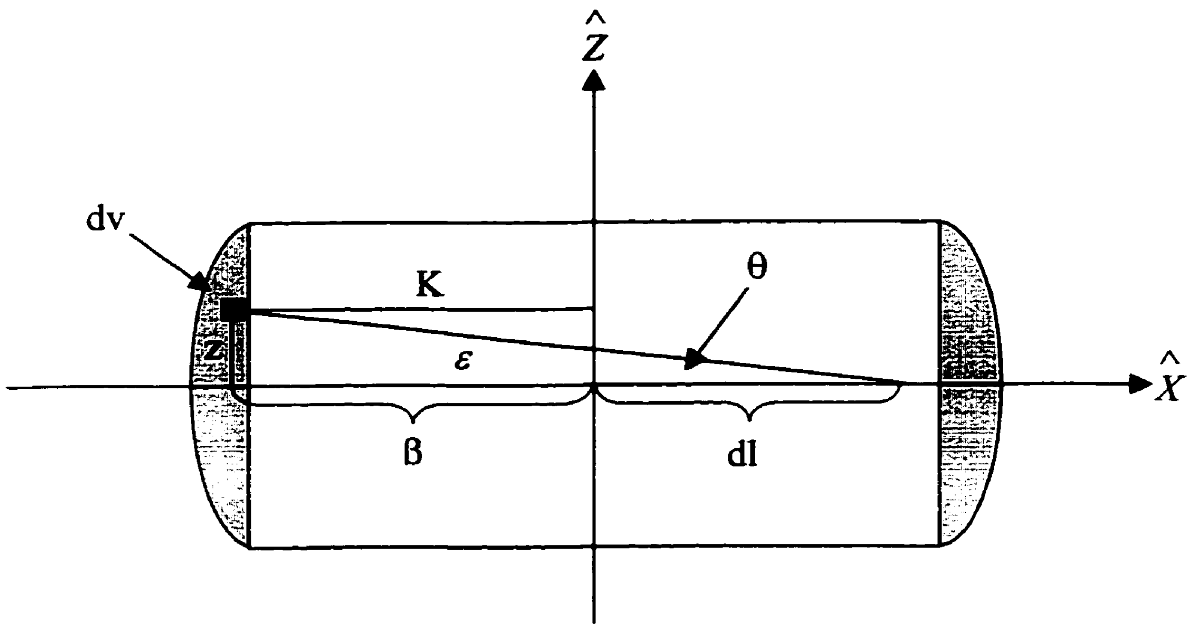


Figure 4.4.1: SIDE VIEW OF CYLINDER WITH SHADED END-CAPS.

When K^2 is substituted for r^2 in equation 4.1.2 and *dm* is replaced by $\sigma\,\varepsilon^2\,\sin\theta\,d\theta\,d\phi\,d\varepsilon$, the solution for the moment of inertia becomes:

$$I = \pi\sigma\left[\begin{array}{l} dl\left\{\frac{2}{3}dl(1-\cos(a))\left(\varepsilon^3-b^3\right)-\frac{1}{2}\sin^2(a)\left(\varepsilon^4-b^4\right)\right\}+ \\ \frac{1}{5}\left(\varepsilon^5-b^5\right)\left(\frac{4}{3}-\cos(a)-\frac{1}{3}\cos^3(a)\right) \end{array}\right] \qquad (4.4.4)$$

The moment of inertia for a cylinder with round end-caps would be the sum of equations 4.2.4 and 4.3.9 or 4.4.4 for the appropriate radius of curvature of the hemisphere on either end.

$$I = M\left[\frac{1}{12}K^2 + \frac{1}{4}\rho^2\right] + \pi\sigma\left[\begin{array}{l} dl\left\{\frac{2}{3}dl(1-\cos(a))\left(\varepsilon^3 - b^3\right) - \frac{1}{2}\sin^2(a)\left(\varepsilon^4 - b^4\right)\right\} \pm \\ \frac{1}{5}\left(\varepsilon^5 - b^5\right)\left(\frac{4}{3} - \cos(a) - \frac{1}{3}\cos^3(a)\right) \end{array}\right] \quad (4.4.5).$$

The "±" will determine whether or not the radius of the end-cap is less than or larger than half of the cylinder length.

These equations have been implemented into the computer program as a subroutine such that it can be called upon when the inertia is required. A copy of the routine can be found in Appendix B.3. For these experiments the subroutine was used to help calculate the torque on the cylinder about the central axis.

The next two chapters will present the experimental set-ups used, the experimental results and the theoretical results.

<u>Note:</u> Experimentally it is observed that the cylinders rotate about the central axis as defined in this chapter. For this reason the inertia about this axis alone is required.

Chapter 5

Experimental Set-up

There are many designs that can be used to trap particles such as the first trap designed by Ashkin [19] where two opposing beams were used. There is also a trap that is designed by using a diffraction grating [20] where the interference of the beam is used to trap particles.

In this chapter three laser trap designs will be presented. The first to be discussed is the trap labeled as the "top-down" design. The second trap talked about is known as the "bottom-up" design. The final design is known as the "horizontal" trap. In the first system there will be a brief description of the components that are used in all three designs then a description of the procedures used to align the system will follow. Any variance in the components or alignment for the other two designs will be discussed in their separate sections.

5.1.1 Setup of "Top-Down" Laser Trap

The first setup implemented was the "top-down" laser trap. The name "top-down" is used as the laser comes in from the top and is directed downward. This is a standard design that is fairly easy to setup but harder to align because it has more components than the other designs. Figure 5.1.1 is an illustration of the trap.

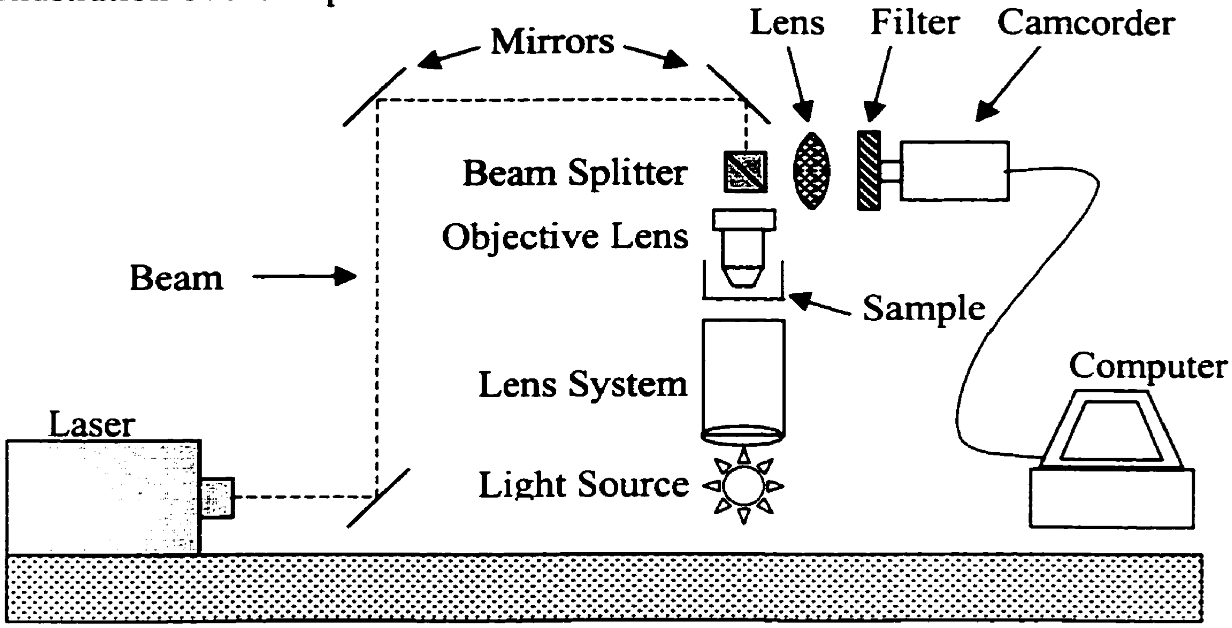


Figure 5.1.1: TOP-DOWN LASER TRAP DESIGN.

The system is placed upon a stable table as the trap is sensitive to vibrations. The monochromatic light of the laser is directed by a series of mirrors such that it is guided in through the top of the system. The beam travels through the beam splitter first. After the beam splitter the beam continues on into the objective lens which focuses the beam to a width of approximately 1 μm.

The light source is used to see the particles trapped in the beam. The light travels upwards through the lens system which focuses the light to a small point to allow as much light into the objective as possible. The light then travels from the objective into the beam splitter where the beam is directed towards the camcorder, which records a video of the objects in the trap region. The lens is used to focus the light into the camcorder while the filter is used to filter out the laser light otherwise the camcorder will be saturated. The computer is used to capture the video images as single frames.

5.1.2 Component Specifications

For the experiments an Argon-ion laser is used. The wavelength, λ_o, of the laser is 515 nm. The maximum power output of the laser is 1.5 watts. The mirrors are front surface mirrors that are height adjustable. The beam splitter is a 50% intensity splitter. The beam splitter is used to direct the signal from the light source to the camcorder. The objective that is used is a 100X oil immersion lens with a numerical aperture (N.A.) of 1.30. The container to hold the samples is a 5 cm diameter petrie disk. The lens system was created using a program that designs optical systems, which was invented by R. C. Gauthier. Figure 5.1.2 is a schematic of the lens system.

There are two lenses used in the system. The first is an asymmetric lens with the specifications of 27.0 mm diameter and a focal length of 20.0 mm. The second lens consists of two plano-convex lenses each with a diameter of 27.0 mm and a focal length of 26.0 mm put together flat side to flat side.

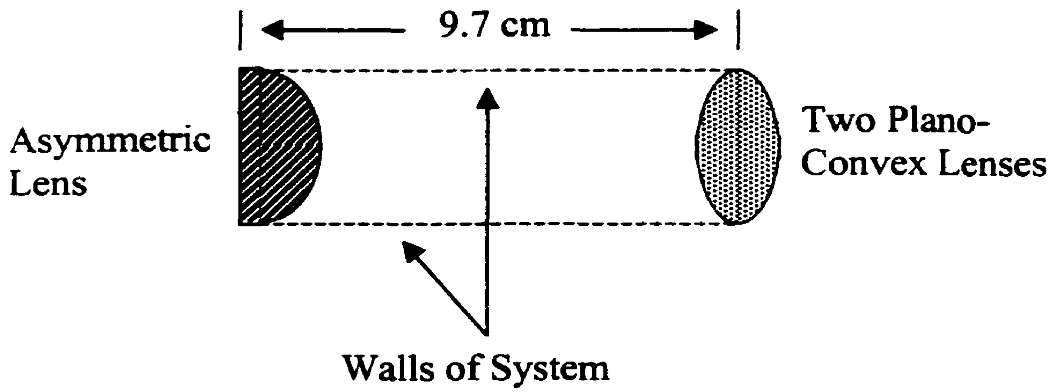


Figure 5.1.2: LENS SYSTEM FOR LASER TRAP DESIGNS.

A piece of aluminum was hollowed out using a lathe and both ends where machined to fit the two lenses described above. The system is placed approximately 20 mm away from the light source and approximately 21 mm away from the objective. The light source that is used is a standard automobile headlight bulb. Three different types of halogen bulbs were tested with a 12V, 65W/55W light from Motomaster (ID# 9003 HB2 H4) being the most effective. The lens in front of the camcorder is used to focus and increase the intensity of the signal from the beam splitter, which has a long focal length of 200 mm. The filter was used to block out the laser light from the beam splitter so that the camcorder did not become saturated. The images are recorded with a camcorder, or a camera, which can be connected to a VCR where the images are stored. The computer is then used to capture pictures from the VCR. These are all the components that are required for a single beam laser trap.

5.1.3 Alignment Procedures

A very important step to building a laser trap, besides stability, is the alignment of all the components. The trap can be divided into two sections, the trap section and the lighting section. The trap section consists of the laser, mirrors, beam splitter, sample holder and the objective lens. The lighting section involves the light source and the lens system. Both need to be aligned properly to produce a good working trap that will give quality pictures. This section will discuss the procedures to aligning both the trap and lighting sections so that the experiments can be reproduced if required.

The alignment of the trap is the more crucial of the two sections. If any one component is slightly out of alignment then the beam will not enter the objective lens correctly. The laser beam will not focus to a tight waist therefore there will not be a large enough intensity gradient and objects will not trap. The first thing to do for the alignment is to make sure that the laser is secure and level to the table so that it will not move at all if the table is bumped or jarred. The mirrors are then set into place. The three mirrors also have to be aligned. It is suggested to align the mirror nearest to the laser first. The first mirror is used to deflect the beam straight up to the next mirror, which deflects the beam to the third mirror. The third mirror reflects the beam down to the beam splitter. To align the first mirror it is suggested to use a pinhole shutter. The pinhole shutter is parallel to the tabletop and attached to two posts such that it can move vertically only as shown in figure 5.1.3.

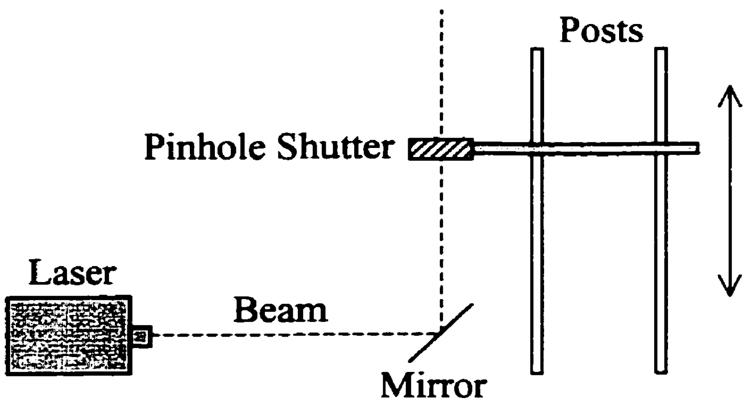


Figure 5.1.3: ALIGNMENT OF FIRST MIRROR.

The shutter is closed which produces a pinhole. The purpose is to deflect the beam straight up to the next mirror, which will then be a 45° angle. If the beam is at a 45° angle then the beam should pass through the pinhole at the top and bottom of the posts. If it is not at a 45° angle then the mirror must be adjusted until it is. This same procedure can be done for the second mirror by allowing the shutter to only move horizontally as shown in figure 5.1.4.

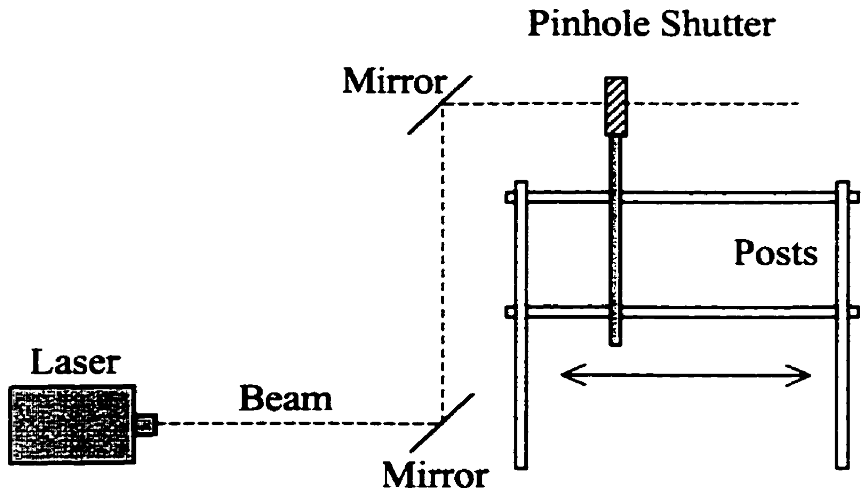


Figure 5.1.4: ALIGNMENT OF SECOND MIRROR.

The final mirror can be aligned by two ways. The shutter can be used in the same way as the first mirror was aligned or a front surface mirror can be used. The front surface mirror is placed on the tabletop below the third mirror. The mirror will reflect the beam back upon itself if the third mirror is aligned properly. If it is not then the third mirror is adjusted until the beam does reflect back to the laser. It is easy to do this alignment, as it is possible to follow the beam while the mirror is being adjusted.

The next component to be introduced into the trap section is the beam splitter. The beam splitter can translate the beam a small distance parallel to the original beam causing the system to be misaligned. The alignment can be done with the small mirror again. By placing the mirror below the beam splitter the beam can be aligned.

The next component in the trap system is the sample holder. This is the component that is moved around when trying to find the particles. The alignment of the holder is not as critical as the other components but a level holder will help prevent the petrie dish from crashing into the objective lens. Once again the front surface mirror can be used for the alignment. It is preferred to have the holder on a tilt stage to facilitate its inclination adjustments. The holder is attached to a XYZ stage, which allows the petrie dish to be moved when objects are being trapped.

The last and most important part to any trap is the objective lens. The objective is the component that focuses the beam to a very small, tight, waist, which produces a large enough intensity gradient that is capable of trapping objects. The alignment of this component is the most critical. The objective is mounted on a XYZ stage so that it has three degrees of freedom. To align the lens the front surface mirror is used. The mirror is placed on top of the lens as shown in figure 5.1.5

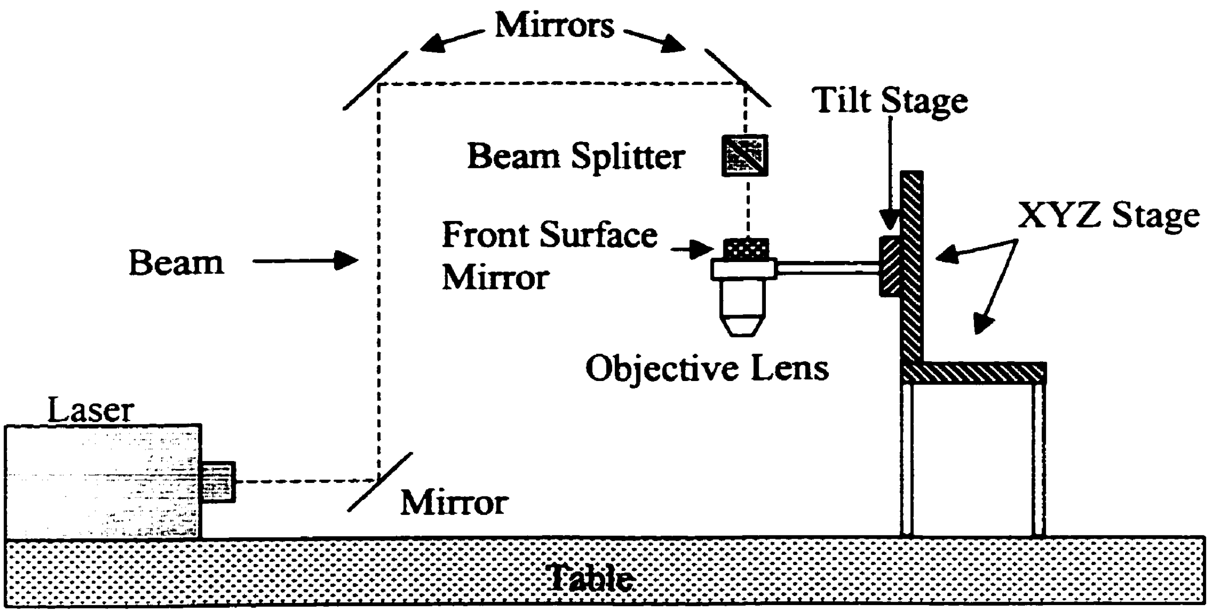


Figure 5.1.5: FRONT SURFACE MIRROR PLACED ON TOP OF OBJECTIVE. OBJECTIVE IS MOUNTED ON XYZ STAGE AND TILT STAGE IF REQUIRED.

The mirror will reflect the beam back upon itself. The tilt stage can be used to correct the objective if it is on a tilt. This can be seen, as the beam will reflect off of the mirror at an angle. Once the beam is aligned the mirror can be removed. It is usually found that the beam does not enter the objective exactly in the center so the XYZ stage can be used to position the objective. Adjusting the XYZ stage will not change the alignment of the objective but once the lens is in place the XYZ stage is not to be touched. This completes the alignment of the trap system.

To be able to view the objects that are trapped the lighting system must be included. It also has to be aligned to be able to get as much light in the objective lens to produce enough contrast so that a good quality image may be captured. For the alignment the laser should be off. It will also be easier to adjust the light source and lens system if both are mounted on XYZ stages. The alignment uses a very simple tool, a piece of paper. After the lens system and the light source are introduced to the trap, place a small piece of paper on top of the opening of the objective lens. Turn out the lights in the room but turn on the light source. When any amount of light enters the objective through the bottom of the lens the light should appear as a bright spot on the paper. Adjust the light source and the lens system until the spot on the paper is the brightest possible.

The filter and long focal length lens are adjusted in front of the camcorder until the best quality picture is achieved. If the filter that is used blocks out all of the laser light then there are two techniques that can be used to find the focal spot on the camcorder. The first is to place a shutter in front of the camcorder aligned with the beam and then close the shutter to a pinhole. The beam can be found by adjusting the camera until the pinhole is viewed on the monitor that is receiving the signal from the camcorder. When closing the shutter viewing the monitor will help determine the direction the camcorder is required to move. A faster way of determining the location of the laser beam on the camcorder is to place the edge of an object in the path of the beam that is directed towards the camera and try to locate the edge on the monitor. This concludes the alignment of the entire top-down laser trap. The next trap to be discussed is the bottom-up trap.

5.2 Setup of "Bottom-Up" Laser Trap

The bottom-up trap is designed such that the laser beam will be directed upwards through the system. This is the opposite of the top-down design as the objective lens is now pointing upwards so the trap region is above the objective. This design has an advantage over the top-down design as the beam is directed upwards so that the trap will work against gravity producing a more effective trap. The schematics of this trap are shown in figure 5.2.1.

All the components are the same with the exception of the sample holder and two less mirrors. The difference in the sample holder is that it is not a petrie dish that is used. The petrie dish can not be used because the working distance of the objective lens is 0.2 mm so the focus would be in the glass. To overcome this problem a new sample holder was designed.

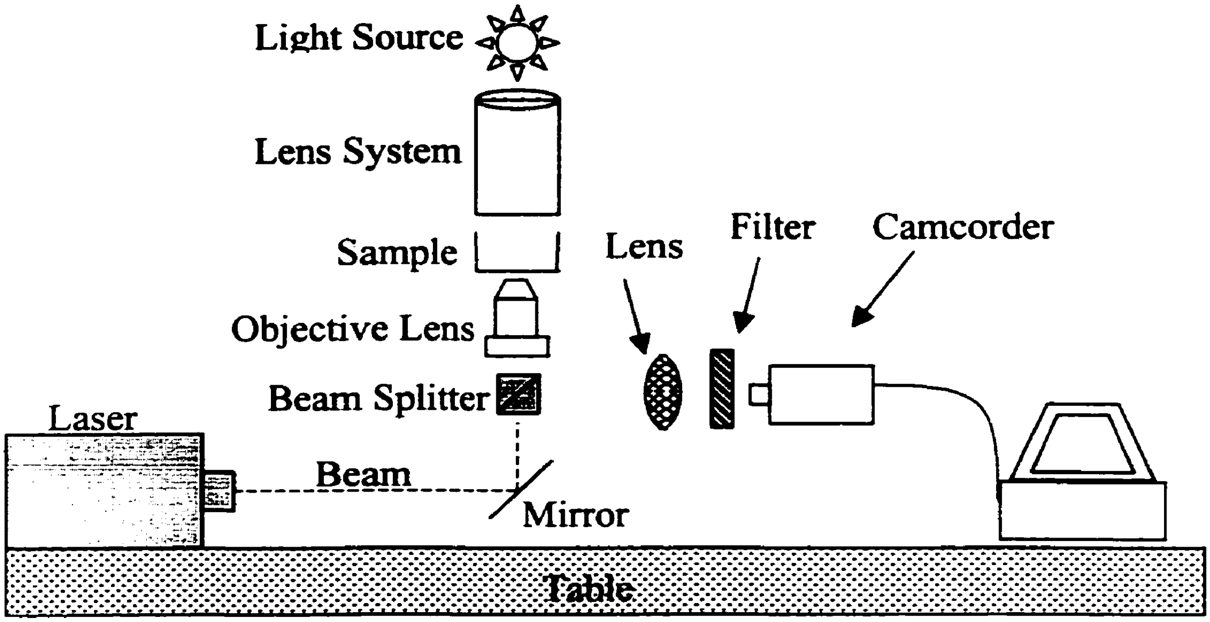


Figure 5.2.1: SCHEMATIC OF BOTTOM-UP LASER TRAP DESIGN.

It was found that microscope slide cover slips have a thickness of approximately 0.12 mm which would leave a working distance of 0.08 mm or 80 μm. This is acceptable as the cylinders that are used in the experiments have diameters of 5 and 7 μm and a length of no more than 70 μm. The cover slip is attached to a rectangular piece of plastic using silicon, and is the base of the sample holder. A 13 mm hole is drilled into plastic to forge the walls.

In the alignment of the components the same techniques can be applied as was discuss in Section 5.1.3. For the sample cell the alignment is important as the cover slip is very fragile and when the cell is being moved to find cylinders it is extremely easy to break the cover slip if it is not aligned properly. A small level such as a line level will make sure that the cell is parallel to the table in both directions but to make sure that the cell is parallel to the objective another technique is used. With the laser on, slowly move the cell down using the Z stage and watch the lower mirror. As the cover slip moves into the focus there will be a Fresnel ring reflected back onto the mirror. The closer the slip approaches to the focus region, the smaller the ring will become. When the cell is aligned properly to the objective the ring will be a perfect circle that focuses to a point on top of the beam. Also when the cell is moved past the focus the ring will appear again and it too should be a perfect circle if everything is done properly. This completes the section on the bottom-up laser trap. The last design to be discussed is the "horizontal" laser trap.

5.3 Setup of the "Horizontal" Laser Trap

The last design to be considered is described as a "horizontal" laser trap. It is called a horizontal laser trap because the laser comes in horizontally. A group at the University of California under the direction of W. Wang was able to produce a horizontal trap [19]. The schematic of the horizontal trap is given in figure 5.3.1.

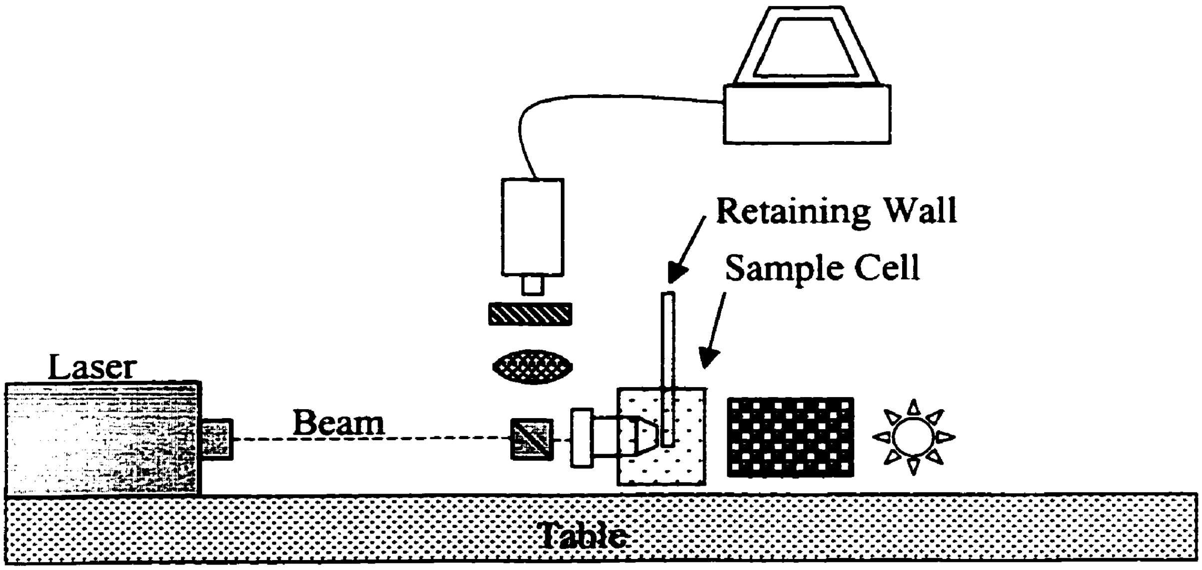


Figure 5.3.1: SCHEMATIC OF HORIZONTAL LASER TRAP DESIGN.

The change made for this trap is the sample cell. The design of the cell was the most difficult of the whole setup and is shown in figure 5.3.2.

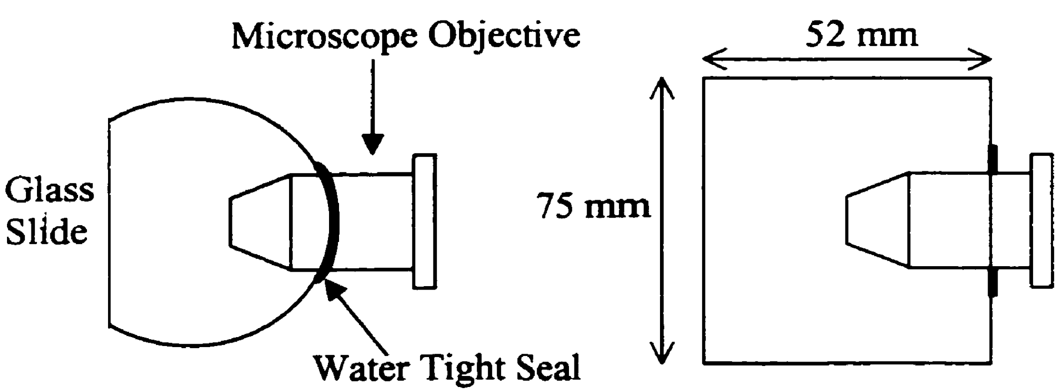


Figure 5.3.2: TOP AND SIDE VIEW OF SAMPLE CELL.

The problem with this cell is that the objective cannot be moved relative to the sample volume holder. Trying to locate a particle was very difficult and relied entirely on them drifting by the focal region. There was no guarantee that the particle would trap with the presence of minute currents in the cell liquid. To over come these problems a small retaining wall was built. The wall and cell were made out of a clear piece of plastic. The wall was highly polished to allow light to pass through it. It was then attached to a post that was mounted on a XYZ stage so that it had three degrees of freedom. The design is shown in figure 5.3.3.

The wall is placed in the cell where it was able to move freely by the XYZ stage. The wall is small enough to allow the light from the source to pass through it without being hindered but large enough to slow down the currents in the liquid.

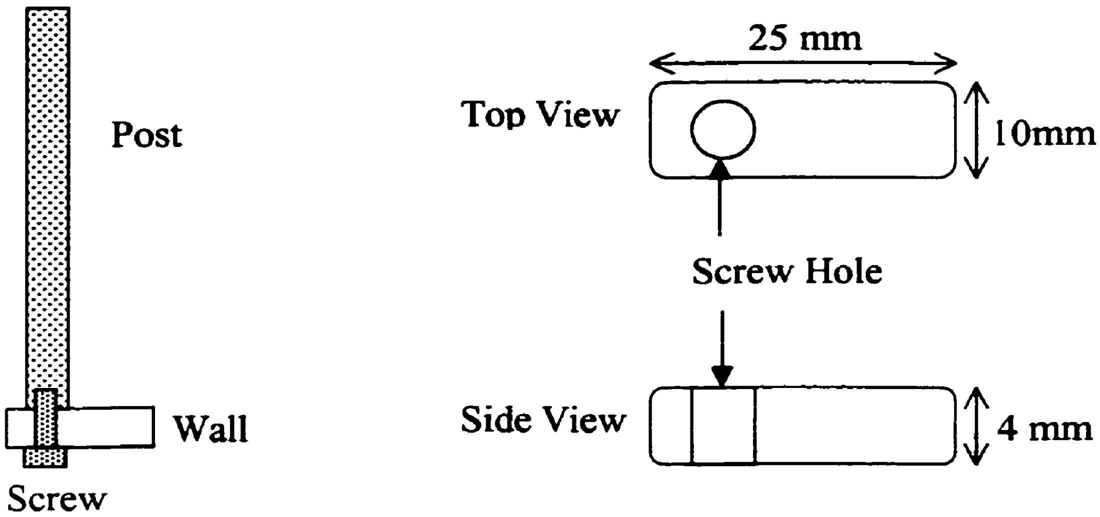


Figure 5.3.3: WALL ATTACHED TO POST WITH THE DIMENSIONS ON THE RIGHT.

We also used this holder to move objects into the focal spot of the beam.

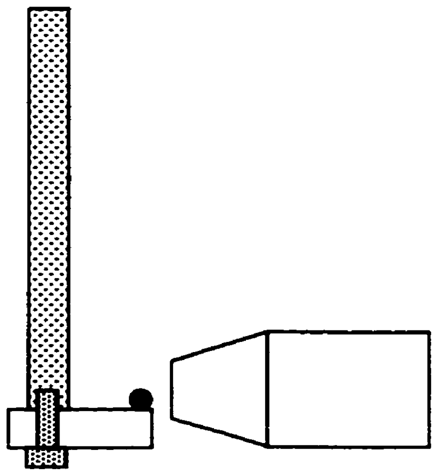


Figure 5.3.4: WALL USED TO MOVE OBJECTS INTO THE FOCAL SPOT OF BEAM.

For the alignment of the horizontal trap the best tool to use is the front surface mirror. The mirror will make sure that the beam comes back on itself. The XYZ stages, which the lens system, objective lens and the light source are mounted on, will give the freedom necessary to ensure that the laser will enter the objective correctly and the light source provides enough light into the system. This concludes the description and procedures of building the three separate laser traps. The next chapter will give the experimental results found using the three systems.

Chapter 6

Theoretical and Experimental Results

In a micro-machine the mechanical parts will include components called micro-actuators. To secure the actuator in place posts will be required but the actuator will still be able to rotate around the post. In the future it will be possible to build micro-machines using lasers to position the posts which can be cured into place then an actuator can be placed onto the post and fixed into place producing a micro-rotor. The actuator can be modeled out of multiple cylinders with a common center. The posts that would be used are cylindrical in shape as well. It has been predicted that the cylinders in a laser trap will experience centralizing forces and rotational torques. The forces and torques can be utilized to orient and rotate cylinders.

In this chapter the orientation of the cylinder in the laser trap and the continuous rotation of a cylinder, using multiple beams, will be presented.

6.1.1 Cylinder Orientation (Theoretical)

In the simple modeling of a cylinder in a single beam, it was observed that the cylinder would orient itself in one of three alignments: on its side, straight up and down or along its longest axis which is "corner to corner". The orientation is length and radius dependent. Figures 6.1.1 show the three different orientations.

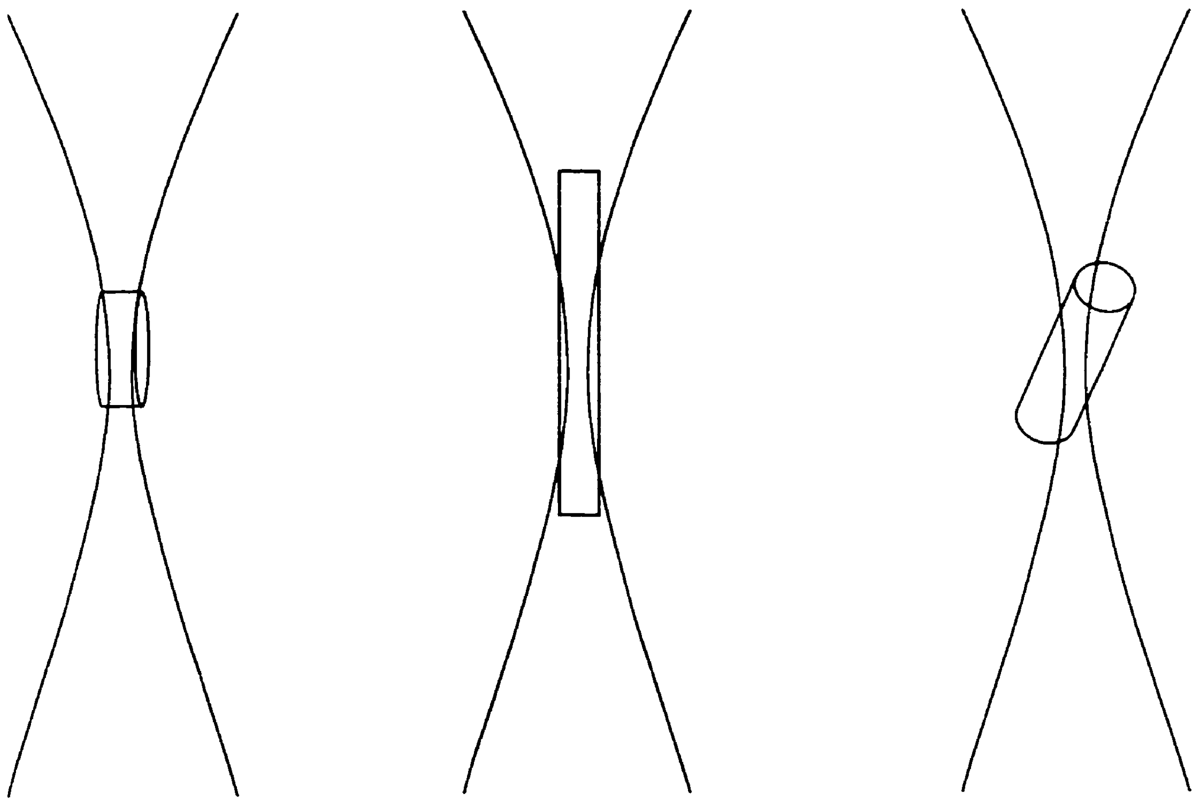


Figure 6.1.1: THE THREE CYLINDER ORIENTATIONS IN THE MINIMUM WAIST REGION; ON THE SIDE, STRAIGHT UP AND DOWN AND ON AN ANGLE.

Simulations were performed to determine the orientation of the cylinder length with respect to five different radii. The different radii chosen are 2.5, 3.5, 5, 7, and 10 μm. The 2.5 and 3.5 μm radii where chosen since cylinders of that size are available in our laboratory and permits experiments to be performed and verify the theoretical behavior.

The 5, 7 and 10 μm radii simulations were performed to extend the data. In Table 6.1.1 the stable angle for the different length radii cylinders and flat end-caps are presented.

Length μm	Angle (θ)				
	2.5μm	3.5μm	5μm	7μm	10μm
1	63	88	90	90	90
2	84	90	90	87	84
3	73	78	89	89	87
4	58	68	79	85	90
5	44	58	69	77	83
6	35	52	64	72	79
7	27	43	59	67	76
8	24	38	54	64	72
9	19	34	50	60	69
10	17	30	44	57	67
15	9	19	30	41	55
20	3	13	22	32	44
25	3	9	17	26	37
30	0	7	12	21	31
40	0	2	8	15	23
50	0	0	6	10	18
60	0	0	2	8	13
70	0	0	1	5	11
80	0	0	0	3	9

Table 6.1.1: ANGLE OF STABILITY FOR THE DIFFERENT LENGTHS VERSUS THE RADIUS.

Table 6.1.2 lists the parameters used for the simulations.

Beam Waist	1.25μm
Beam Power	100mW
Upper End Cap Radius	100μm
Lower End Cap Radius	100μm

Table 6.1.2: PARAMETERS FOR SIMULATIONS.

The results from Table 6.1.1 have been graphed and are presented below in figure 6.1.2.

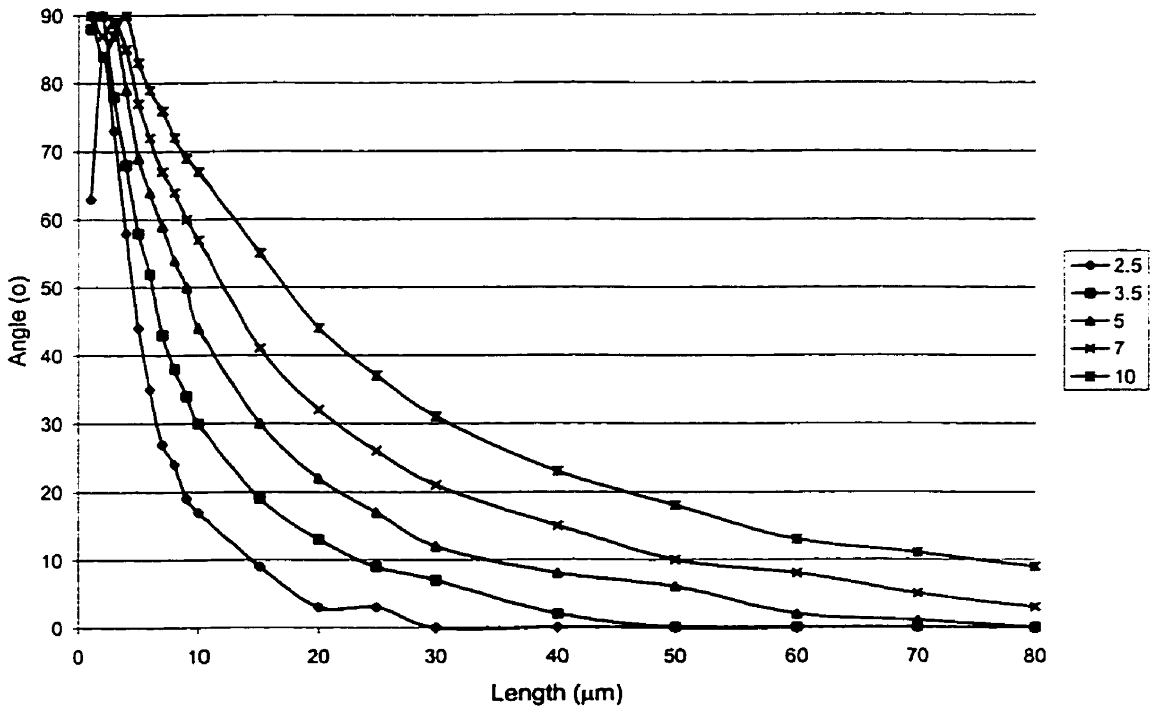
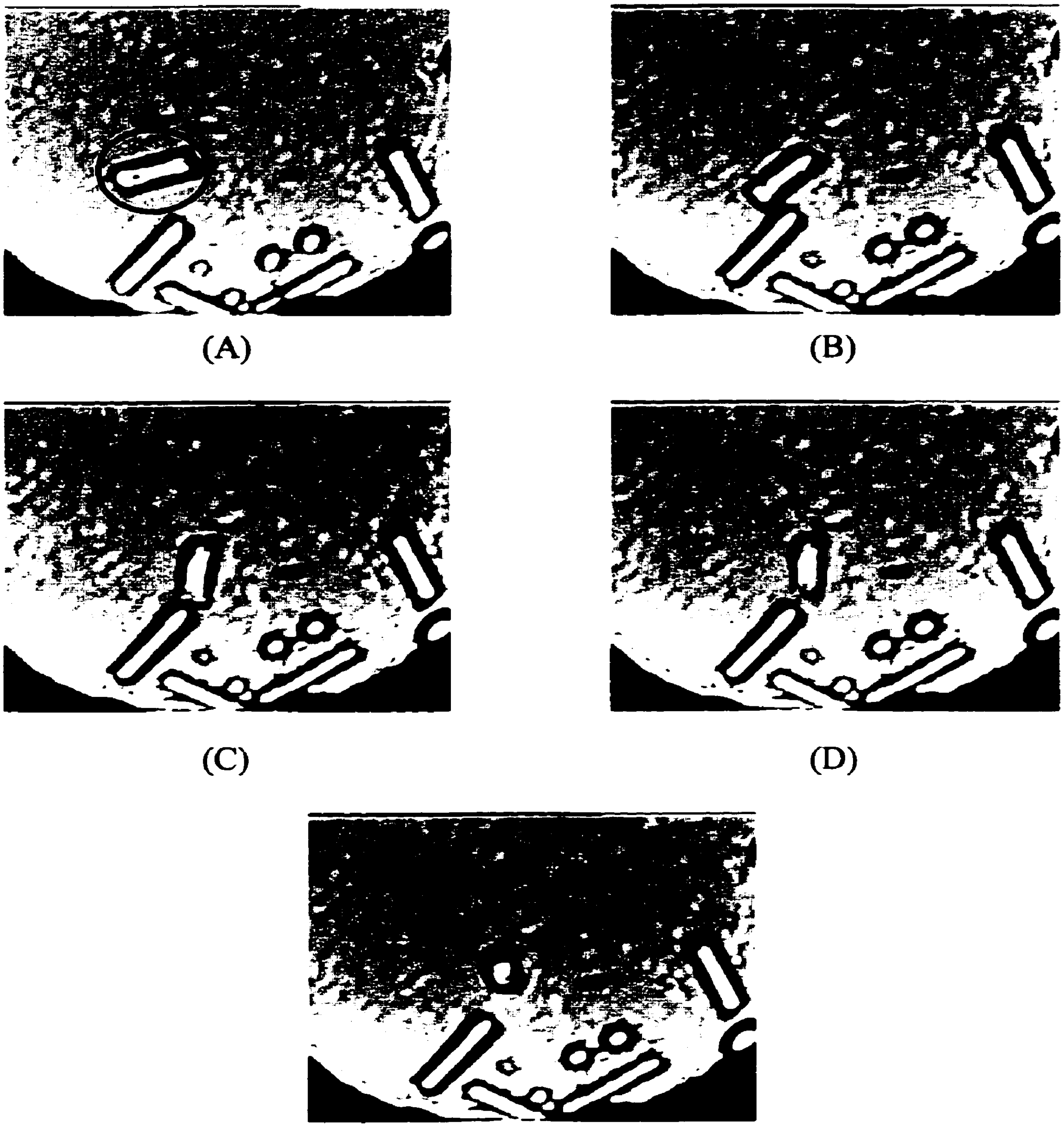


Figure 6.1.2: PLOT OF THE ANGLE OF STABILITY VERSUS LENGTH OF THE CYLINDER FOR THE RADII OF 2.5, 3.5, 5, 7, AND 10 μm.

The line of best fit of the three plots can be used as a reference for the building of micro-rotors. The line represents the angle that the cylinder will be stable in the beam for a particular length. This curve will aid in the creation of micro-rotors as the cylinders (posts) can be placed at a particular position and the desired angle using the laser trap. The medium that it is in can then be flushed out and replaced with a solution that is wavelength reactive to cure the post in place. This is the first step towards a micro-rotor and a micro-actuator. The results given here are for conditions found in the experiments. The next section will show an experimental verification that the cylinders will orient themselves in one of the three alignments.

6.1.2 Cylinder Orientation (Experimental)

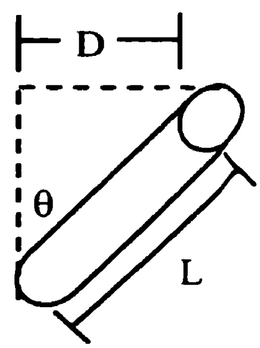
In chapter 5 three different laser trap systems were presented. The data presented in the previous section needed to be verified so the top-down trap was used first. If the laser beam does not have enough power then the cylinder will not be picked up. In figures 6.1.3A to 6.1.3C the 2.5 μm radius, 13 μm length cylinder, circled in figure 6.1.3A, is drawn towards the beam. In figure 6.1.3D the cylinder is starting to be lifted off of the bottom of the sample cell as the end of the cylinder towards the bottom of the picture indicates. The cylinder then aligns with the beam in figure 6.1.3E, which is a view of the end surface of the cylinder.



(A) (B) (C) (D) (E)

Figure 6.1.3: ALIGNMENT OF A 2.5 μm RADIUS CYLINDER IN A LASER BEAM.

In the above figure spheres can be seen along with other cylinders. These figures show a 2.5 μm radius cylinder with a length of approximately 13 μm aligning almost vertically. According to the data in table 6.1.1 the cylinder should be on an angle of approximately 12°. It can be seen that the cylinder is on a slight angle. If it was aligned perfectly up and down then it should have the same roundness that the spheres have in the picture. The cylinder is slightly elongated as the arrow indicates. This elongation is possible if either the cylinder has a deformity on the end or if the cylinder is on a slight incline. If there is a deformity in the cylinder then it should be seen when the cylinder was on its side. This then indicates that the cylinder is on an incline. Since the diameter of the cylinder is known then it is possible to determine the length. By measuring the distance (D) of the tilted cylinder, it is possible to determine the angle of the cylinder. By using elementary trigonometry, $\sin\theta = \frac{D}{L}$ which can be seen from the diagram to the right, the angle for the above cylinder was calculated to be 14±3°. When measuring D the length of the diameter of the cylinder is subtracted.



The power of the laser was approximately 100 mW for all the data taken unless specified.

In figures 6.1.4A – 6.1.4E a 2.5 μm radius cylinder with a length of approximately 26 μm is shown. From the theoretical values, the cylinder should be on an angle of 3°. The calculated angle is 3±0.7°.

It should be noted that the apparent size of the cylinders in the figures vary which is due to the results being taken on different days. The experiment was torn down and rebuilt several times and in each time, the camera position was different which results in the different sizes of the images. Also, the circled mark in figure 6.1.4D is a burn mark on the camera and can be seen in figures 6.1.5, 6.1.6 and 6.1.7. For figures 6.1.3 and 6.1.8 a different camera was used.

It was also found that the cylinders would align vertically using the bottom-up trap. Figure 6.1.5A to figure 6.1.5E shows the sequence of a 3.5 μm radius, 21 μm long cylinder aligning vertically. For this length of cylinder, the angle that it rests at should be 12°. The calculated angle is 11±1.2°.

Figure 6.1.5E shows the end of the cylinder where it is hard to tell but appears to be on a slight angle as the end is not perfectly round but is more oval shaped. This is expected for a cylinder on an angle.

(A)

(B)

(C)

(D)

(E)

Figure 6.1.4: VERTICAL ALIGNMENT OF 2.5 μm RADIUS 26 μm LENGTH CYLINDER.

The horizontal trap was also able to trap the cylinders. In figures 6.1.6A and 6.1.6B it can be seen that a cylinder drops from the top to the bottom where it is initially out of focus. As the cylinder enters the trap region, figure 6.1.6C, it starts to come into focus. The cylinder becomes ensnared in the trap and the cylinder becomes fully focused, figure 6.1.6D to figure 6.1.6F. This cylinder is aligned perpendicular to the laser beam or vertical to the tabletop. The laser could not overcome gravity to pull it into the beam though.

The horizontal trap is able to hold the cylinders horizontal to the tabletop or parallel to the beam. Figures 6.1.7A-6.1.7F shows the cylinder coming into the beam and straightens up. In both figures 6.1.6A and 6.1.7A (arrow) the angular line is the seam of the beam splitter

(A)

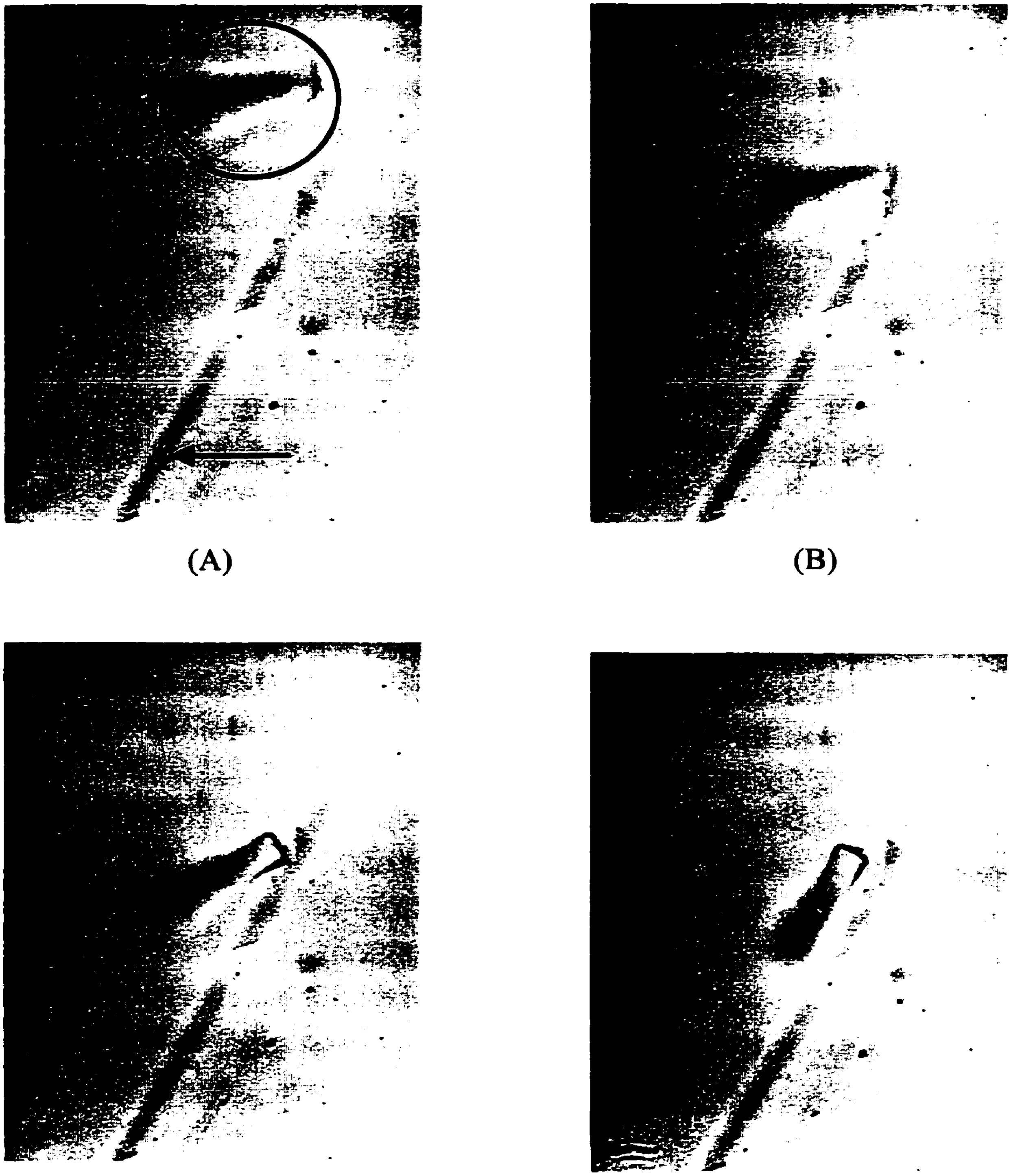
(B)

(C)

(D)

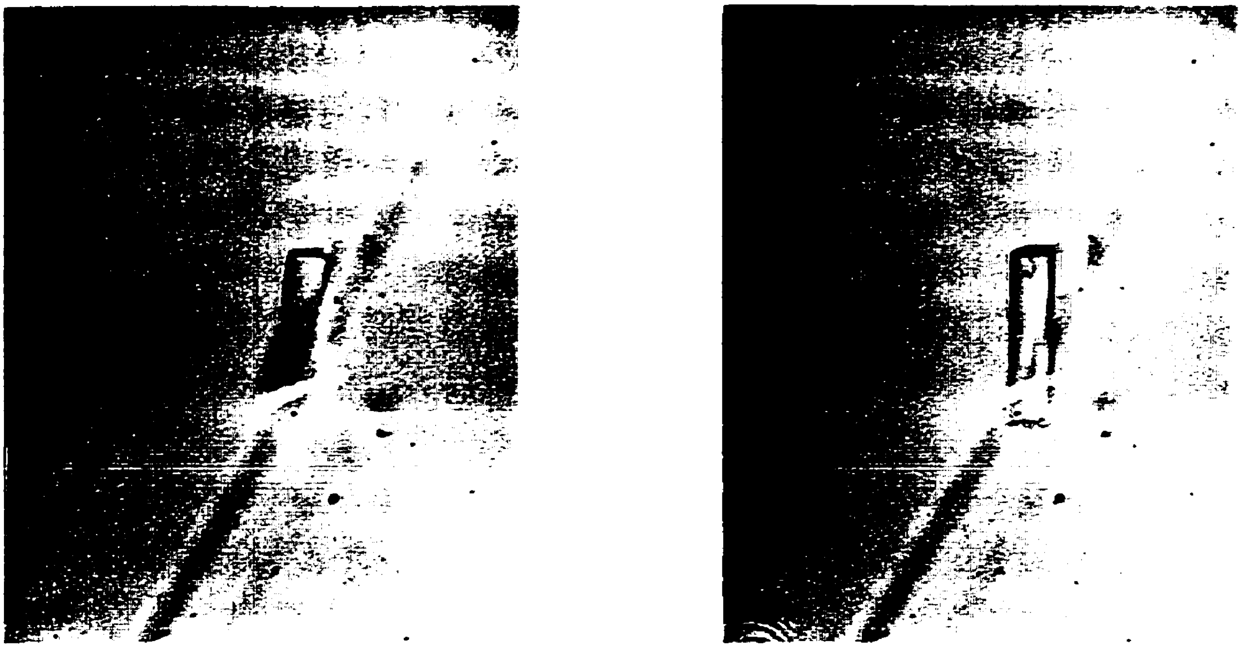
(E)

Figure 6.1.5: ALIGNMENT OF A 3.5μm RADIUS, 21 μm LENGTH CYLINDER.



(A) (B) (C) (D)

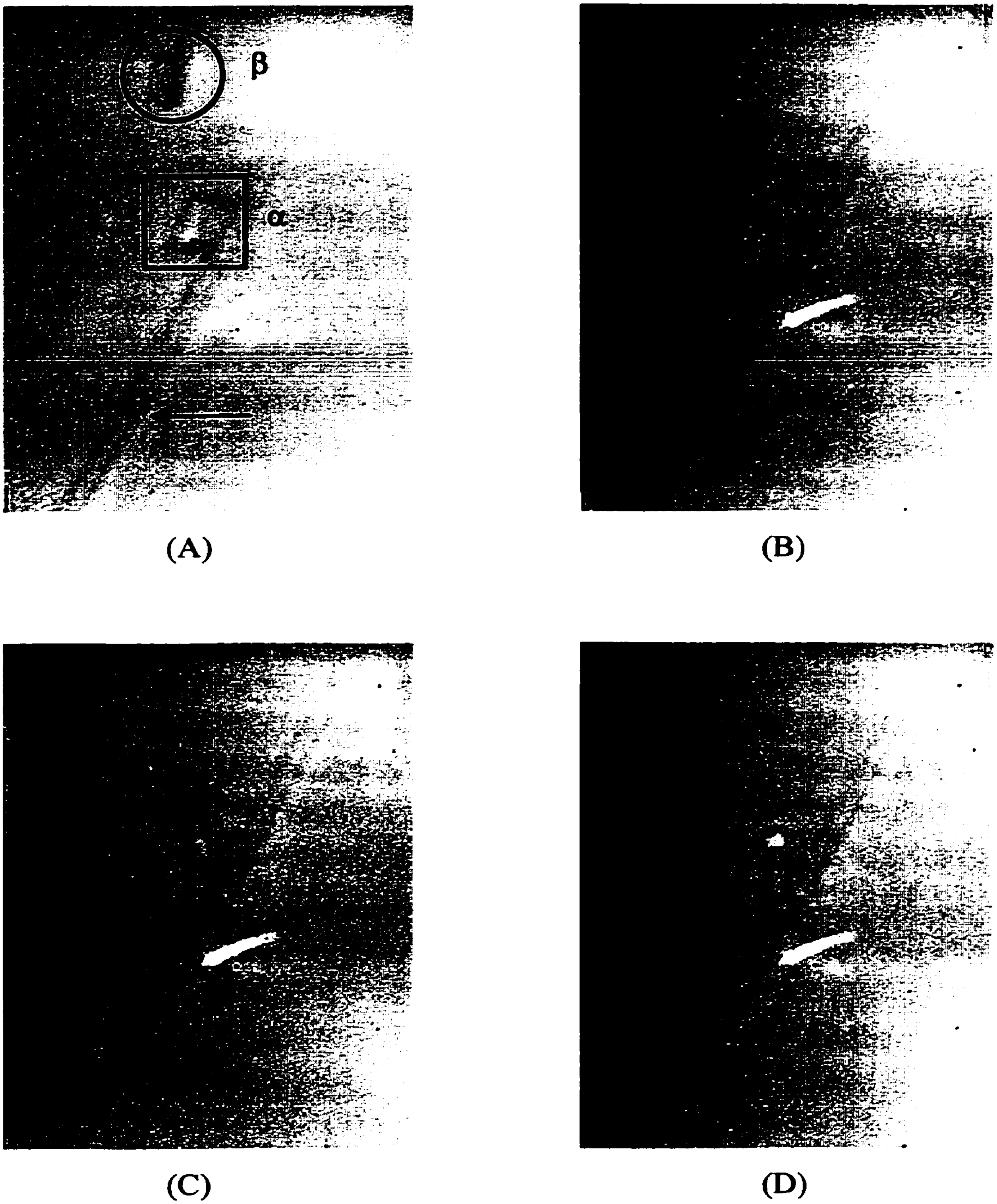
Figures 6.1.6A-6.1.6D: PERPENDICULAR ALIGNMENT OF CYLINDER TO BEAM USING THE HORIZONTAL TRAP.



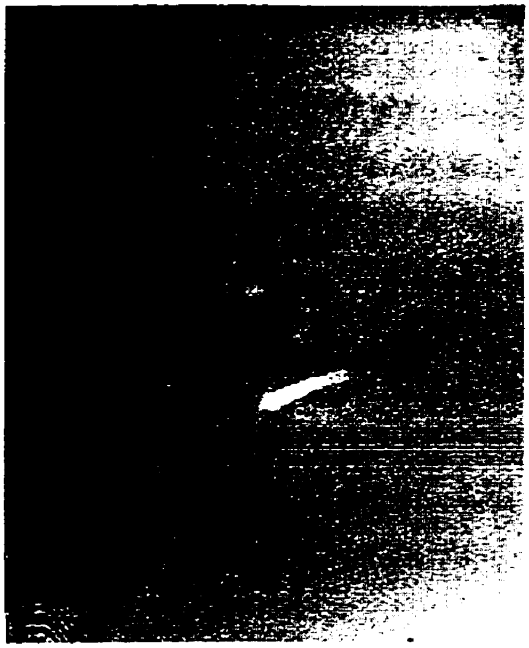
(E) (F)

Figures 6.1.6E-6.1.6F: PERPENDICULAR ALIGNMENT OF CYLINDER WITH BEAM USING THE HORIZONTAL TRAP.

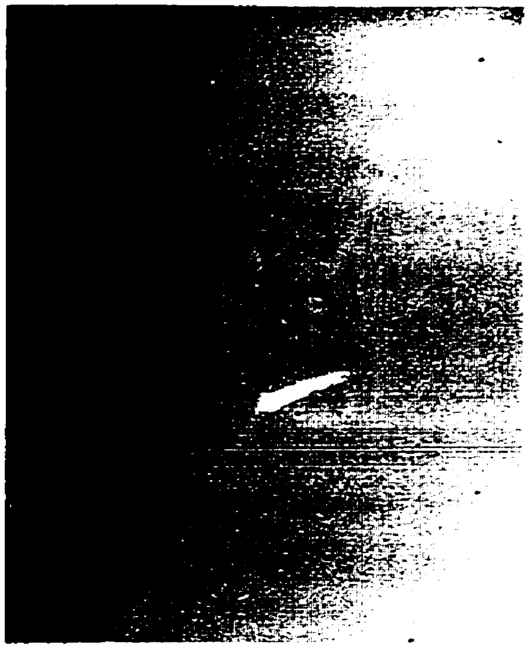
In figure 6.1.7A a trapped cylinder, labeled α and in the square, is in the path of a falling cylinder labeled β and in the circle. The cylinder, β, strikes the trapped cylinder, figure 6.1.7C, which causes the cylinder, α, to be pushed out while at the same time the cylinder, β, aligns itself with the beam, figure 6.1.7D. The beam momentarily traps cylinder β while α experience the force of gravity and starts to fall. β begins to fall as well as the beam is unable to keep the cylinder from falling out of alignment, figures 6.1.7F-6.1.7H.



Figures 6.1.7A-6.1.7D: PARALLEL ALIGNMENT OF CYLINDER WITH BEAM USING THE HORIZONTAL TRAP.

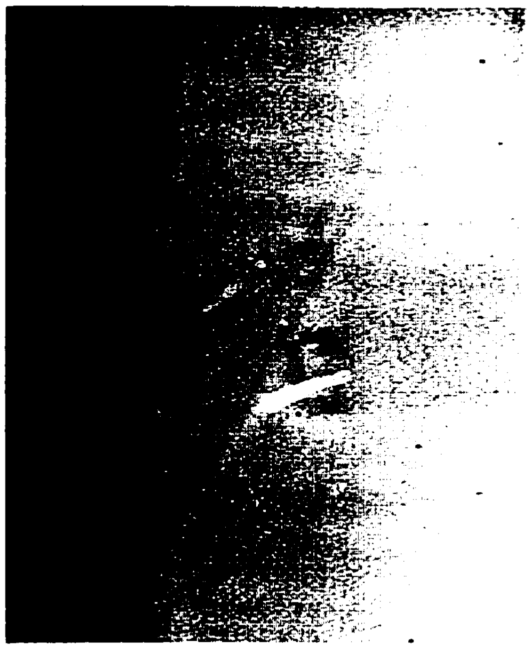


(E)



(F)

(G)

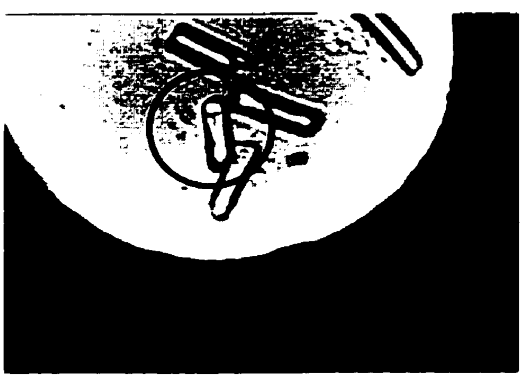


(H)

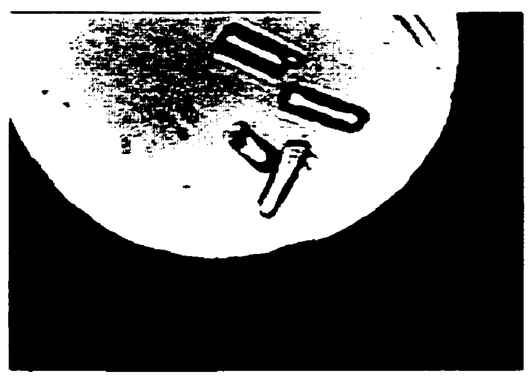
Figures 6.1.7E-6.1.7H: PARALLEL ALIGNMENT OF CYLINDER WITH BEAM USING THE HORIZONTAL TRAP.

Figures 6.1.6 and 6.1.7 demonstrate that the horizontal trap is capable of trapping cylinders. Due to gravity though it is extremely hard to keep the cylinders trapped parallel to the beam from falling into the perpendicular position as seen in figure 6.1.7.

One of the requirements needed to be able to create micro-machines is the ability to move the components to the desired location before they are set into place. In figures 6.1.8A-6.1.8C a cylinder is aligned in the beam using the top-down trap. While straight up and down in the trap the cylinder is moved to another location, figures 6.1.8D-6.1.8H. This is a 2.5 μm radius cylinder with a length of approximately 15 μm. It can be seen in figure 6.1.8E that the cylinder is also on an angle. The arrow points to the elongation that is seen if the cylinder is on a slant otherwise a circle should appear if it is completely vertical. Theoretically, the cylinder should be at an angle of approximately 9°. The calculated angle is 13±3°.

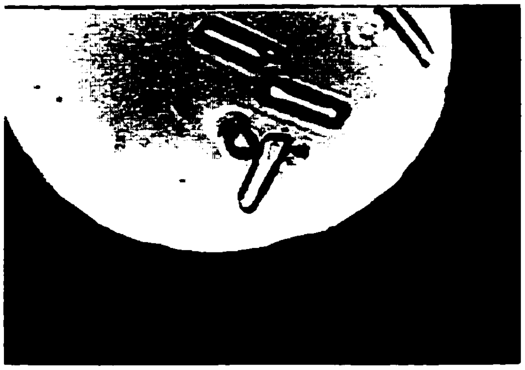


(A)

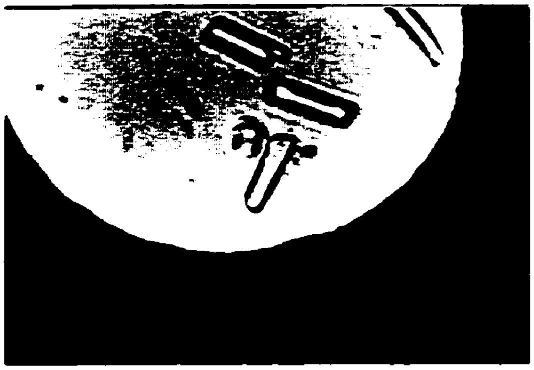


(B)

Figures 6.1.8A and 6.1.8B: ALIGNMENT AND MOVEMENT OF 2.5μm RADIUS, 15 μm LENGTH CYLINDER.

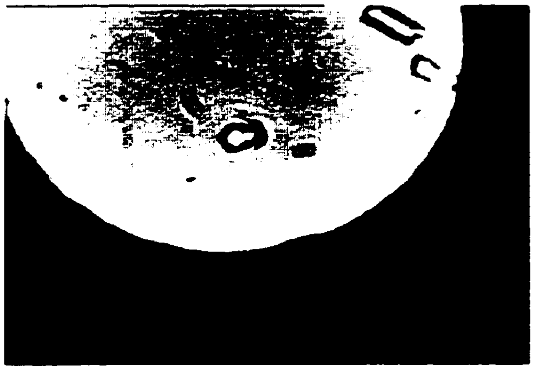


(C)

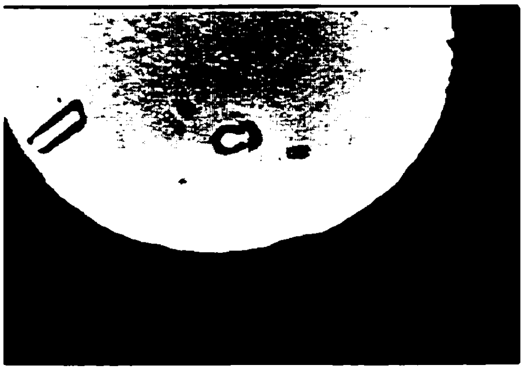


(D)

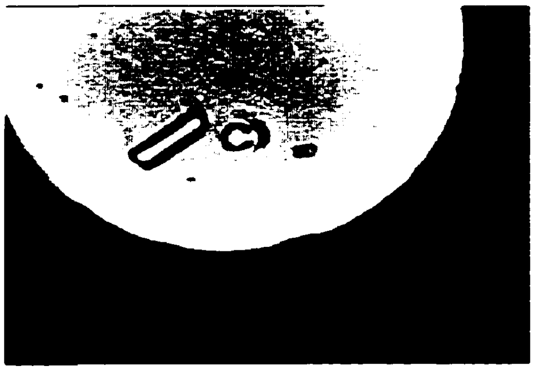
(E)



(F)



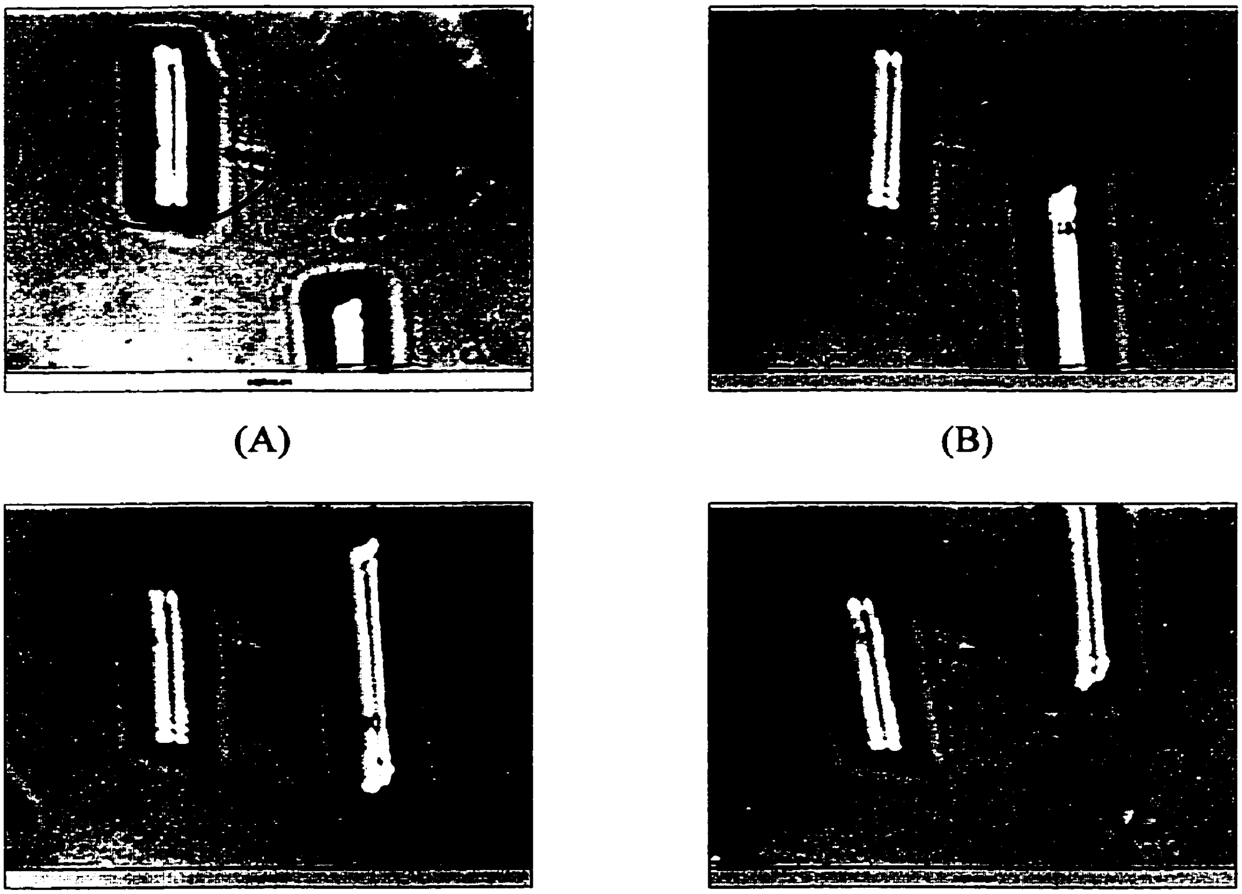
(G)



(H)

Figures 6.1.8C-6.1.8H: ALIGNMENT AND MOVEMENT OF A 2.5 μm RADIUS, 15 μm LENGTH CYLINDER.

Figure 6.1.8 demonstrated that it is possible to translate a cylinder while it is vertically trapped in the beam. It is also possible to translate a cylinder while it is trapped horizontally. Figure 6.1.9 shows a cylinder with a 2.5 μm radius and length of 13 μm trapped in the beam. The cylinder is being translated around a neighboring cylinder.



(A) (B) (C) (D)

Figure 6.1.9: TRANSLATION OF A 2.5 μm RADIUS, 25 μm LENGTH CYLINDER AROUND ANOTHER CYLINDER.

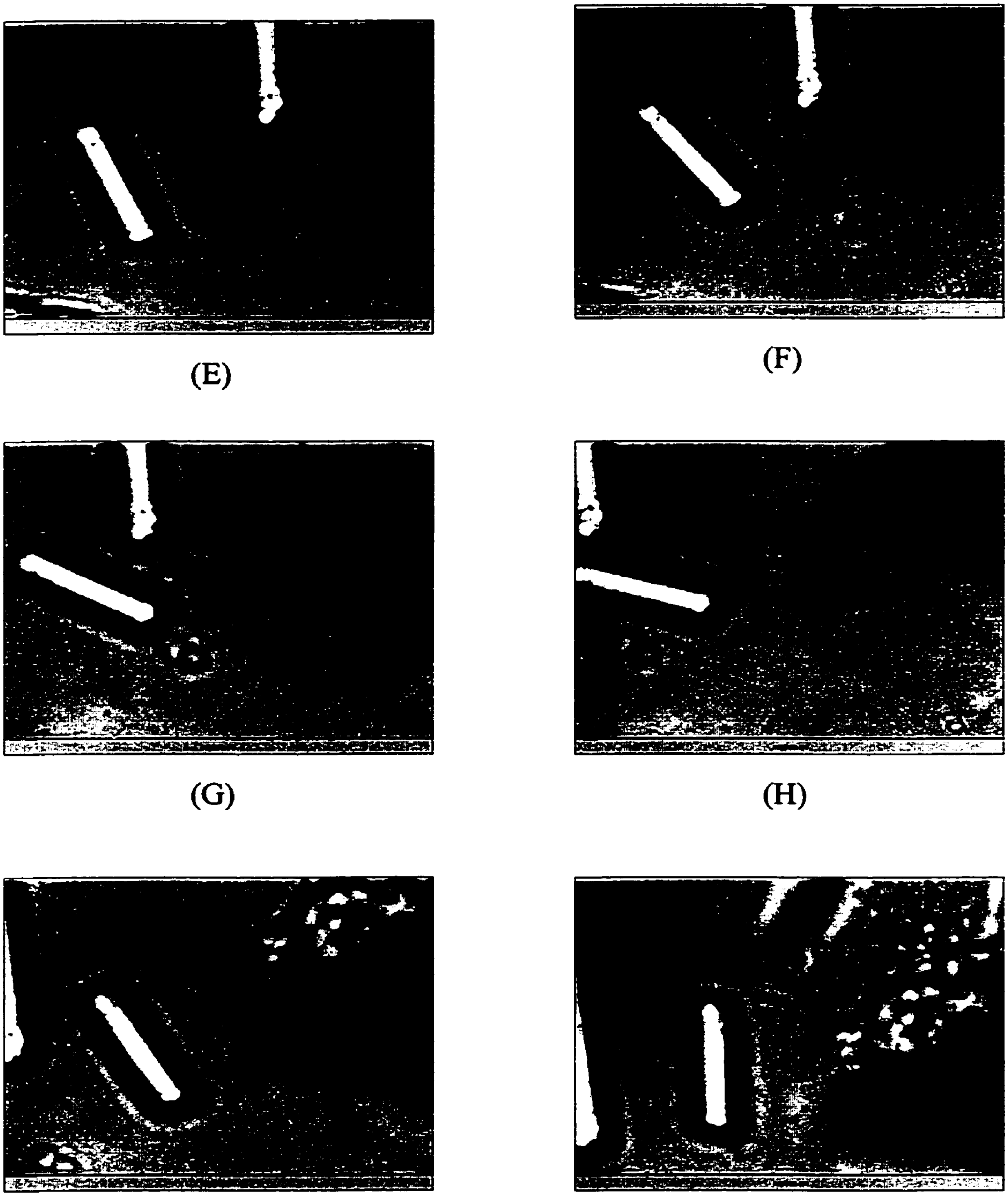


Figure 6.1.9: TRANSLATION OF A 2.5 μm RADIUS, 13 μm LENGTH CYLINDER AROUND ANOTHER CYLINDER.

This section has shown that the cylinders will line up along two possibilities, straight up and down as in figure 6.1.4 and on its longest axis as in figures 6.1.3, 6.1.5 and 6.1.8. This verifies the theoretical data that was presented in the previous section. The theoretical angle at which the cylinder is stable was also shown to coincide with the calculated angle from the experiments. It has also been shown that it is possible to manipulate the cylinder when it is either standing up or on its side. These features are important when it comes to building a micro-machine. The next section will look at the torque versus angle that is produced on a cylinder when one of five parameters is changed over a specified range.

6.2 Design of Cylinder

The previous section has shown that it is physically possible to move and align cylinders. This illustrates that cylinders are feasible to be used as posts that will be able to secure gears for the development of micro-machines. The gear is the mechanical component of a system. If a cylinder is secured to a post such that it is still free to rotate then the cylinder can be thought of as a micro-rotor.

It was shown in figure 6.1.4 that the cylinder was drawn into the beam before it started to vertically align itself. If this cylinder was pinned down but free to rotate, then the cylinder would just spin around this point. The continuous rotation of a cylinder is now important, as it would demonstrate that it would be feasible to produce a micro-rotor, which would be the first step to a micro-actuator.

To produce a rotor with the best design the cylinder will be the starting point where several factors will be looked at that will give the optimal torque as it rotates. The factors that will be considered are the dimensions of the cylinder, the radius of the end-caps and the beam waist.

The optimal design of the cylinder requires the least amount of power to rotate it but generates the maximum amount of torque. This would mean that the power supply to run the machine would last longer making it more cost efficient and convenient as the supply would not have to be changed or replaced as quickly.

To find this optimal design the computer program was modified to include a subroutine that rotates the cylinder at 1° intervals and then calculates the torque the beam would apply to the cylinder. The program is designed to modify one of five possible parameters at a time; beam waist, radius of the cylinder, length of cylinder, top end cap and bottom end cap. The subroutine returns the data, which is stored in a file where the program EXCEL can be used to plot the results. The code for the subroutine is in Appendix C.1.

The first parameter to be modified was the beam waist. Figure 6.2.1 is a surface plot of the torque versus angle for beam waists ranging from 0.5 to 3 μm. The parameters of the cylinder are in the brackets in the title where R represents the radius of the cylinder, L is the length, LEC is the lower end cap radius and UEC is the upper end cap radius.

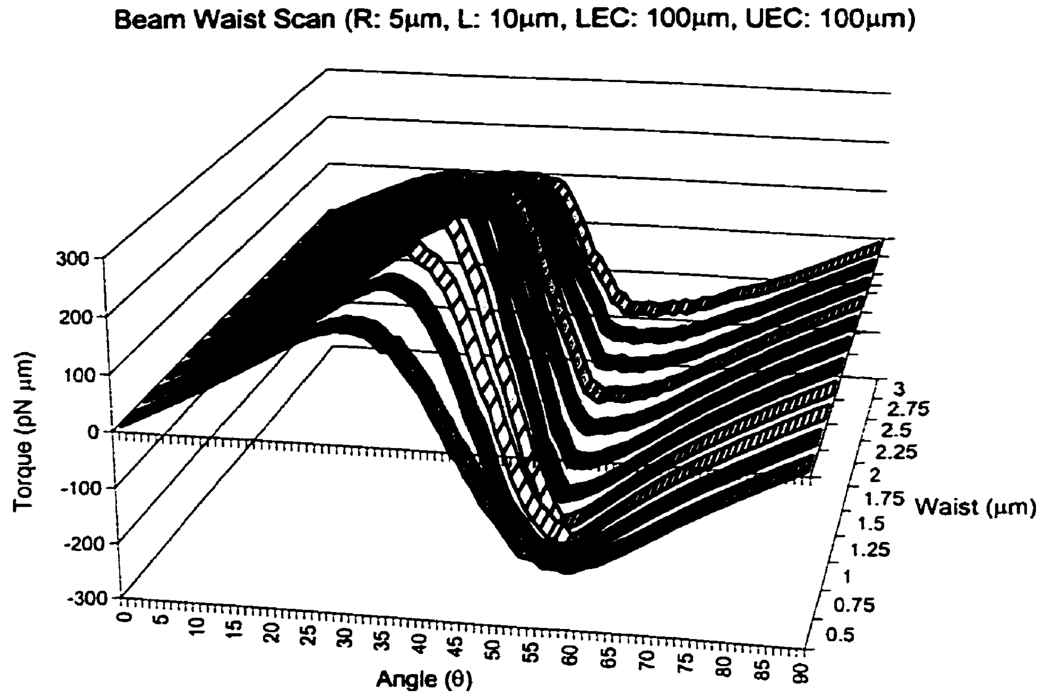


Figure 6.2.1: Data of torque versus angle for beam radii ranging from 0.5 to 3 μm in width.

Each beam width was then plotted separately also a trend line of a running average over a 45° span was found and plotted on the graph. The 45° span is used since a symmetric rotor of eight points has a separation of 45° or 4 beams at 45° can be used to rotate a single cylinder. The running average will determine the greatest torque that will be imparted to the cylinder. Figures 6.2.2 and 6.2.3 are plots of the 0.5 μm and 3 μm waist data taken from the above graph.

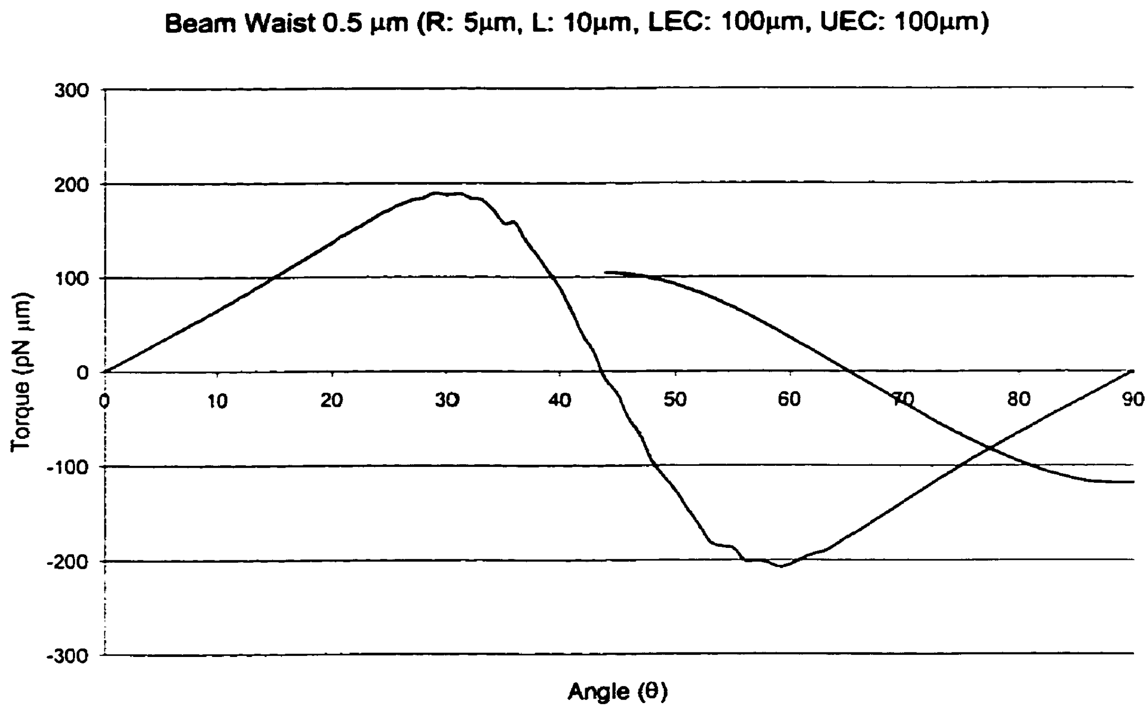


Figure 6.2.2: 0.5 µm BEAM WAIST SCAN OF TORQUE VERSUS ANGLE

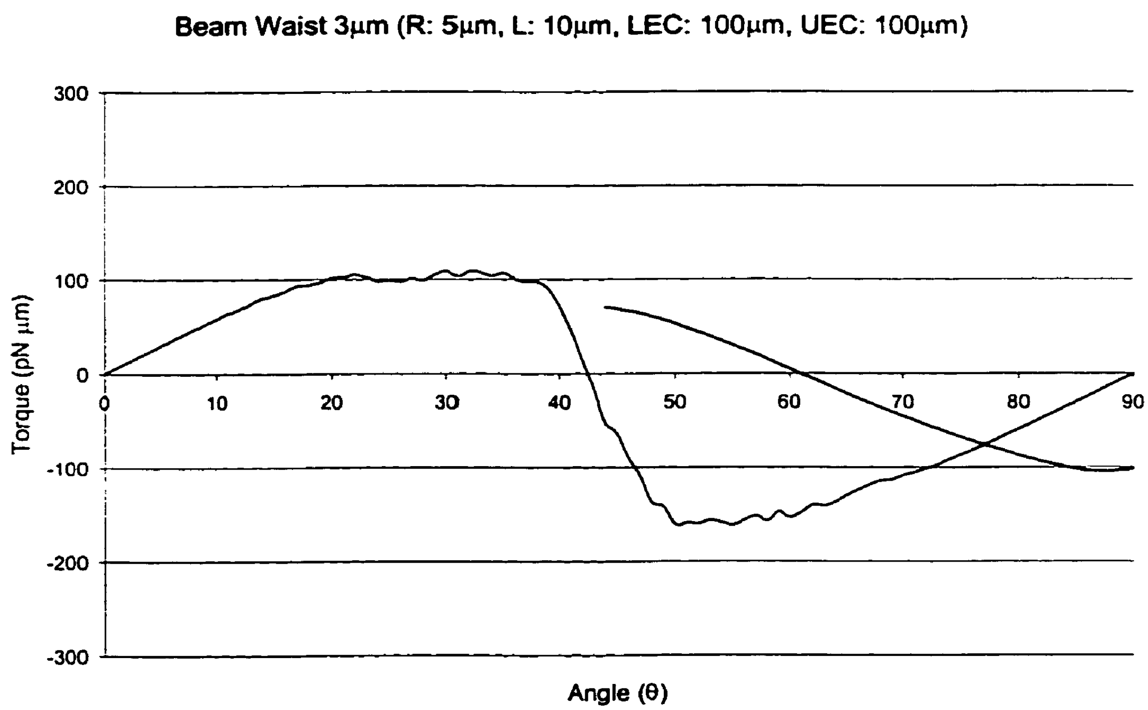


Figure 6.2.3: 3.00µm BEAM WAIST SCAN OF TORQUE VERSUS ANGLE.

The rough plot in both figures could have been smoothed out if more rays were used in the simulation but the increase of rays would cause the simulations to take longer to run. To save time an optimal number of rays was used. The beam waist that was chosen to produce the greatest torque was found to be 1.25 μm. The plot of this waist is shown below in figure 6.2.4. All other relevant plots can be found in Appendix C.2.

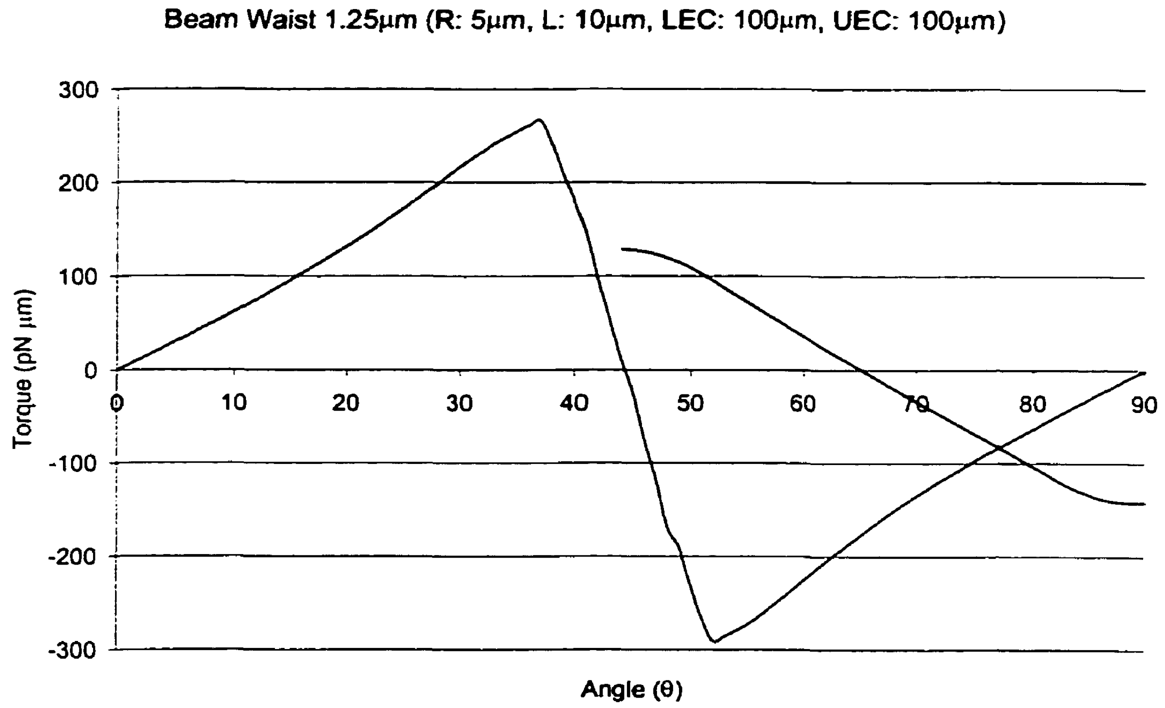


Figure 6.2.4: 1.25 μM BEAM WAIST SCAN OF TORQUE VERSUS ANGLE.

The plot of the 1.25 μm beam waist has two notable features. The first is that the curve has the highest and lowest torque values. The greatest torque value corresponds to a torque that would push the cylinder out of the beam while the negative torque would pull the cylinder back into the beam. The second point is that the slope of the average trend line is the largest as compared to the other plots.

After the best size of the beam waist was determined another parameter was modified. The parameters that were kept constant were; beam waist (B.W.) = 1.25 μm; radius (R) = 5 μm; the lower end cap (L.E.C.) = 100 μm; the upper end cap (U.E.C.) = 100 μm. The modified parameter was the length (L), which was varied from 5 μm to 100 μm. There were several designs that gave the greatest torque. Starting at 70 μm the average trend line and the data line are almost identical with the cylinders ranging up to 100 μm in length. This can be seen in the plot of all the data shown on the next page. At the arrow a minimum value is reached which is maintained approximately the same from 70 to 100 μm. For the optimal design of the cylinder the length of 70 μm was chosen. This length was chosen as it was found that in the samples used there were more cylinders in the range of 40 to 70 μm. There were few, if any, with lengths greater than 70 μm. The plot of the torque versus angle for the 70 μm cylinder length is shown in figure 6.2.6. Other plots can be found in Appendix C.3.

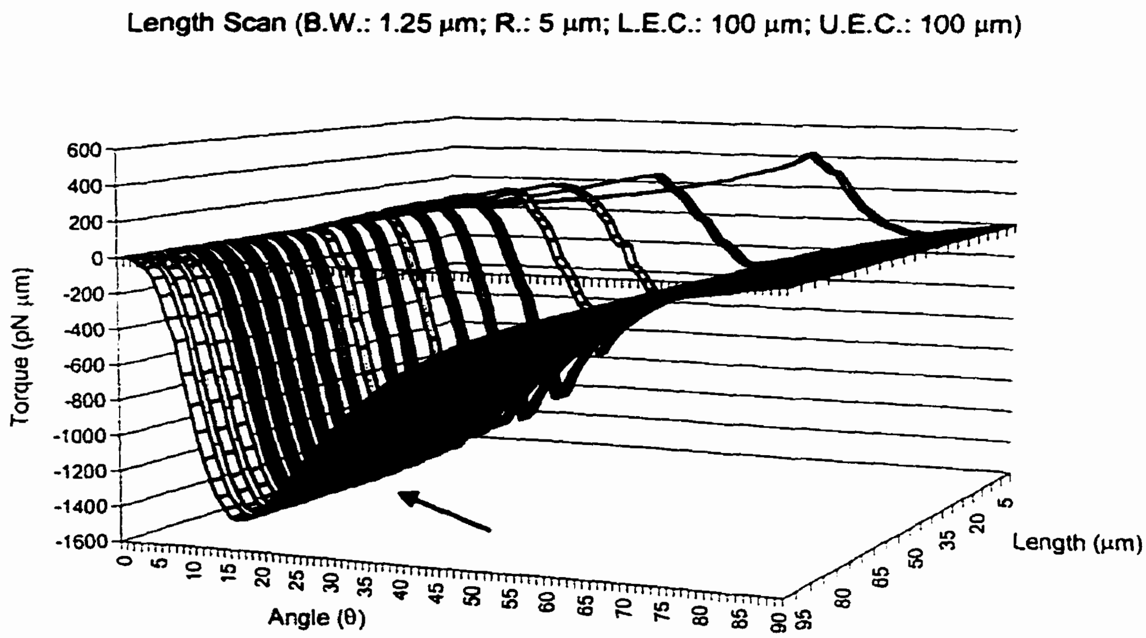


Figure 6.2.5: PLOT OF LENGTH SCAN FROM 5 μm TO 100 μm.

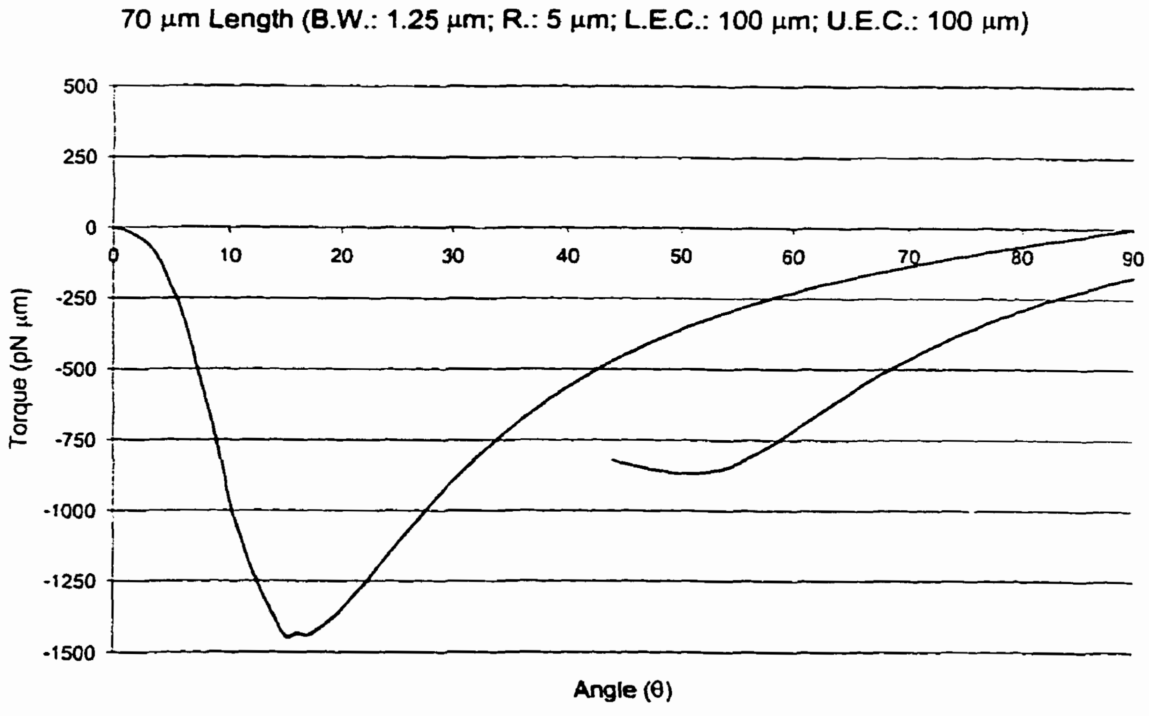


Figure 6.2.6: TORQUE VERSUS ANGLE FOR A CYLINDER OF LENGTH 70 μm.

Now that the beam waist has been chosen and the optimal length has been determined, the beam waist was varied a second time with the length at 70 μm instead of 10 μm. This was done to ensure that the beam waist of 1.25 μm was the optimal value. It is possible that another beam waist might produce the greatest torque for the 70 μm length. After running the simulation, it was found that the 1.25 μm and the 1.50 μm waist have very similar plots, figure 6.2.7 shows the 1.25 μm plot. Neither of these plots produce the greatest torque but they both have smooth curves.

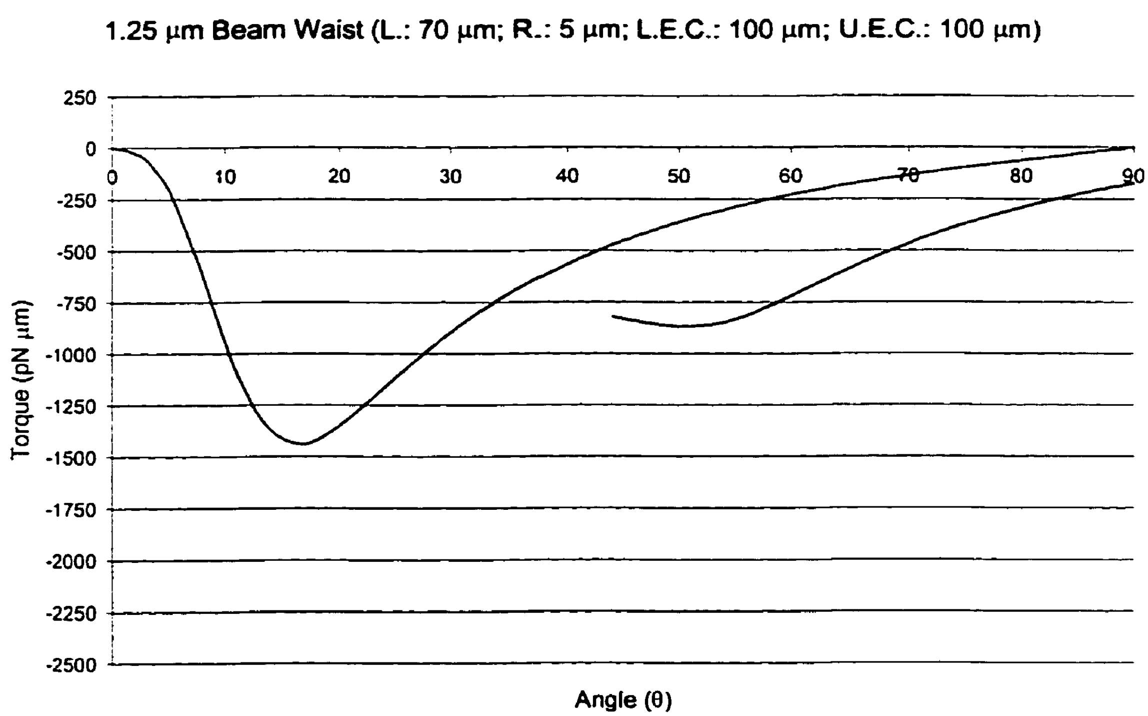


Figure 6.2.7: PLOT OF OPTIMAL BEAM WAIST FROM SIMULATION.

The curve in figure 6.2.7 is smoother when compared to the 1.50 μm curve of figure 6.2.8. The importance is that the laser has to be modulated to produce a continuous torque on the cylinder and the smoother the curve, the easier it will be to adjust the laser accordingly. Therefore, 1.25 μm is the ideal beam waist for the continuous rotation of the cylinder.

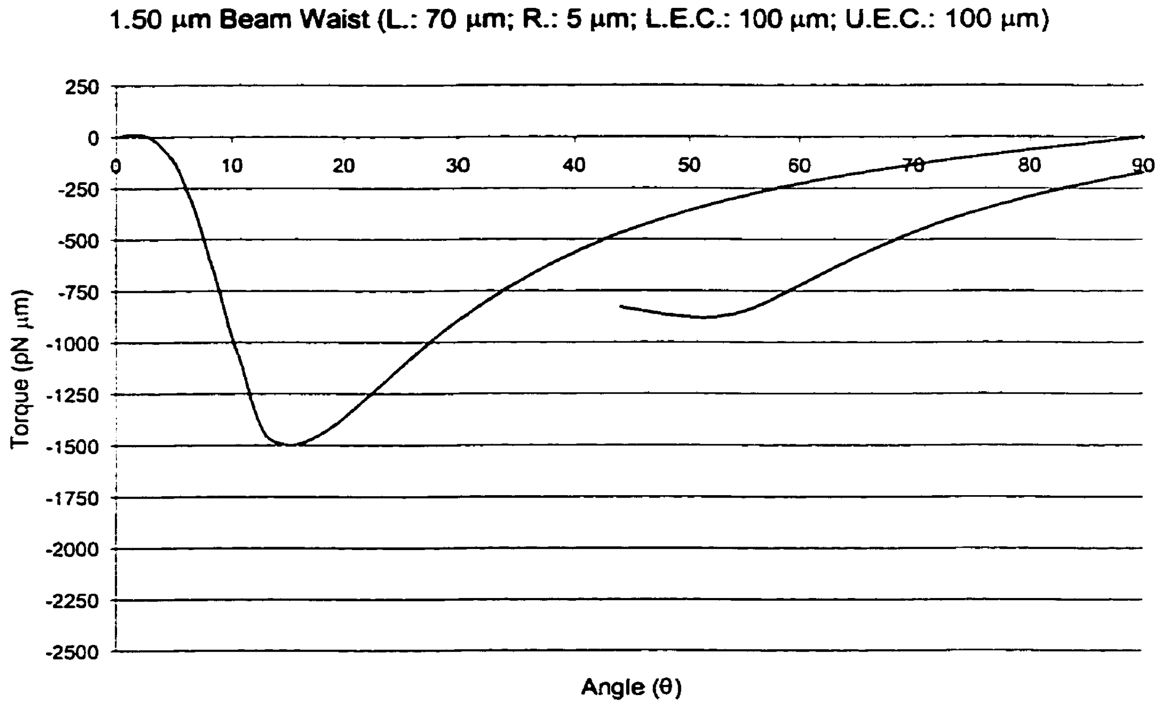


Figure 6.2.8: TORQUE VERSUS ANGLE FOR THE 1.50 μm BEAM WAIST.

The next parameter to be modified is the radius of the cylinder. It was found that a radius of 15 μm would produce the greatest average torque. The plot is shown below in figure 6.2.9 but it is not the ideal radius for a micro-gear. The reason will be discussed in the next section. The chosen radius is 10 μm, which is shown in figure 6.2.10. Appendix C.4 has other relevant plots of the radius scans.

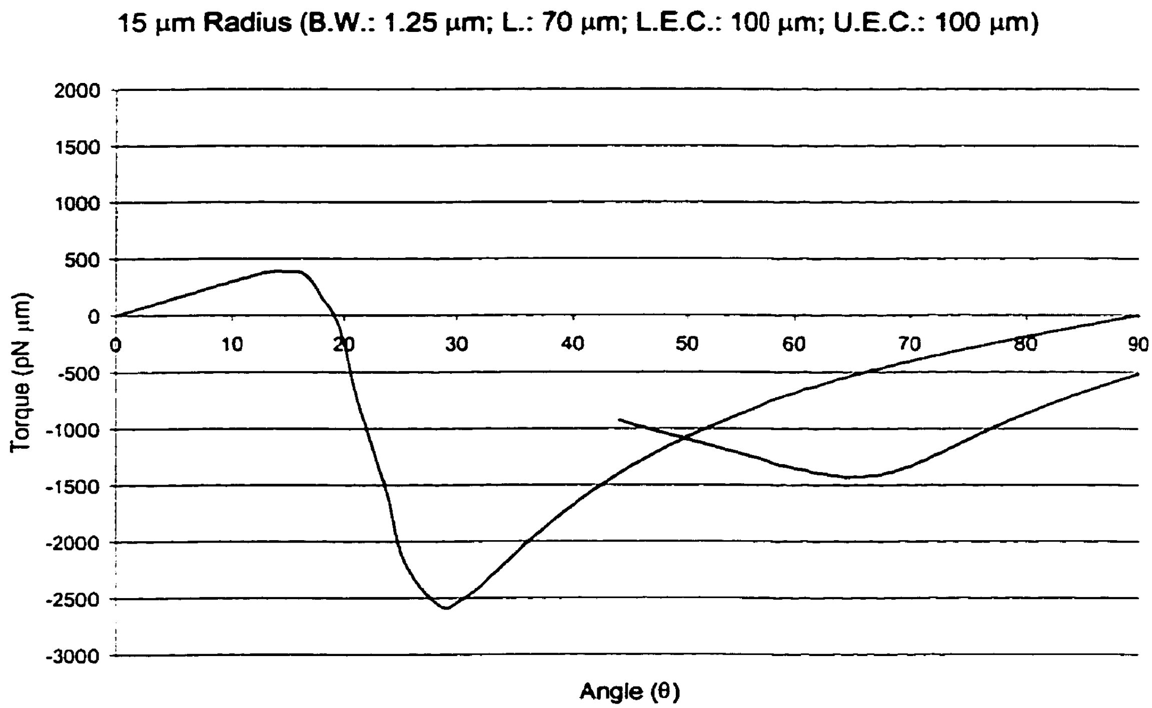


Figure 6.2.9: PLOT OF TORQUE VERSUS ANGLE FOR THE 15 μm RADIUS CYLINDER.

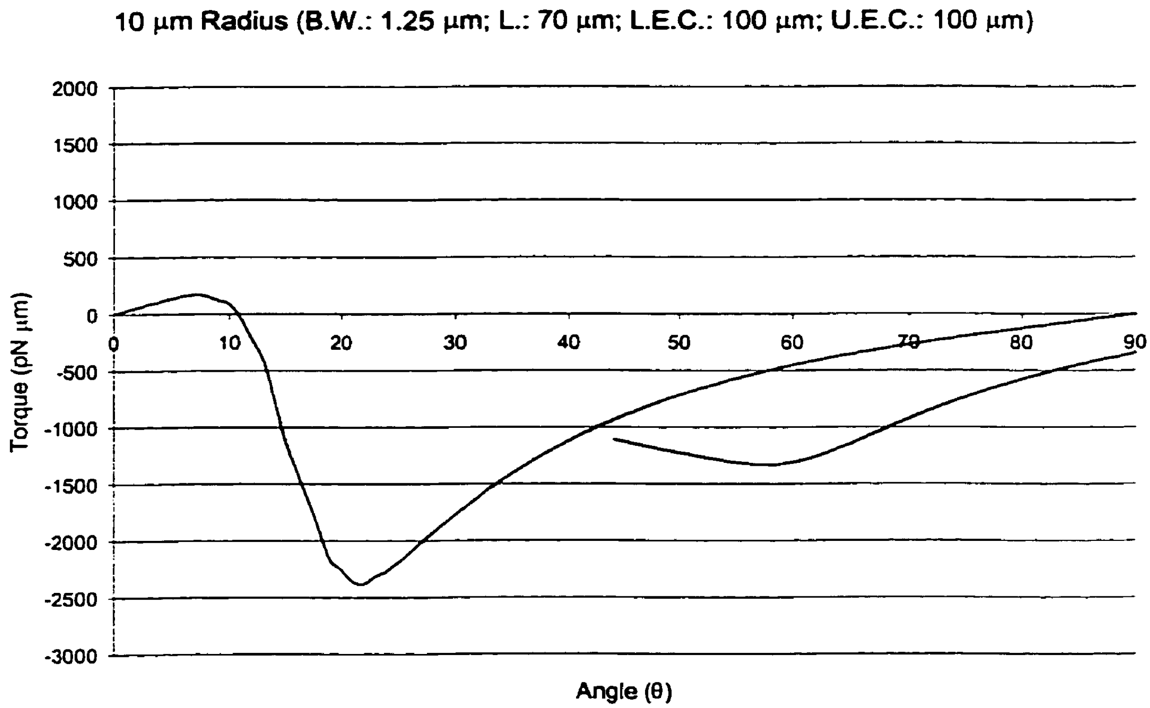


Figure 6.2.10: PLOT OF DATA FOR A 10 μm RADIUS CYLINDER.

For the ideal cylinder the parameters of a length of 70 μm, radius of 10 μm with both end-caps having a radius of 100 μm (Flat ends) and a beam radius of 1.25 μm is desired. The flat ends of the cylinder are desired, as the design of a micro-gear is more difficult with round end-caps.

For completeness of the theoretical design the end-cap radii were modified. The lower end-cap was first modified where it was found that the best end-cap radius was 10 μm. For the upper end-cap radius it was found that the radius of 15 μm was the optimal. Both plots can be seen on the next page in figures 6.2.11 and 6.2.12. Appendices C.5 and C.6 have other relevant plots of the data for the lower and upper end-caps.

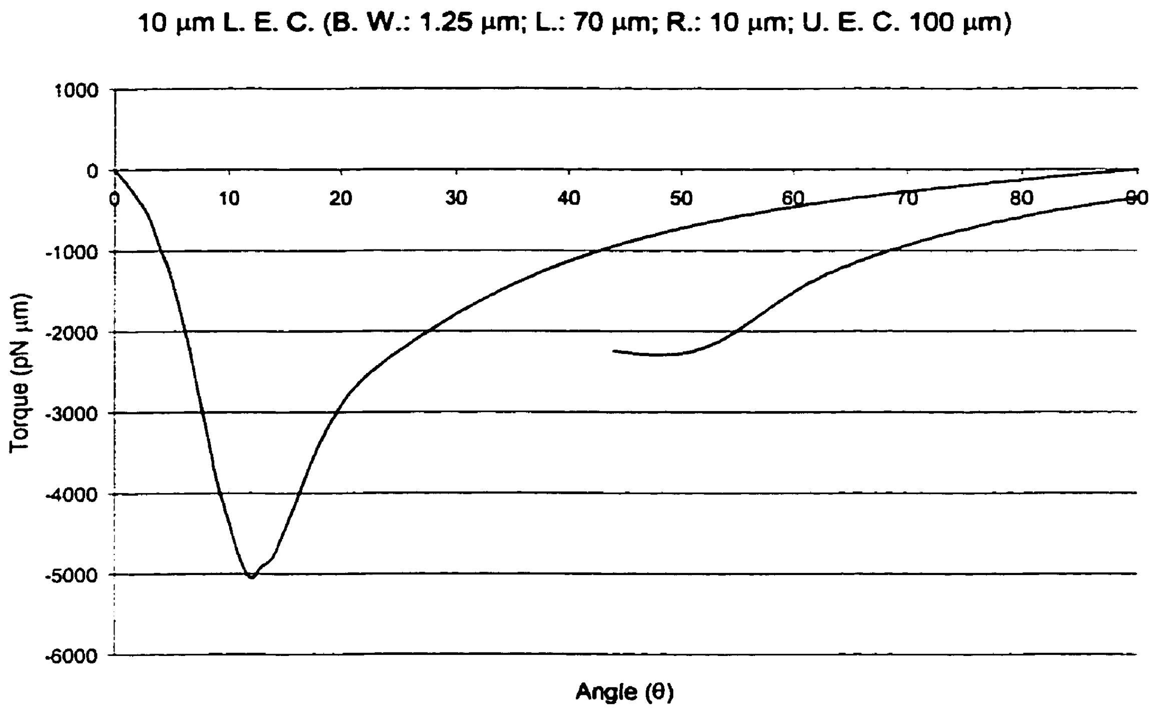


Figure 6.2.11: OPTIMAL RADIUS OF 10 μm FOR THE LOWER END-CAP.

The 15 μm radius produces the greatest torque but the different radii of the end-caps will cause a problem in the design of a micro-rotor given that it is not symmetric. For this reason the radius of the upper end-cap of 10 μm is better suited for the design of the micro-rotor. The 10 μm plot is shown in figure 6.2.13.

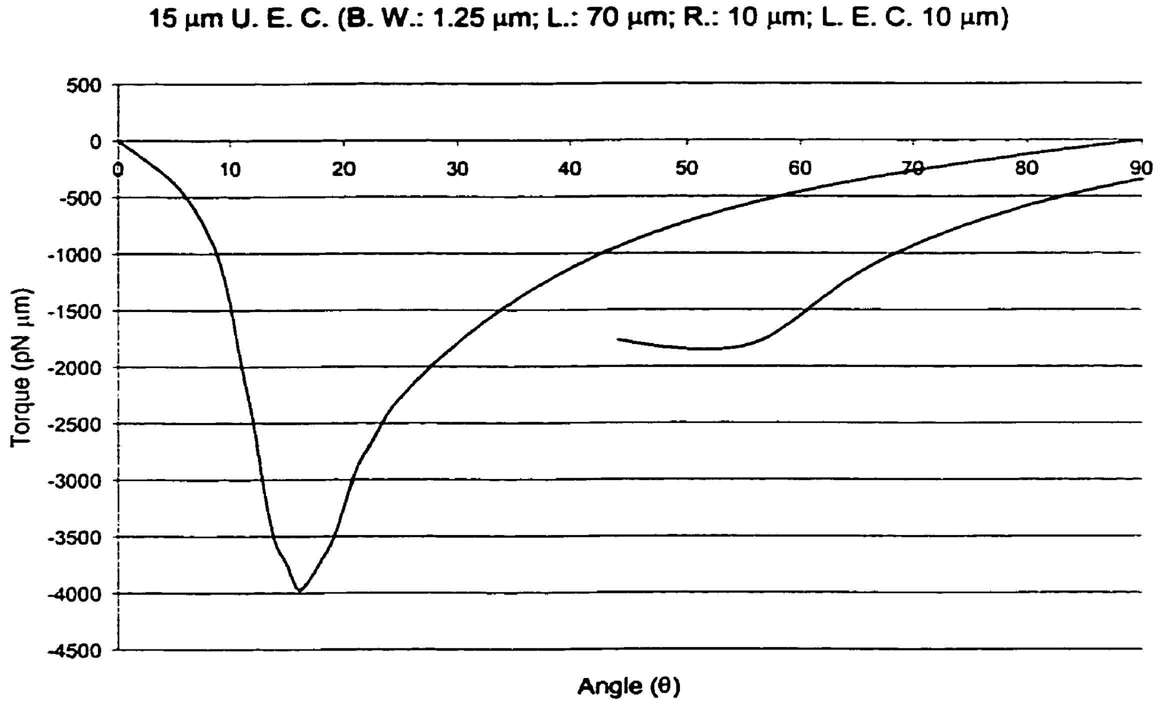


Figure 6.2.12: OPTIMAL RADIUS OF 15 μm FOR THE UPPER END-CAP.

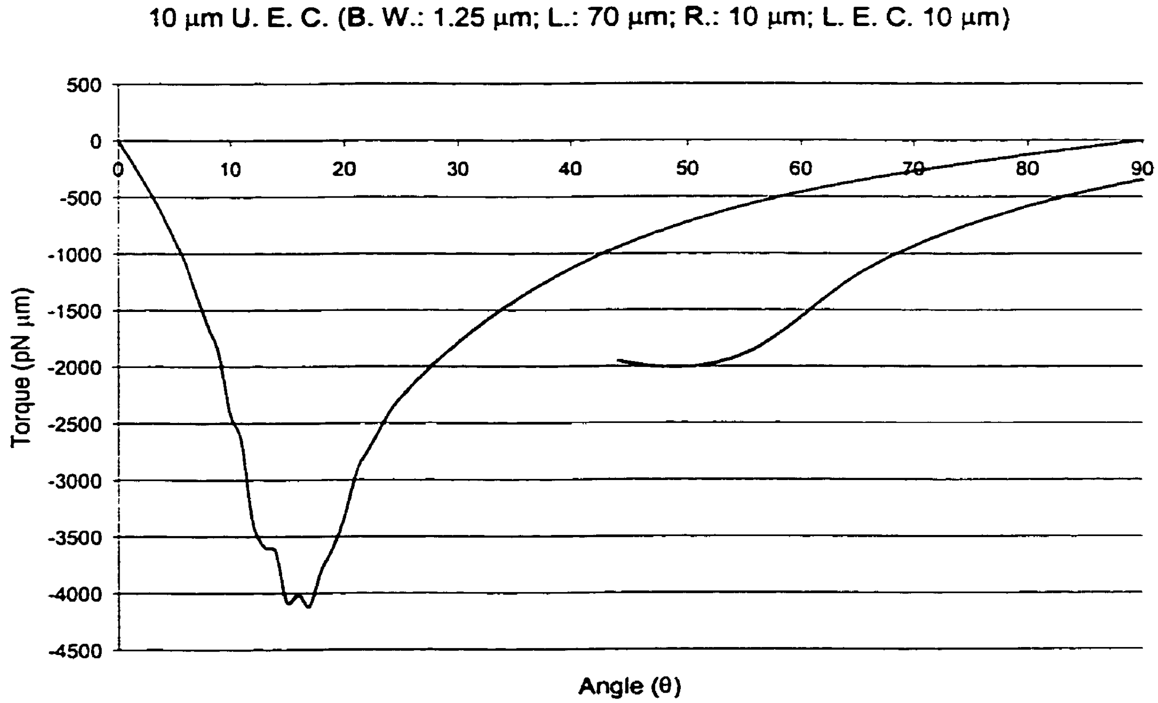


Figure 6.2.13: PLOT OF 10 μm UPPER END-CAP RADIUS SCAN.

One of the final goals of building a micro-actuator is to have its rotors rotating at a continuous speed and this is the topic of the next section.

The overall best design of a cylinder based rotor has been determined to be one with a length of 70 μm, radius of 10 μm, lower end-cap of 100 μm, upper end-cap of 100 μm in a beam with a radius of 1.25 μm.

6.3 Continuous Rotation of Cylinder

In the previous section it was mentioned that the cylinder radius of 10 μm was chosen over the 15 μm radius. The explanation for this choice can be justified from equation 3.2.18.

$$\frac{d\gamma(t)}{dt} = \frac{\tau}{b}\left(1 - \exp\left(-\frac{b}{I}t\right)\right) \qquad (3.2.18)\ldots(6.3.1)$$

This is the angular velocity equation derived in chapter 3. For a micro-actuator the rotors would be required to attain its maximum angular velocity as quickly as possible. For this reason the smaller radius of the cylinder is desired because the moment of inertia as well as the damping factor are smaller for the 10 μm radius than the 15 μm.

Another factor to consider is the power required in maintaining a constant rotational rate. The torque magnitude is linearly related to the incident laser beam power. By rearranging the equation, the torque (τ) required to rotate the cylinder can be determined provided the angular velocity, damping factor (b) and the moment of inertia (I) are known.

$$\tau(t) = \frac{b\frac{d\gamma(t)}{dt}}{1-\exp\left(-\frac{b}{I}t\right)} \tag{6.3.2}$$

From this the power versus time curve for a constant rotation rate can be plotted. Using the main program designed by Gauthier, the required torque will be calculated from equation 6.3.2 by specifying a rotational velocity in revolutions per second. Based on the current rotational angle of the cylinder the program will calculate the actual torque present for this orientation, $\tau_{computed}$, when the laser beam has a power of 100 mW. Forming a ratio of these two torque values the power required from the beam in order to maintain a constant rotational rate can be obtained.

$$P_{required} = \frac{\tau(t)}{\tau_{computed}} 100mW \tag{6.3.3}$$

When the cylinder starts at rest $(t=0)$ the denominator of equation 6.3.2 will be zero implying that there would have to be an infinite amount of torque required to instantaneously start the cylinder rotating at the desired rotational velocity. From equation 6.3.3 it also indicates that an infinite power from the laser would be required to generate the infinite torque value.

The program designed by Gauthier has a sub program included that would use a four-beam system to continuously rotate a micro-cylinder. An angle of 45° offsets the four beams as shown below in figure 6.3.1.

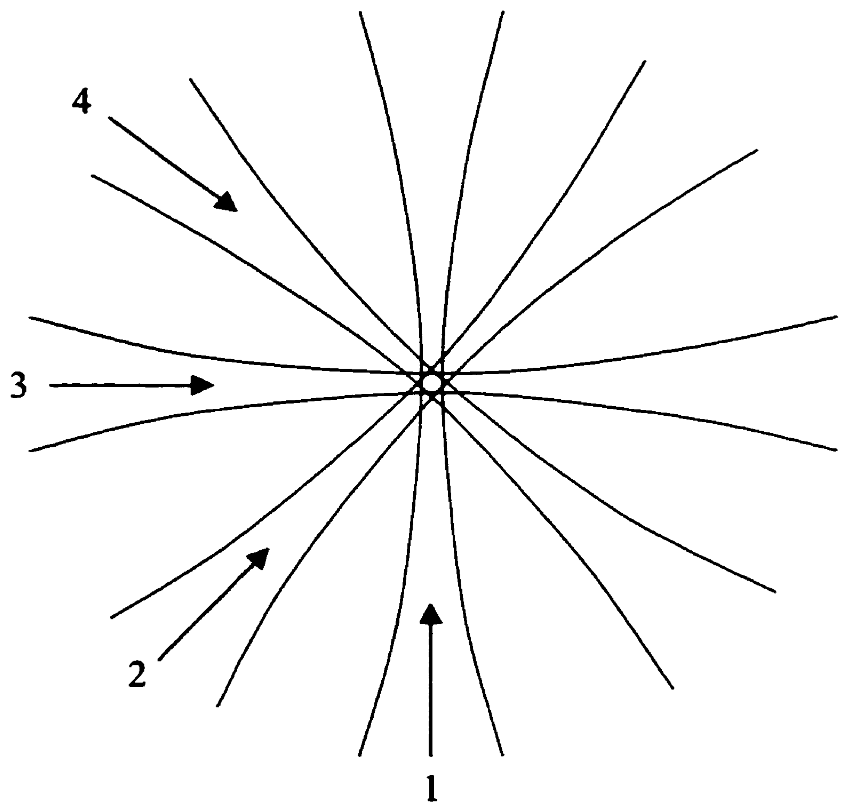


Figure 6.3.1: MULTIPLE BEAM CONFIGURATION FOR CONTINUOUS ROTATION OF A MICRO-CYLINDER.

The program is designed such that the laser beam that is activated is dependent upon the orientation of the cylinder. The orientation is defined from the plots given in section 6.2. From figure 6.2.9 the greatest absolute average torque occurs at 58°. Subtracting 45° from the 58° gives the initial value of 13°, which is the beginning value for the average given to 58°. The program is designed such that between the range of 13° and 58° the laser marked as 1, will be turned on. This continues in the pattern that between 58° and 103° laser two is turned on then between 103° and 148° laser three is activated, 148° and 193° laser four is activated. This sequencing based on cylinder orientation continues on for the full 360° rotation then repeats again for additional rotations.

Three other values are included into the calculation for the power required to rotate the cylinder at a constant speed. The first is the velocity that the cylinder is required to rotate. The chosen velocity for the modeling is 2 revolutions per second. The next parameter is the limiting power of the laser. The final parameter is the damping factor, which is defined in units of inertia. The values of the damping factor used are 1, 10, 100, 1000 and 5000 times the moment of inertia (I). The cylinders used for these simulations have a length of 70 μm, a radius of 10 μm and the end-caps have radii of 100 μm.

Starting with the damping factor of 5000I, figure 6.3.2 is a plot of the power required to rotate the cylinder at a continuous angular velocity of 2 revolutions per second. It can be seen that to start the cylinder from rest to an instantaneous angular velocity of 2 revolutions per second the power required would have to be in the order of $2x10^9$ mW or 2 MW.

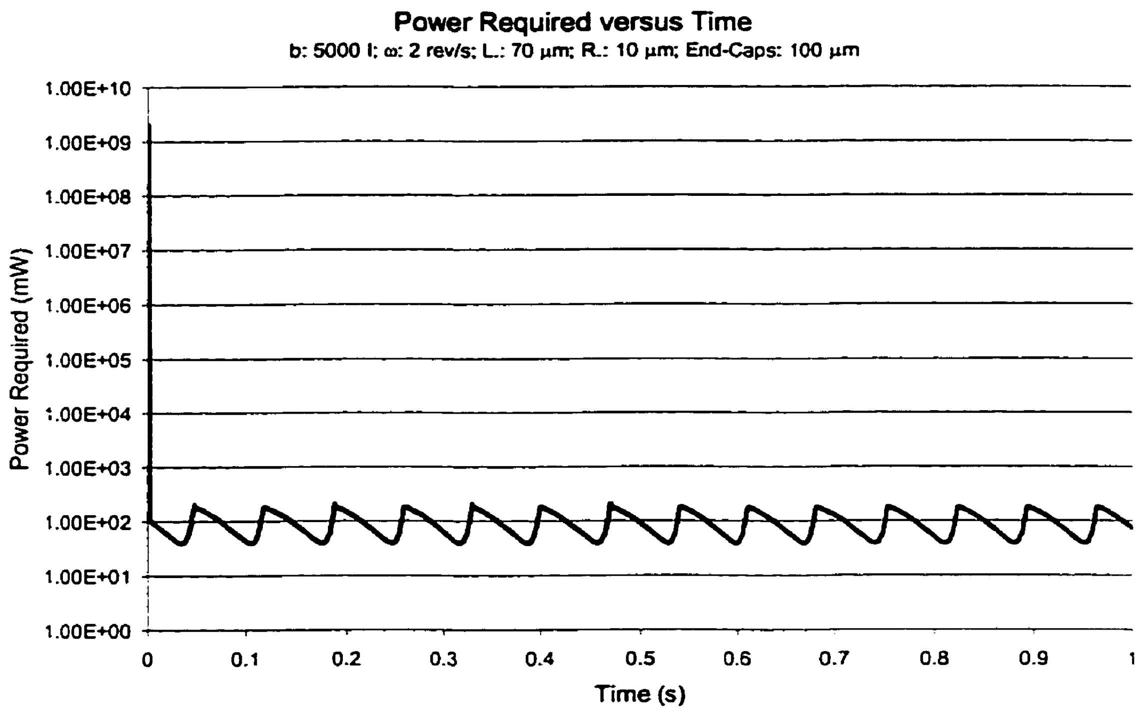


Figure 6.3.2: POWER REQUIRED TO ROTATE THE CYLINDER AT AN ANGULAR VELOCITY OF 2 REVOLUTIONS/SECOND.

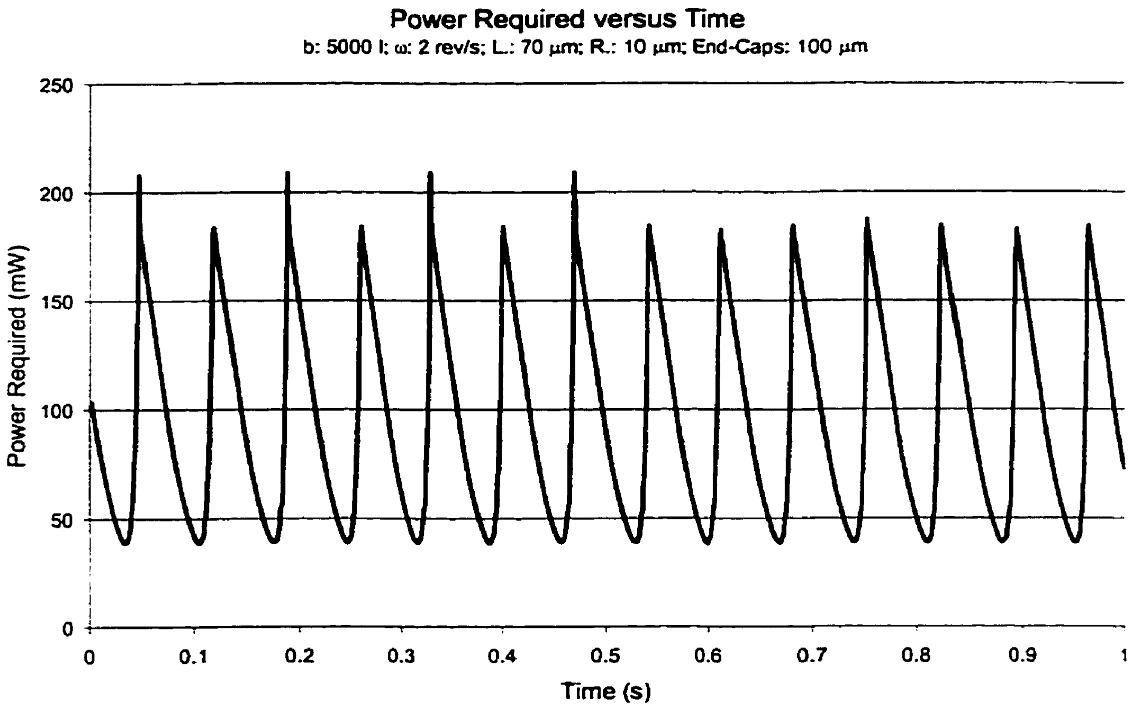


Figure 6.3.3: ENLARGED PLOT OF POWER REQUIRED VERSUS TIME .

By reducing the scale of the power required to 250 mW (figure 6.3.3) it can be seen that the power needed to rotate the cylinder decreases rapidly, within 0.001 s. Each peak represents the initialization of a different laser beam. The first peak would be the second laser while the second peak is the third laser and so on.

Typically in any application the power available to the actuation of the rotor will be limited. Figure 6.3.4 demonstrates the power available to the actuation when clipped at the upper level of 100 mW.

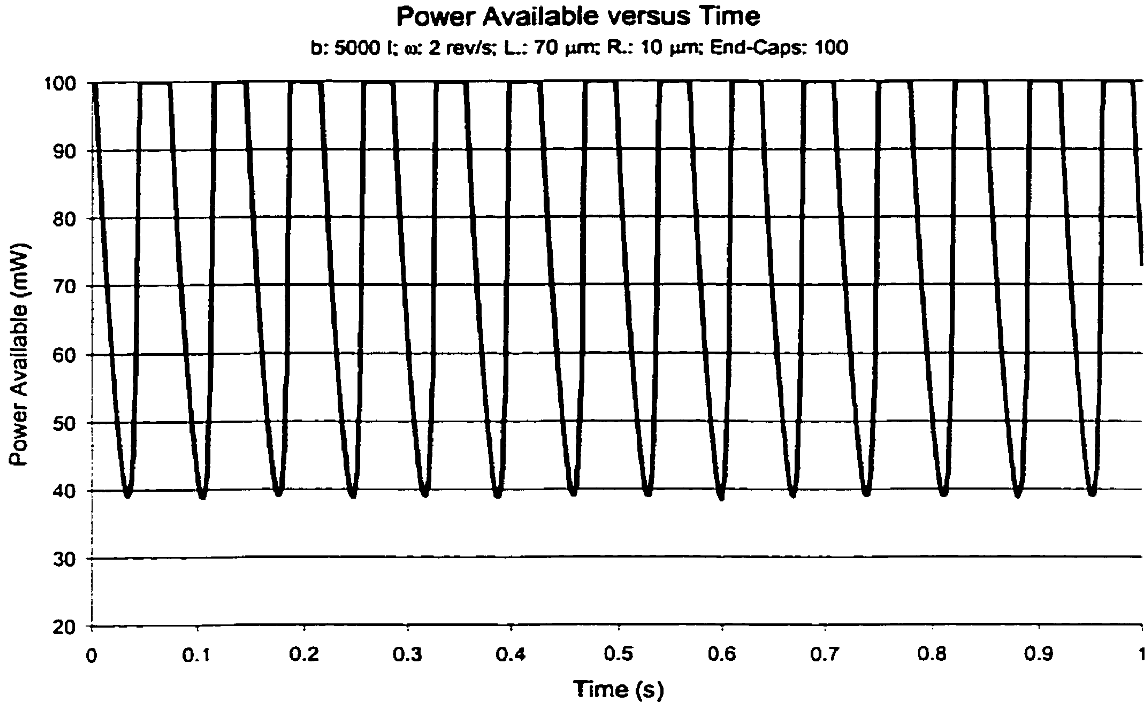


Figure 6.3.4: POWER AVAILABLE TO MAINTAIN 2 REVOLUTIONS PER SECOND.

It can be seen that it takes 0.002 s before the cylinder is rotating at 2 revolutions per second. The damping factor is also too great for the cylinder as the power available clips at 100 mW periodically.

By reducing the damping factor from 5000I to 1000I the initial power to instantaneously accelerate the cylinder to 2 revolutions per second is still in the order of 2×10^9 mW but the power afterwards never goes over 40 mW as shown in figure 6.3.5. The power does not "clip" and 2 revolutions per second is maintained.

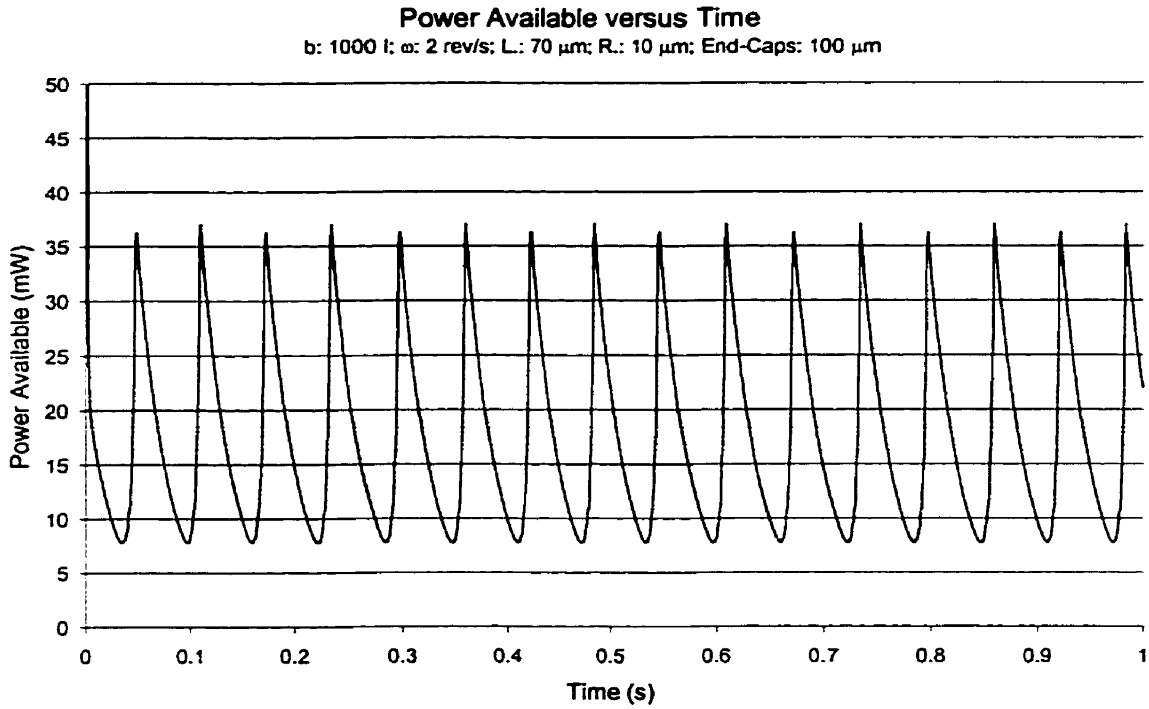


Figure 6.3.5: POWER VERSUS TIME FOR 1000I DAMPING.

When the damping factor is further reduced to 100I the power available is shown in figure 6.3.6. The reduction of the damping factor by an order of magnitude reduces the power available by an order of magnitude as well. If the damping factor is reduced to 10I and then to 1I it can be seen that the power also reduces by the same. See figures 6.3.7 and 6.3.8. The plots 6.3.4 to 6.3.8 demonstrate that the power available linearly scales with the damping factor for large t. This observation is expected. The limit of the torque in equation 6.3.1 as $t \rightarrow \infty$ is $\tau = b\frac{d\gamma(t)}{dt}$ where $\tau \propto P_O$. The initial power to instantaneously start the cylinder at 2 revolutions per second remians the same in all plots at $2x10^9$ mW.

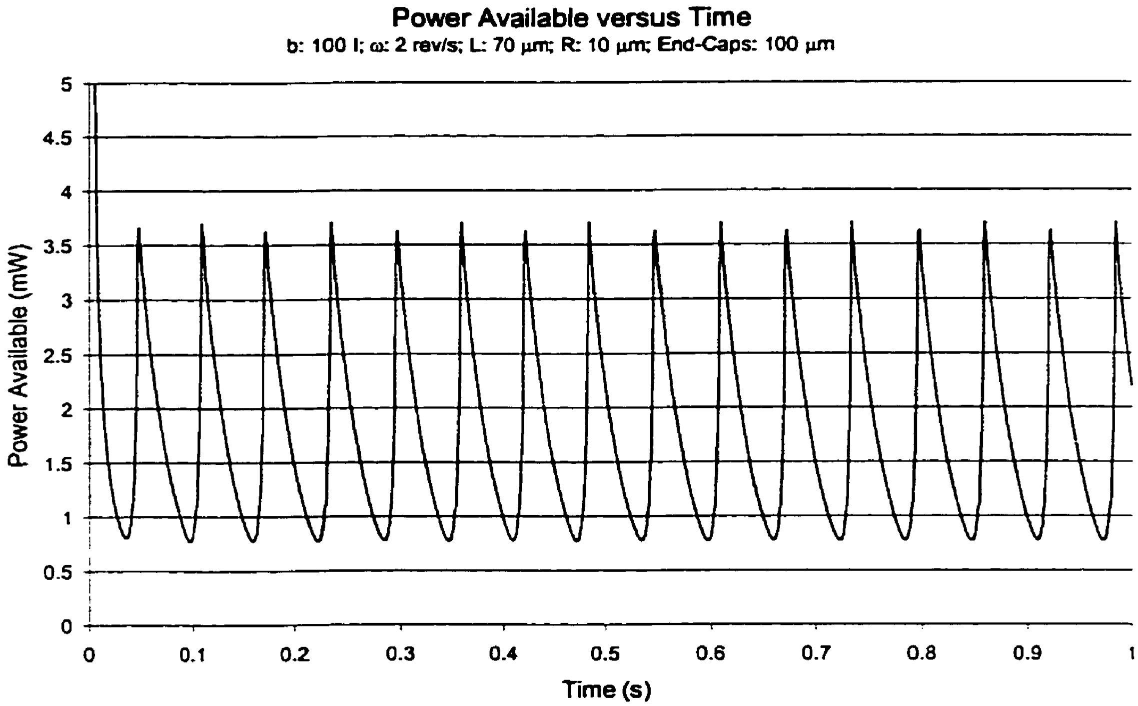


Figure 6.3.6: POWER AVAILABLE TO MAINTAIN 2 REVOLUTIONS PER SECOND.

The damping factor was calculated to be $2.3x10^{-16}$ Nm^3 for the described micro-cylinder in water using equation 3.3.22, $b = 6\pi\eta rL^2$. It is then possible to find the power needed and power required starting and maintaining a cylinder at any angular velocity in any medium used. The four beam system has been considered plausible as A. Ashkin [21] placed a patent on a design which uses four laser beams aligned in the configuration shown in figure 6.3.10.

In future, the next step is to produce a micro-rotor that can be tested using one laser beam. It has been simulated for an eight-point rotor, shown below, where the one laser beam was sufficient enough to rotate the rotor. Also, it is believed that the testing of a real micro-rotor can be achieved in the very near future. It is believed that a micro-actuator is not too far away from creation as well which will lead into a micro-system.

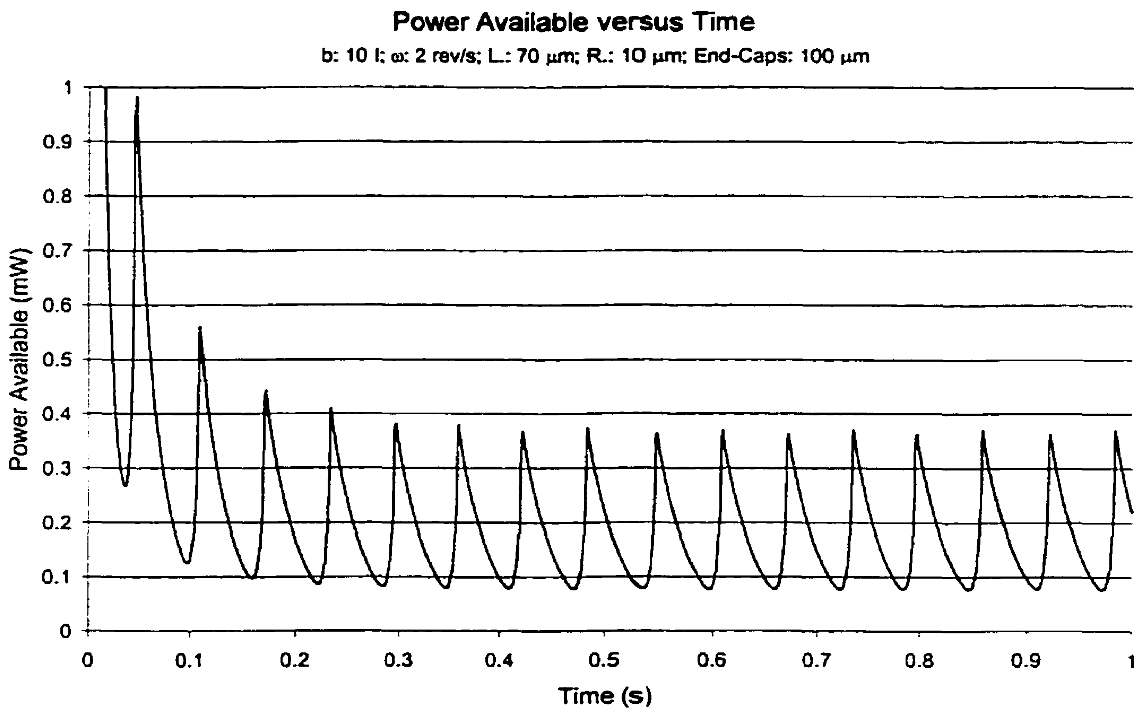


Figure 6.3.7: POWER AVAILABLE TO MAINTAIN 2 REVOLUTIONS PER SECOND FOR 10I.

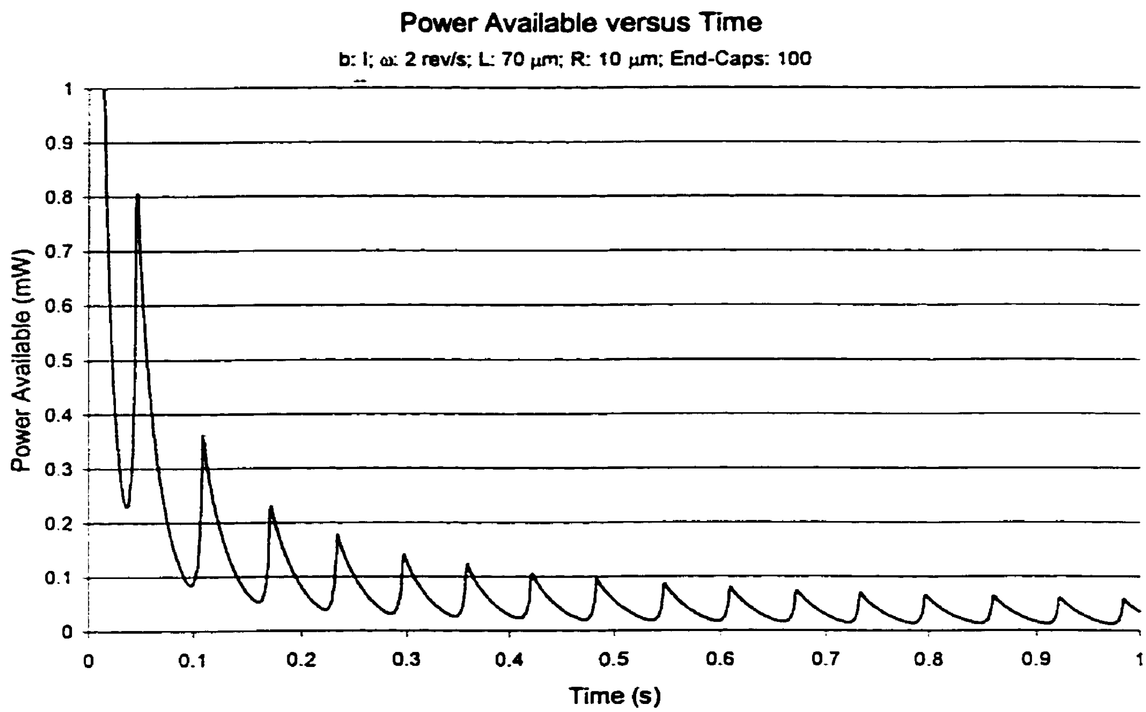
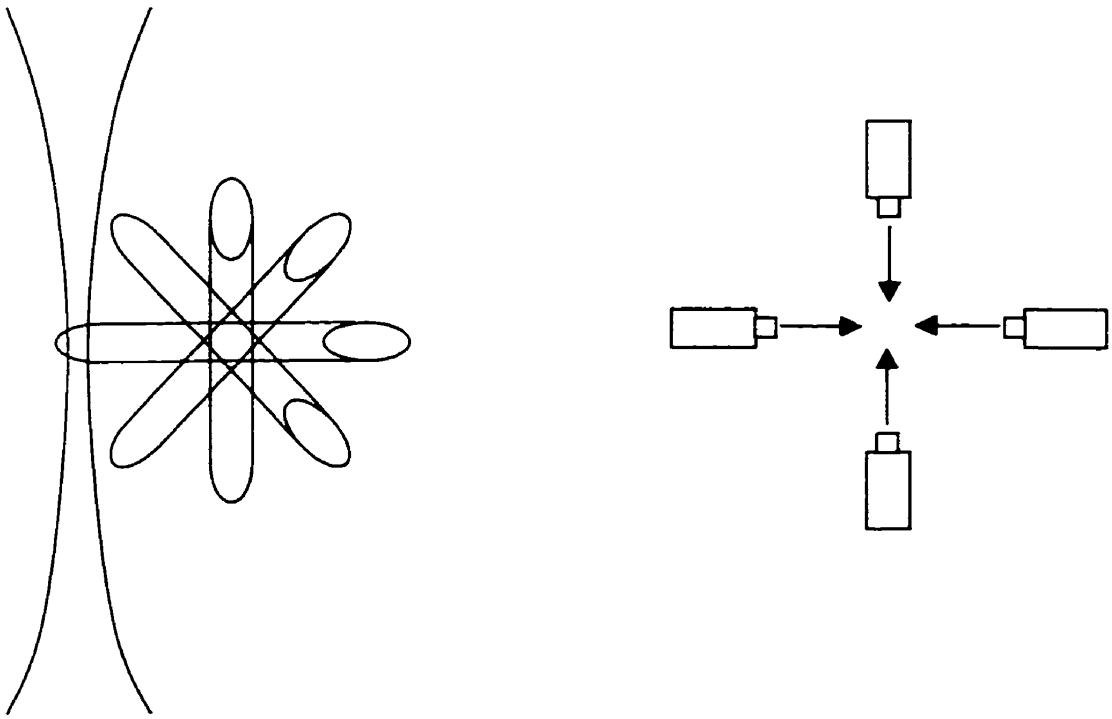


Figure 6.3.8: POWER AVAILABLE TO MAINTAIN 2 REVOLUTIONS PER SECOND FOR 1I.



Figures 6.3.9 and 6.3.10: MICRO-ROTOR TURNED BY SINGLE LASER AND ASHKIN'S FOUR BEAM TRAP.

Conclusion

The experimental study of micro-cylinders was completed using three different trapping designs, the “Top-Down”, “Bottom-Up” and the “Horizontal” traps. It is thought that the cylindrical shape will be an important part in the creation of a micro-actuator. The equations for the angular acceleration, the damping factor due to the medium, and the expressions of torque were derived then implemented into a computer program, as they were not available before this thesis. Using the program the best designs of a cylinder and laser beam were determined which would produce the maximum amount of torque. The dimensions of the cylinder were determined to have a length of 70 μm, radius of 10 μm, the end-caps had radii of 100 μm and the laser beam had a waist of 1.25 μm.

Using a laser trapping system it was shown that the cylinders would align at different orientations according to their length, which is in agreement with the theoretical results obtained from the sophisticated computer program. The calculated angle at which the cylinders become stable also agrees with the theoretical results obtained from the program.

It was also shown theoretically through the program, that the cylinders could be continuously rotated using a four-beam system. The simulations provided the data for the power that would be required to start a cylinder into a specified instantaneous angular velocity also the power required to continue the cylinder rotating at the specified angular velocity. It was also found that an eight-point rotor would rotate under the influence of a single laser beam.

It is hoped that in the near future that it will be possible to create and test a real rotor in the hopes of producing a micro-actuator, which will lead to a micro-system. The activation of the system will all be possible through the power of laser light.

Bibliography

[1] ITP Nelson. *CANADIAN DICTIONARY of the ENGLISH LANGUAGE* International Thomson Publishing Company, Toronto, 1997.

[2] http://www.nobel.se/announcement-97/playbak97.html

[3] Benjamin G. Levich. *THEORETICAL PHYSICS 1*. Wiley & Sons Inc., New York, 1970 page 47.

[4] B. E. A. Saleh and M. C. Teich. *Fundamentals of Photonics*. John Wiley & Sons, New York, 1991 page 391.

[5] A. Ashkin and J. M. Dziedzic. Optical Levitation by Radiation Pressure. *Appl. Phys. Lett.*, **19** (8), 15 Oct. 1971, 283-285.

[6] A. Ashkin, J. M. Dziedzic, K. E. Bjorkholm, S. Chu. Observation of a Single-beam Gradient Force Optical Trap for Dielectric Particles. *Optics Letters*, **19** (5), May 1986, 288-290.

[7] H. Tashiro, M. Uchida, M Sato-Maeda. Three-dimensional Cell Manipulator by Means of Optical Trapping for the Specification of Cell-to-Cell Adhesion. *Optical Engineering*, **32** (11), Nov. 1993, 2812-2817.

[8] E. Higurashi, O. Ohguchi, T. Tamamura, H. Ukita, R. Sawada. Optically Induced Rotation of Dissymmetrically Shaped Fluorinated Polyimide Micro-Objects in Optical Traps. *J. Appl. Phys.* **82** (6), 15 Sept. 1997, 2773-2779.

[9] J. R. Taylor, C. D. Zafiratos. *Modern Physics (For Scientists and Engineers).* Prentice-Hall Inc. 1991. Page 118.

[10] R. C. Gauthier. Optical Trapping a Tool to Assist Optical Machining. Optics and Laser Technology. **29** (7). Oct. 1997, 389-399.

[11] J. P. Barton, D. R. Alexander, S. A. Schaub. Theoretical Determination of the Net Radiation Force and Torque for a Spherical Illumination by a Focused Laser Beam. . *J. Appl. Phys.* **66** (10), Nov. 1989, 4594-4602.

[12] G. Roosen. La Levitation Optique de Sphere. *Can. J. Phys.* **57**, 1979, 1260-1279.

[13] B. E. A. Saleh and M. C. Teich. *Fundamentals of Photonics.* John Wiley & Sons, New York, 1991, Chapter 3.

[14] Arthur B. Simon. *Algebra and Trigonometry with Analytical Geometry.* W. H, Freeman and Company, San Francisco, 1979, 390.

[15] Arthur B. Simon. *Algebra and Trigonometry with Analytical Geometry.* W. H, Freeman and Company, San Francisco, 1979, 429.

[16] D. Halliday and R. Resnick. *Fundamentals of Physics: Third Edition Extended*. John Wiley & Sons, New York, 1988, page 242.

[17] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations and Boundary Value Problems: Fifth Edition*. John Wiley & Sons, New York, 1992, page 155.

[18] D. Halliday and R. Resnick. *Fundamentals of Physics: Third Edition Extended*. John Wiley & Sons, New York, 1988, page 236.

[19] A. Ashkin. Acceleration and Trapping of Particles by Radiation Pressure. *Phys. Rev. Lett.*, **24** (4), 26 Jan 1970, 156-159.

[20] W. Wang *et. al.* Optical Trapping and Flourescence Detection in Laminar Flow Streams. Appl. Phys. Lett., **67**(8), 21 August 1995, 1057-1059.

[21] Arthur Ashkin. Apparatuses for Trapping and Accelerating Neutral Particles. United States Patent, Patent number 3,710,279. Jan. 9, 1973.

APPENDIX A

A.1 Point Ray Clarification

Chapter 2 presented the concept of the "point ray" which was said to be different from the geometrical properties of a classical ray in optics. This Appendix clarifies the use of the term "point ray" to represent the photons of the Gaussian beam.

If figure A.1.1, shown below, the waist W(z) of a Gaussian beam ($W_o=\lambda_o=1.0$ μm) is traced through the minimum region to the far field shown as -*- on the graph.

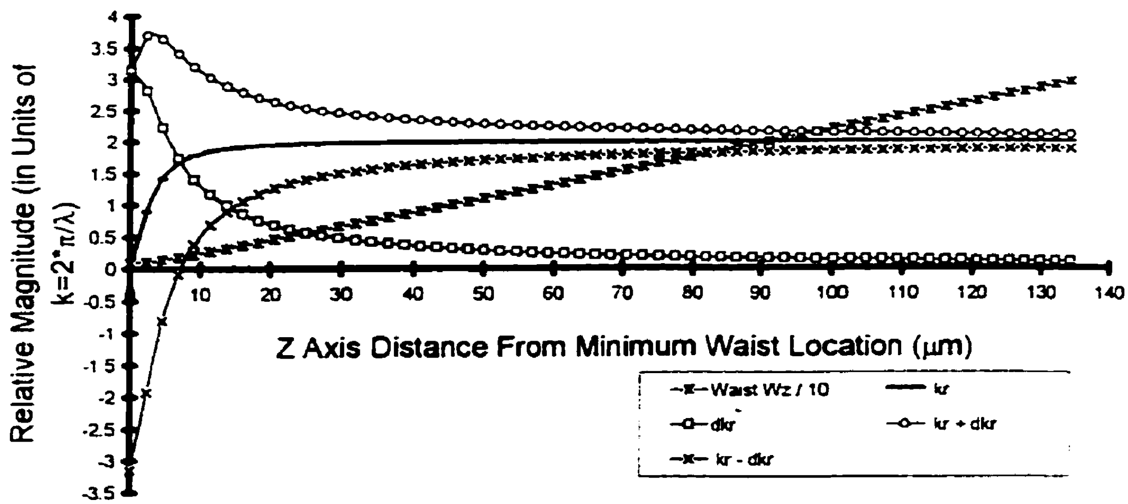


Figure A.1.1: RELATIVE MAGNITUDE VERSUS Z AXIS DISTANCE.

The photons in the minimum region are highly confined due to the small size of the waist while the photons are spread over a larger area far from the waist due to the increased beam width. The photons passing at the minimum waist are restricted radially to a spread of dr which is equal to twice the waist,. $2*W_o$. If a restriction of this type is imposed on the position coordinate then from the Heisenberg Uncertainty Principle there is an uncertainty in the momentum component that is associated with the coordinate. The minimum uncertainty in the radial component of the momentum vector for the photons in the waist region W_o is obtained from:

$$dP_r = \hbar\, dk_r = \hbar\pi / W_o \qquad \text{(A.1.1)}$$

The radial component of the photon's momentum in the minimum waist region is then represented by:

$$P_r = \hbar\, k_{ro} \pm \hbar\, dk_r \qquad \text{(A.1.2)}$$

where $\hbar k_{ro}$ is the statistical average of the radial momentum component of the photons passing through the minimum waist. By finding the radial component of the wave vector, which is perpendicular to the wavefront surface in the minimum waist region and at the radial point of interest, the statistical average is determined. The wavefront is perpendicular to the z axis in the minimum waist region which makes $k_{ro}=0$ but, there is an uncertainty in the value of dk_r that is determined by equation A.1.1. In the minimum waist region a true classical ray is not considered. What is done instead is the determination of the statistical average of the photons at a point of interest in the focus region where the averaged photon direction and momentum is then obtained. If the point of interest is far from the minimum waist the classical ray tracing techniques can be used otherwise these values are only used at the point of interest at or near the minimum waist.

Figure A.1.1 also shows the plot of k_{ro}, dk_r, and $k_{ro} \pm dk_r$ from the minimum waist region to the far field region. $W(z)$ was chosen as the radial coordinate value for the photons for the calculations displayed in the figure. It can be seen that in the minimum waist region the uncertainty dk_r is large compared to k_{ro} but as the z coordinate increases past the Rayleigh range the uncertainty decreases and k_{ro} approaches a uniform value. This also results in the uncertainty on the radial component of the photon momentum to decrease accordingly. The radial component of the momentum can then be obtained for a nearly spherical wavefront, at a point in the far field, with an uncertainty that is small. This then corresponds to a classical ray that can be propagated using the normal ray tracing techniques since the component magnitudes of the wave vector no longer change with increasing z.

For the difficulty of the conservation of momentum, figure A.1.1 shows that the radial component of the wave vector, k_{ro}, is conserved to the limits and accuracy imposed by the uncertainties. It can be seen that when the far field value is extended linearly to the minimum waist region it always lies within the limits of $\pm dk_r$. Taking $\vec{P} = \hbar\vec{k}$, beam momentum is conserved if the uncertainties in the momentum components are allowed for. It must also be stated that it is physically incorrect to tag a photon in the minimum waist region and propagate it to the far field region. What is done instead is that the average properties of the photons are treated as dictated by quantum mechanics.

The last problem to be looked at is the difficulty with bending ray trajectories. This can be overcome if it is accepted that at any point in interest not only one photon propagation direction passes by the point but photon directions are spread over a solid angle distribution by the uncertainty principle. It is the statistical average of these directions that is used to determine the direction of a ray at that point only in the beam.

R. C. Gauthier addressed the two major problems of conservation of momentum and curving of ray trajectories without interacting with matter. His modified ray theory overcame the problems that arise when the ray and wave theories are mixed in the focal region of Gaussian beams. This implies that in the focal region of a Gaussian beam the classical interpretation of ray's propagating in a straight line is not employed. What is done is the statistical average of the photons passing through a point can be interpreted as a ray passing through the point and have a direction given by the statistical average of the photon wave vectors determined at the point.

A.2 Direction Cosines

In this section the mathematical details are given for obtaining the reflected and refracted photon cosines when given the incident direction cosines, point of intercept, and the outward directed surface normal at the interface.

For the reflected direction cosines the angle of incidence between the photon direction and the surface normal is obtained by taking the dot product of the incident wave vector and the outward directed surface normal (see figure A.2.1):

$$\theta_{in} = \cos^{-1}\left(\left|al_i + bm_i + cn_i\right|\right) \tag{A.2.1}$$

Since the incident wave vector and the surface normal point in the opposite directions the absolute value is required.

The reflected angle for the photons, with respect to the surface normal, is given by thelaw of reflection as:

$$\theta_{ref} = \theta_{in} \tag{A.2.2}$$

It is known that the reflected photons and incident photons all lie in the same plane so, a point on the reflected wave vector can be obtained by finding a point (x_n, y_n, z_n) which is a distance $r\cos(\theta_{in})$ from the point of incidence, measured along the surface normal. This point is obtained using:

$$
\begin{aligned}
x_n &= x_i + ar\cos(\theta_{in}) \\
y_n &= y_i + br\cos(\theta_{in}) \\
z_n &= z_i + cr\cos(\theta_{in})
\end{aligned}
\tag{A.2.3}
$$

with

$$r = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2} \tag{A.2.4}$$

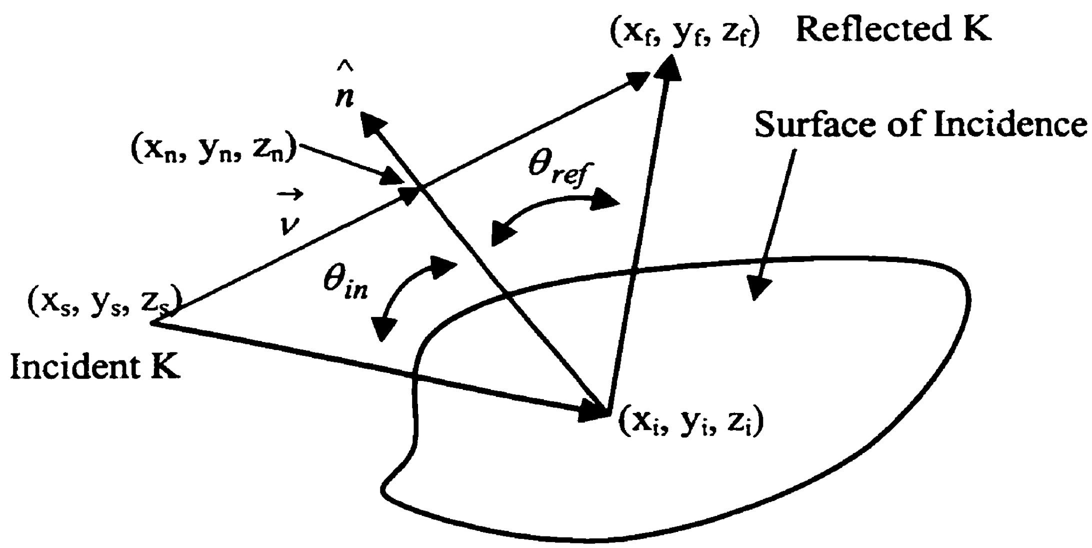


Figure A.2.1: GEOMETRY USED TO OBTAIN THE REFLECTED DIRECTION COSINES GIVEN THE INCIDENT DIRECTION COSINES AND NORMAL AT THE POINT OF INTERCEPT.

A vector $\vec{v}$, with a length of $r\sin(\theta)_{in}$, is formed which originates at (x_s, y_s, z_s) and ends at (x_n, y_n, z_n). The vector is perpendicular to the surface normal and lies in the plane of incidence. The direction cosines for this vector are expressed as:

$$a_v = (x_n - x_s) / r\sin(\theta_{in})$$
$$b_v = (y_n - y_s) / r\sin(\theta_{in}) \qquad \text{(A.2.5)}$$
$$c_v = (z_n - z_s) / r\sin(\theta_{in})$$

The point on the reflected wave vector (x_r, y_r, z_r) can be obtained by:

$$x_r = x_s + 2ra_v\sin(\theta_{in})$$
$$y_r = y_s + 2rb_v\sin(\theta_{in}) \qquad \text{(A.2.6)}$$
$$z_r = z_s + 2rc_v\sin(\theta_{in})$$

The direction cosines of the reflected wave are then given by:

$$l_r = (x_r - x_i) / r$$
$$m_r = (y_r - y_i) / r \qquad \text{(A.2.7)}$$
$$n_r = (z_r - z_i) / r$$

where r is now expressed as:

$$r = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2} \qquad \text{(A.2.8)}$$

It is the set (l_r, m_r, n_r) of equation A.2.7 that is used in equation 2.2.8.

The refracted direction cosines are obtained by finding the point (x_d, y_d, z_d) which lies on a flat plane surface with a surface normal of (a, b, c) and a surface equation of $ax+by+cz+d=0$ (see figure A.2.2). This point is located at a distance $r\cos(\theta_{in})$ from the point (x_s, y_s, z_s). This point can be calculated from:

$$x_d = x_s - at_d$$
$$y_d = y_s - bt_d \qquad \text{(A.2.9)}$$
$$z_d = z_s - ct_d$$

where t_d is obtained from:

$$t_d = ax_s + by_s + cz_s + d \qquad \text{(A.2.10)}$$

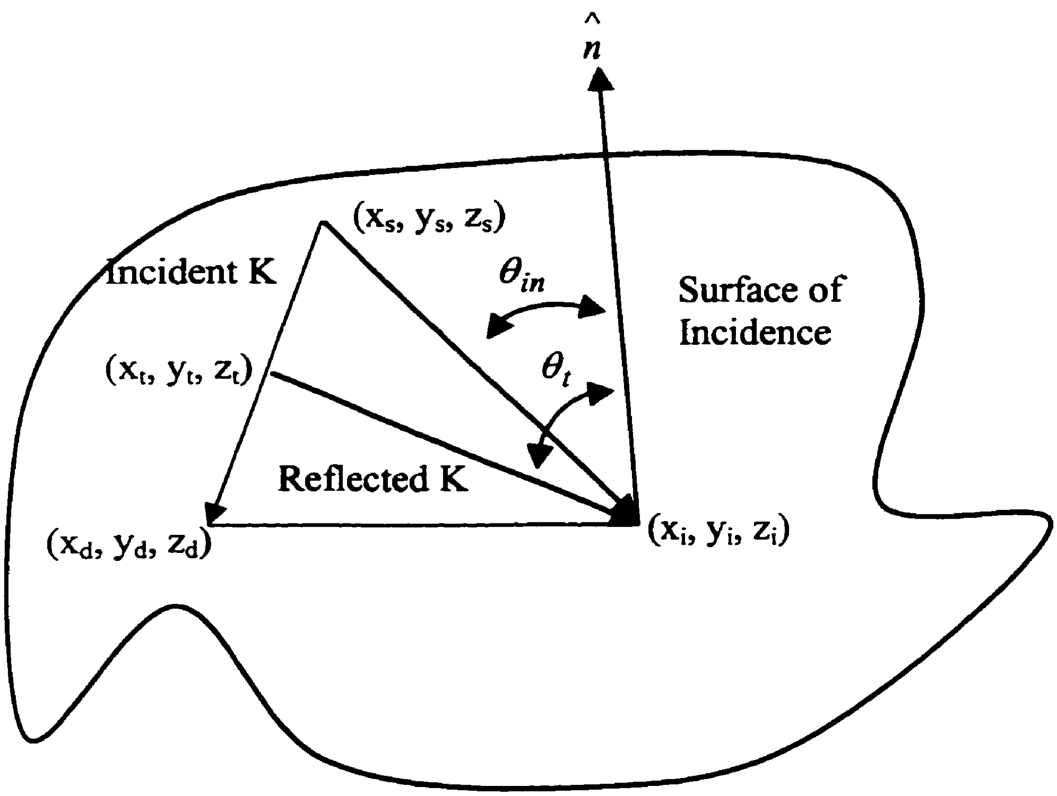


Figure A.2.2: GEOMETRY USED TO OBTAIN THE REFRACTED DIRECTION COSINES GIVEN THE INCIDENT DIRECTION COSINES AND NORMAL AT THE POINT OF INTERCEPT.

The next step is to obtain the point (x_t, y_t, z_t) which lies on the line joining (x_d, y_d, z_d) and (x_s, y_s, z_s) and is located a distance d_t from (x_d, y_d, z_d). It can be seen from the geometry of figure A.2.2 the length d_t can be obtained from:

$$d_t = r\sin(\theta_{in})\tan\left(\pi/2 - \theta_t\right) \tag{A.2.11}$$

where θ_t is calculated from the law of refraction:

$$n_{in}\sin(\theta_{in}) = n_{out}\sin(\theta_t) \tag{A.2.12}$$

The point (x_t, y_t, z_t) is given by:

$$\begin{aligned} x_t &= x_d + ad_t \\ y_t &= y_d + bd_t \\ z_t &= z_d + cd_t \end{aligned} \tag{A.2.13}$$

The direction cosines for the refracted wave are:

$$\begin{aligned} l_t(t) &= (x_t - x_i)/r_t \\ m_t(t) &= (y_t - y_i)/r_t \\ n_t(t) &= (z_t - z_i)/r_t \end{aligned} \tag{A.2.14}$$

with r_t now given as:

$$r_t = \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2 + (z_i - z_t)^2} \tag{A.2.15}$$

It is the set (l_t, m_t, n_t) of equation A.2.14 that is used in equation 2.2.9.

Appendix B

B.1 Computer Simulation of Torque rotation

```
Option Explicit
Global swidth, sheight, swidth30, sheight30 As Double
Global Pi, xmin, xmax, ymin, ymax, a, b, c As Double
Global kill As String
Global i As Integer
Sub sleeper()
 kill = "no"
 Do
  DoEvents
 Loop While kill = "no"
End Sub
Sub main()
 Pi = 4# * Atn(1#)
 Call plot
 Call plot_1a
End Sub

Sub plot()
 Form1.Picture1.Cls
 Form1.Picture1.Print "Plot"
 swidth = Form1.Picture1.ScaleWidth
 sheight = Form1.Picture1.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Form1.Label1.Caption = "x-coordinate"
 Form1.Label2.Caption = "y-coordinate"
 Form1.Label3.Caption = "z-coordinate"
 Call place_line(-100, 0, 100, 0, 0, 255, 0)
 Call place_line(0, -100, 0, 100, 0, 0, 255)
 For i = -100 To 100 Step 20
```

```
  Call place_line(i, 0, i, -ymax / 20, 0, 255, 0)
 Next i
 For i = -100 To 100 Step 20
  Call place_line(0, i, -xmax / 20, i, 0, 0, 255)
 Next i
 Call place_text(0, 110, "y")
 Call place_text(100, 0, "x")
 Call point_set
 Call coord_set
 Call coord_change
 Form1.Label6.Caption = "Point's Initial Placement"
 Form1.Label10.Caption = "Point's Position After Phi Rotation"
End Sub

Sub place_text(x1, y1, text)
 Call place_line(x1, y1, x1, y1, 255, 255, 255)
 Form1.Picture1.Print text
End Sub

Sub place_line(x1, y1, x2, y2, rgb1, rgb2, rgb3)
 Dim px1, px2, py1, py2 As Double
 px1 = ((xmax - xmin) / 20 + x1 - xmin) / (xmax - xmin + (xmax - xmin) /
10) * swidth
 px2 = ((xmax - xmin) / 20 + x2 - xmin) / (xmax - xmin + (xmax - xmin) /
10) * swidth
 py1 = (ymax + (ymax - ymin) / 10 - ((ymax - ymin) / 20 + y1)) / (ymax -
ymin + (ymax - ymin) / 10) * sheight
 py2 = (ymax + (ymax - ymin) / 10 - ((ymax - ymin) / 20 + y2)) / (ymax -
ymin + (ymax - ymin) / 10) * sheight
 Form1.Picture1.PSet (px1, py1), RGB(rgb1, rgb2, rgb3)
 Form1.Picture1.Line -(px2, py2), RGB(rgb1, rgb2, rgb3)
End Sub

Sub point_set()
 Call place_line(Val(Form1.Text1.text), Val(Form1.Text2.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text2.text) + 2, 255, 0, 0)
 Call place_line(Val(Form1.Text1.text) - 2, Val(Form1.Text2.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text2.text), 255, 0, 0)
End Sub
```

```
Sub coord_set()
 Call place_text(Val(Form1.Text1.text) + 3, Val(Form1.Text2.text) + 1, "(")
 Call place_text(Val(Form1.Text1.text) + 5, Val(Form1.Text2.text),
Fix(Val(Form1.Text1.text) * 100) / 100)
 Call place_text(Val(Form1.Text1.text) + 20, Val(Form1.Text2.text), ",")
 Call place_text(Val(Form1.Text1.text) + 22, Val(Form1.Text2.text),
Fix(Val(Form1.Text2.text) * 100) / 100)
 Call place_text(Val(Form1.Text1.text) + 40, Val(Form1.Text2.text) + 1,
")")
End Sub
Sub plot_1a()
 Form1.Picture2.Cls
 Form1.Picture2.Print "Plot"
 Form1.Label10.Caption = "Point's Position After Translation"
 Form1.Text6.text = Fix((Val(Form1.Text1.text) - Val(Form1.Text10.text))
* 10 ^ (Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 Form1.Text7.text = Fix((Val(Form1.Text2.text) - Val(Form1.Text11.text))
* 10 ^ (Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 Form1.Text8.text = Fix((Val(Form1.Text3.text) - Val(Form1.Text12.text))
* 10 ^ (Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 swidth = Form1.Picture2.ScaleWidth
 sheight = Form1.Picture2.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line2(-100, 0, 100, 0, 0, 255, 0)
 Call place_line2(0, -100, 0, 100, 0, 0, 255)
 For i = -100 To 100 Step 20
  Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
 Next i
 For i = -100 To 100 Step 20
  Call place_line2(0, i, -xmax / 20, i, 0, 0, 255)
 Next i
 Call place_text2(0, 110, "y")
 Call place_text2(100, 0, "x")
 Call point_set1a
 Call coord_set1a
```

```
 Call axis_1a
 Call place_text2(Val(Form1.Text10.text) + 3, 110, "y'")
 Call place_text2(100, Val(Form1.Text11.text), "x'")
End Sub
Sub point_set1a()
 Call place_line2(Val(Form1.Text1.text), Val(Form1.Text2.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text2.text) + 2, 255, 0, 0)
 Call place_line2(Val(Form1.Text1.text) - 2, Val(Form1.Text2.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text2.text), 255, 0, 0)
End Sub

Sub coord_set1a()
 Call place_text2(Val(Form1.Text1.text) + 3, Val(Form1.Text2.text) + 1,
"(")
 Call place_text2(Val(Form1.Texti.text) + 5, Val(Form1.Text2.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
 Call place_text2(Val(Form1.Text1.text) + 23, Val(Form1.Text2.text), ",")
 Call place_text2(Val(Form1.Text1.text) + 25, Val(Form1.Text2.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
 Call place_text2(Val(Form1.Text1.text) + 45, Val(Form1.Text2.text) + 1,
")")
End Sub

Sub axis_1a()
 swidth = Form1.Picture2.ScaleWidth
 sheight = Form1.Picture2.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line2(-100, Val(Form1.Text11.text), 100,
Val(Form1.Text11.text), 0, 100, 0)
 Call place_line2(Val(Form1.Text10.text), -100, Val(Form1.Text10.text),
100, 0, 0, 100)
 'For i = -100 To 100 Step 20
  'Call place_line2(i, 0, i, -ymax / 20, 0, 0, 255)
 'Next i
 'For i = -100 To 100 Step 20
```

```
  'Call place_line2(0, i, -xmax / 20, i, 0, 0, 255)
 'Next i
End Sub

Sub main_2()
 Call plot_1b
 Call plot_2
End Sub

Sub plot_1b()
 Form1.Picture1.Cls
 Form1.Picture1.Print "Plot"
 Form1.Label6.Caption = "Point's Position After Translation"
 Form1.Text1.text = Fix((Val(Form1.Text6.text)) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 Form1.Text2.text = Fix((Val(Form1.Text7.text)) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 Form1.Text3.text = Fix((Val(Form1.Text8.text)) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 Form1.Label1.Caption = "x'-coordinate"
 Form1.Label2.Caption = "y'-coordinate"
 Form1.Label3.Caption = "z'-coordinate"
 swidth = Form1.Picture1.ScaleWidth
 sheight = Form1.Picture1.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line(-100, 0, 100, 0, 0, 100, 0)
 Call place_line(0, -100, 0, 100, 0, 0, 100)
 For i = -100 To 100 Step 20
  Call place_line(i, 0, i, -ymax / 20, 0, 100, 0)
 Next i
 For i = -100 To 100 Step 20
  Call place_line(0, i, -xmax / 20, i, 0, 0, 100)
 Next i
 Call place_text(0, 110, "y'")
 Call place_text(100, 0, "x'")
```

```
  Call point_set1b
  Call coord_set1b
End Sub

Sub point_set1b()
  Call place_line(Val(Form1.Text1.text), Val(Form1.Text2.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text2.text) + 2, 255, 0, 0)
  Call place_line(Val(Form1.Text1.text) - 2, Val(Form1.Text2.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text2.text), 255, 0, 0)
End Sub

Sub coord_set1b()
  Call place_text(Val(Form1.Text1.text) + 3, Val(Form1.Text2.text) + 1, "(")
  Call place_text(Val(Form1.Text1.text) + 5, Val(Form1.Text2.text),
Fix(Val(Form1.Text1.text) * 100) / 100)
  Call place_text(Val(Form1.Text1.text) + 23, Val(Form1.Text2.text), ",")
  Call place_text(Val(Form1.Text1.text) + 25, Val(Form1.Text2.text),
Fix(Val(Form1.Text2.text) * 100) / 100)
  Call place_text(Val(Form1.Text1.text) + 45, Val(Form1.Text2.text) + 1,
")")
End Sub

Sub plot_2()
  Form1.Picture2.Cls
  Form1.Picture2.Print "Plot"
  Form1.Label10.Caption = "Point's Position After Phi Rotation"
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call coord_change
  Call place_line2(-100, 0, 100, 0, 0, 255, 0)
  Call place_line2(0, -100, 0, 100, 0, 0, 255)
  For i = -100 To 100 Step 20
    Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
  Next i
```

```
  For i = -100 To 100 Step 20
   Call place_line2(0, i, -xmax / 20, i, 0, 0, 255)
  Next i
  Call place_text2(0, 110, "y")
  Call place_text2(100, 0, "x")
  Call point_set2
  Call coord_set2
  Call axis_2
  Call place_text2(100 * (-Sin(Pi / 180 * Val(Form1.Text4.text))), 110 *
Cos(Pi / 180 * Val(Form1.Text4.text)), "y'")
  Call place_text2(100 * Cos(Pi / 180 * Val(Form1.Text4.text)), 110 * Sin(Pi
/ 180 * Val(Form1.Text4.text)), "x'")
End Sub

Sub place_line2(x1, y1, x2, y2, rgb1, rgb2, rgb3)
  Dim px1, px2, py1, py2 As Double
  px1 = ((xmax - xmin) / 20 + x1 - xmin) / (xmax - xmin + (xmax - xmin) /
10) * swidth
  px2 = ((xmax - xmin) / 20 + x2 - xmin) / (xmax - xmin + (xmax - xmin) /
10) * swidth
  py1 = (ymax + (ymax - ymin) / 10 - ((ymax - ymin) / 20 + y1)) / (ymax -
ymin + (ymax - ymin) / 10) * sheight
  py2 = (ymax + (ymax - ymin) / 10 - ((ymax - ymin) / 20 + y2)) / (ymax -
ymin + (ymax - ymin) / 10) * sheight
  Form1.Picture2.PSet (px1, py1), RGB(rgb1, rgb2, rgb3)
  Form1.Picture2.Line -(px2, py2), RGB(rgb1, rgb2, rgb3)
End Sub

Sub place_text2(x1, y1, text)
  Call place_line2(x1, y1, x1, y1, 255, 255, 255)
  Form1.Picture2.Print text
End Sub

Sub point_set2()
  Call place_line2(Val(Form1.Text1.text), Val(Form1.Text2.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text2.text) + 2, 255, 0, 0)
  Call place_line2(Val(Form1.Text1.text) - 2, Val(Form1.Text2.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text2.text), 255, 0, 0)
End Sub
```

```
Sub coord_set2()
  Call place_text2(Val(Form1.Text1.text) + 3, Val(Form1.Text2.text) + 1,
"(")
  Call place_text2(Val(Form1.Text1.text) + 5, Val(Form1.Text2.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
  Call place_text2(Val(Form1.Text1.text) + 23, Val(Form1.Text2.text), ",")
  Call place_text2(Val(Form1.Text1.text) + 25, Val(Form1.Text2.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
  Call place_text2(Val(Form1.Text1.text) + 45, Val(Form1.Text2.text) + 1,
")")
End Sub

Sub axis_2()
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line2(-100 * Cos(Pi / 180 * Val(Form1.Text4.text)), -100 *
Sin(Pi / 180 * Val(Form1.Text4.text)), 100 * Cos(Pi / 180 *
Val(Form1.Text4.text)), 100 * Sin(Pi / 180 * Val(Form1.Text4.text)), 0,
100, 0)
  Call place_line2(-100 * Cos((Pi / 180 * Val(Form1.Text4.text)) + Pi / 2), -
100 * Sin((Pi / 180 * Val(Form1.Text4.text)) + Pi / 2), 100 * Cos((Pi / 180 *
Val(Form1.Text4.text)) + Pi / 2), 100 * Sin((Pi / 180 *
Val(Form1.Text4.text)) + Pi / 2), 0, 0, 100)

  'For i = -100 To 100 Step 20
   'Call place_line2(i, 0, i, -ymax / 20, 0, 0, 255)
  'Next i
  'For i = -100 To 100 Step 20
   'Call place_line2(0, i, -xmax / 20, i, 0, 0, 255)
  'Next i
End Sub

Sub coord_change()
```

```
 Form1.Text6.text = Fix((Val(Form1.Text1.text) * Cos(Pi / 180 *
Val(Form1.Text4.text)) + Val(Form1.Text2.text) * Sin(Pi / 180 *
Val(Form1.Text4.text))) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 Form1.Text7.text = Fix((Val(Form1.Text1.text) * (-Sin(Pi / 180 *
Val(Form1.Text4.text))) + Val(Form1.Text2.text) * Cos(Pi / 180 *
Val(Form1.Text4.text))) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 Form1.Text8.text = Fix(Val(Form1.Text3.text) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
End Sub
Sub main_3()
 Call plot_3
 Call plot_4
End Sub
Sub plot_3()
 Form1.Picture1.Cls
 Form1.Picture1.Print "Plot"
 Form1.Picture2.Cls
 Form1.Picture2.Print "Plot"
 swidth = Form1.Picture1.ScaleWidth
 sheight = Form1.Picture1.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line(-100, 0, 100, 0, 0, 255, 0)
 Call place_line(0, -100, 0, 100, 0, 0, 255)
 For i = -100 To 100 Step 20
  Call place_line(i, 0, i, -ymax / 20, 0, 255, 0)
 Next i
 For i = -100 To 100 Step 20
  Call place_line(0, i, -xmax / 20, i, 0, 0, 255)
 Next i
 Call place_text(0, 110, "y")
 Call place_text(100, 0, "x")
 Call point_set
 Call coord_set3
```

```
 Call axis_3
 Call place_text(100 * (-Sin(Pi / 180 * Val(Form1.Text4.text))), 110 *
Cos(Pi / 180 * Val(Form1.Text4.text)), "y'")
 Call place_text(100 * Cos(Pi / 180 * Val(Form1.Text4.text)), 110 * Sin(Pi /
180 * Val(Form1.Text4.text)), "x'")
 Form1.Label1.Caption = "x'-coordinate"
 Form1.Label2.Caption = "y'-coordinate"
 Form1.Label3.Caption = "z'-coordinate"
 Form1.Label6.Caption = "Point's Position After Phi Rotation"
 a = Fix((Val(Form1.Text1.text) * Cos(Pi / 180 * Val(Form1.Text4.text)) +
Val(Form1.Text2.text) * Sin(Pi / 180 * Val(Form1.Text4.text))) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 b = Fix((Val(Form1.Text1.text) * (-Sin(Pi / 180 * Val(Form1.Text4.text)))
+ Val(Form1.Text2.text) * Cos(Pi / 180 * Val(Form1.Text4.text))) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 c = Fix(Val(Form1.Text3.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 Form1.Text1.text = a
 Form1.Text2.text = b
 Form1.Text3.text = c
End Sub

Sub axis_3()
 swidth = Form1.Picture1.ScaleWidth
 sheight = Form1.Picture1.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line(-100 * Cos(Pi / 180 * Val(Form1.Text4.text)), -100 *
Sin(Pi / 180 * Val(Form1.Text4.text)), 100 * Cos(Pi / 180 *
Val(Form1.Text4.text)), 100 * Sin(Pi / 180 * Val(Form1.Text4.text)), 0,
100, 0)
 Call place_line(-100 * Cos((Pi / 180 * Val(Form1.Text4.text)) + Pi / 2), -
100 * Sin((Pi / 180 * Val(Form1.Text4.text)) + Pi / 2), 100 * Cos((Pi / 180 *
Val(Form1.Text4.text)) + Pi / 2), 100 * Sin((Pi / 180 *
Val(Form1.Text4.text)) + Pi / 2), 0, 0, 100)
 'For i = -100 To 100 Step 20
```

```
   'Call place_line(i, 0, i, -ymax / 20, 0, 0, 255)
  'Next i
  'For i = -100 To 100 Step 20
   'Call place_line(0, i, -xmax / 20, i, 0, 0, 255)
  'Next i
End Sub

Sub coord_set3()
 Call place_text(Val(Form1.Text1.text) + 3, Val(Form1.Text2.text) + 1, "(")
 Call place_text(Val(Form1.Text1.text) + 5, Val(Form1.Text2.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
 Call place_text(Val(Form1.Text1.text) + 23, Val(Form1.Text2.text), ",")
 Call place_text(Val(Form1.Text1.text) + 25, Val(Form1.Text2.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
 Call place_text(Val(Form1.Text1.text) + 45, Val(Form1.Text2.text) + 1,
")")
End Sub

Sub plot_4()
 Form1.Picture2.Cls
 Form1.Picture2.Print "Plot"
 swidth = Form1.Picture2.ScaleWidth
 sheight = Form1.Picture2.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line2(-100, 0, 100, 0, 0, 255, 0)
 Call place_line2(0, -100, 0, 100, 255, 0, 0)
 For i = -100 To 100 Step 20
  Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
 Next i
 For i = -100 To 100 Step 20
  Call place_line2(0, i, -xmax / 20, i, 255, 0, 0)
 Next i
 Call place_text2(3, 110, "z''")
 Call place_text2(100, 0, "x''")
 Call point_set3
```

```
  Call coord_set4
  Form1.Label7.Caption = "x"-coordinate"
  Form1.Label8.Caption = "y"-coordinate"
  Form1.Label9.Caption = "z"-coordinate"
  Form1.Label10.Caption = "Point's Position Before Theta Rotation"
End Sub

Sub axis_4()
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line2(-100 * Cos(Pi / 180 * Val(Form1.Text5.text)), -100 *
Sin(Pi / 180 * Val(Form1.Text5.text)), 100 * Cos(Pi / 180 *
Val(Form1.Text5.text)), 100 * Sin(Pi / 180 * Val(Form1.Text5.text)), 0, 0,
255)
  Call place_line2(-100 * Cos((Pi / 180 * Val(Form1.Text5.text)) + Pi / 2), -
100 * Sin((Pi / 180 * Val(Form1.Text5.text)) + Pi / 2), 100 * Cos((Pi / 180 *
Val(Form1.Text5.text)) + Pi / 2), 100 * Sin((Pi / 180 *
Val(Form1.Text5.text)) + Pi / 2), 0, 0, 255)

  For i = -100 To 100 Step 20
    Call place_line2(i, 0, i, -ymax / 20, 0, 0, 255)
  Next i
  For i = -100 To 100 Step 20
    Call place_line2(0, i, -xmax / 20, i, 0, 0, 255)
  Next i
End Sub

Sub coord_set4()
  Call place_text2(Val(Form1.Text1.text) + 3, Val(Form1.Text3.text) + 1,
"(")
  Call place_text2(Val(Form1.Text1.text) + 5, Val(Form1.Text3.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
  Call place_text2(Val(Form1.Text1.text) + 23, Val(Form1.Text3.text), ",")
```

```
  Call place_text2(Val(Form1.Text1.text) + 25, Val(Form1.Text3.text),
Fix(Val(Form1.Text8.text) * 100) / 100)
  Call place_text2(Val(Form1.Text1.text) + 45, Val(Form1.Text3.text) + 1,
")")
End Sub

Sub point_set3()
  Call place_line2(Val(Form1.Text1.text), Val(Form1.Text3.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text3.text) + 2, 0, 0, 255)
  Call place_line2(Val(Form1.Text1.text) - 2, Val(Form1.Text3.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text3.text), 0, 0, 255)
End Sub

Sub main_4()
  Call plot_5
  Call plot_6
End Sub

Sub plot_5()
  Form1.Picture1.Cls
  Form1.Picture1.Print "Plot"
  swidth = Form1.Picture1.ScaleWidth
  sheight = Form1.Picture1.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line(-100, 0, 100, 0, 0, 255, 0)
  Call place_line(0, -100, 0, 100, 255, 0, 0)
  For i = -100 To 100 Step 20
    Call place_line(i, 0, i, -ymax / 20, 0, 255, 0)
  Next i
  For i = -100 To 100 Step 20
    Call place_line(0, i, -xmax / 20, i, 255, 0, 0)
  Next i
  Call place_text(3, 110, "z""")
  Call place_text(100, 0, "x""")
  Call point_set4
```

```
 Call coord_set5
 a = Fix(Val(Form1.Text6.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 b = Fix(Val(Form1.Text7.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 c = Fix(Val(Form1.Text8.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 Form1.Text1.text = a
 Form1.Text2.text = b
 Form1.Text3.text = c
 Form1.Label1.Caption = "x"-coordinate"
 Form1.Label2.Caption = "y"-coordinate"
 Form1.Label3.Caption = "z"-coordinate"
 Form1.Label6.Caption = "Point's Position Before Theta Rotation"
End Sub

Sub point_set4()
 Call place_line(Val(Form1.Text1.text), Val(Form1.Text3.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text3.text) + 2, 0, 0, 255)
 Call place_line(Val(Form1.Text1.text) - 2, Val(Form1.Text3.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text3.text), 0, 0, 255)
End Sub

Sub coord_set5()
 Call place_text(Val(Form1.Text1.text) + 3, Val(Form1.Text3.text) + 1, "(")
 Call place_text(Val(Form1.Text1.text) + 5, Val(Form1.Text3.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
 Call place_text(Val(Form1.Text1.text) + 23, Val(Form1.Text3.text), ",")
 Call place_text(Val(Form1.Text1.text) + 25, Val(Form1.Text3.text),
Fix(Val(Form1.Text8.text) * 100) / 100)
 Call place_text(Val(Form1.Text1.text) + 45, Val(Form1.Text3.text) + 1,
")")
End Sub
Sub plot_6()
 Form1.Picture2.Cls
 Form1.Picture2.Print "Plot"
 a = Fix((Val(Form1.Text6.text) * Cos(Pi / 180 * Val(Form1.Text5.text)) +
Val(Form1.Text8.text) * (-Sin(Pi / 180 * Val(Form1.Text5.text)))) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
```

```
 b = Fix(Val(Form1.Text7.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 c = Fix((Val(Form1.Text6.text) * Sin(Pi / 180 * Val(Form1.Text5.text)) +
Val(Form1.Text8.text) * Cos(Pi / 180 * Val(Form1.Text5.text))) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
 Form1.Text6.text = a
 Form1.Text7.text = b
 Form1.Text8.text = c
 swidth = Form1.Picture2.ScaleWidth
 sheight = Form1.Picture2.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line2(-100, 0, 100, 0, 0, 255, 0)
 Call place_line2(0, -100, 0, 100, 255, 0, 0)
 For i = -100 To 100 Step 20
  Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
 Next i
 For i = -100 To 100 Step 20
  Call place_line2(0, i, -xmax / 20, i, 0, 255, 0)
 Next i
 Call place_text2(3, 100, "z''")
 Call place_text2(100, 0, "x''")
 Call point_set5
 Call coord_set6
 Call axis_5
 Call place_text2(100 * (Sin(Pi / 180 * Val(Form1.Text5.text))) + 3, 100 *
Cos(Pi / 180 * Val(Form1.Text5.text)), "z'''")
 Call place_text2(100 * Cos(Pi / 180 * Val(Form1.Text5.text)), 100 * -
Sin(Pi / 180 * Val(Form1.Text5.text)), "x'''")
 Form1.Label7.Caption = "x'''-coordinate"
 Form1.Label8.Caption = "y'''-coordinate"
 Form1.Label9.Caption = "z'''-coordinate"
 Form1.Label10.Caption = "Point's Position After Theta Rotation"

End Sub
```

```
Sub point_set5()
 Call place_line2(Val(Form1.Text1.text), Val(Form1.Text3.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text3.text) + 2, 0, 0, 255)
 Call place_line2(Val(Form1.Text1.text) - 2, Val(Form1.Text3.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text3.text), 0, 0, 255)
End Sub

Sub coord_set6()
 Call place_text2(Val(Form1.Text1.text) + 3, Val(Form1.Text3.text) + 1,
"(")
 Call place_text2(Val(Form1.Text1.text) + 5, Val(Form1.Text3.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
 Call place_text2(Val(Form1.Text1.text) + 23, Val(Form1.Text3.text), ",")
 Call place_text2(Val(Form1.Text1.text) + 25, Val(Form1.Text3.text),
Fix(Val(Form1.Text8.text) * 100) / 100)
 Call place_text2(Val(Form1.Text1.text) + 45, Val(Form1.Text3.text) + 1,
")")
End Sub

Sub axis_5()
 swidth = Form1.Picture2.ScaleWidth
 sheight = Form1.Picture2.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line2(-100 * Cos(Pi / 180 * Val(Form1.Text5.text)), -100 * -
Sin(Pi / 180 * Val(Form1.Text5.text)), 100 * Cos(Pi / 180 *
Val(Form1.Text5.text)), 100 * -Sin(Pi / 180 * Val(Form1.Text5.text)), 0,
255, 200)
 Call place_line2(-100 * Sin((Pi / 180 * Val(Form1.Text5.text))), -100 *
Cos((Pi / 180 * Val(Form1.Text5.text))), 100 * Sin((Pi / 180 *
Val(Form1.Text5.text))), 100 * Cos((Pi / 180 * Val(Form1.Text5.text))),
255, 0, 200)
 For i = -100 To 100 Step 20
  Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
 Next i
 For i = -100 To 100 Step 20
```

```
    Call place_line2(0, i, -xmax / 20, i, 255, 0, 0)
  Next i
End Sub

Sub main_5()
  Call plot_7
  Call plot_8
End Sub

Sub plot_7()
  Form1.Picture1.Cls
  Form1.Picture1.Print "Plot"
  swidth = Form1.Picture1.ScaleWidth
  sheight = Form1.Picture1.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line(-100, 0, 100, 0, 0, 255, 200)
  Call place_line(0, -100, 0, 100, 255, 0, 200)
  For i = -100 To 100 Step 20
    Call place_line(i, 0, i, -ymax / 20, 0, 255, 200)
  Next i
  For i = -100 To 100 Step 20
    Call place_line(0, i, -xmax / 20, i, 255, 0, 200)
  Next i
  Call place_text(3, 100, "z""")
  Call place_text(100, 0, "x""")
  Call point_set6
  Call coord_set7
  a = Fix(Val(Form1.Text6.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  b = Fix(Val(Form1.Text7.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  c = Fix(Val(Form1.Text8.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  Form1.Text1.text = a
  Form1.Text2.text = b
```

```
  Form1.Text3.text = c
  Form1.Label1.Caption = "x'''-coordinate"
  Form1.Label2.Caption = "y'''-coordinate"
  Form1.Label3.Caption = "z'''-coordinate"
  Form1.Label6.Caption = "Point's Position After Theta Rotation"
End Sub

Sub point_set6()
  Call place_line(Val(Form1.Text6.text), Val(Form1.Text8.text) - 2,
Val(Form1.Text6.text), Val(Form1.Text8.text) + 2, 0, 0, 255)
  Call place_line(Val(Form1.Text6.text) - 2, Val(Form1.Text8.text),
Val(Form1.Text6.text) + 2, Val(Form1.Text8.text), 0, 0, 255)
End Sub

Sub coord_set7()
  Call place_text(Val(Form1.Text1.text) + 3, Val(Form1.Text3.text) + 1, "(")
  Call place_text(Val(Form1.Text1.text) + 5, Val(Form1.Text3.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
  Call place_text(Val(Form1.Text1.text) + 23, Val(Form1.Text3.text), ",")
  Call place_text(Val(Form1.Text1.text) + 25, Val(Form1.Text3.text),
Fix(Val(Form1.Text8.text) * 100) / 100)
  Call place_text(Val(Form1.Text1.text) + 45, Val(Form1.Text3.text) + 1,
")")
End Sub

Sub plot_8()
  Form1.Picture2.Cls
  Form1.Picture2.Print "Plot"
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  a = Fix((Val(Form1.Text1.text) * Cos(Pi / 180 * Val(Form1.Text9.text)) +
(Val(Form1.Text2.text) * (-Sin(Pi / 180 * Val(Form1.Text9.text))))) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  b = Fix((Val(Form1.Text1.text) * Sin(Pi / 180 * Val(Form1.Text9.text)) +
(Val(Form1.Text2.text) * Cos(Pi / 180 * Val(Form1.Text9.text)))) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  c = Fix(Val(Form1.Text3.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  Form1.Text6.text = a
  Form1.Text7.text = b
```

```
 Form1.Text8.text = c
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Call place_line2(-100, 0, 100, 0, 0, 100, 0)
 Call place_line2(0, -100, 0, 100, 0, 0, 100)
 For i = -100 To 100 Step 20
  Call place_line2(i, 0, i, -ymax / 20, 0, 100, 0)
 Next i
 For i = -100 To 100 Step 20
  Call place_line2(0, i, -xmax / 20, i, 0, 0, 100)
 Next i
 Call place_text2(3, 100, "y''")
 Call place_text2(100, 0, "x''")
 Call point_set7
 Call coord_set9
  Form1.Label10.Caption = "Point's Position After Torque"

End Sub

Sub point_set7()
 Call place_line2(Val(Form1.Text6.text), Val(Form1.Text7.text) - 2,
Val(Form1.Text6.text), Val(Form1.Text7.text) + 2, 255, 0, 0)
 Call place_line2(Val(Form1.Text6.text) - 2, Val(Form1.Text7.text),
Val(Form1.Text6.text) + 2, Val(Form1.Text7.text), 255, 0, 0)
End Sub

Sub coord_set9()
 Call place_text2(Val(Form1.Text6.text) + 3, Val(Form1.Text7.text) + 1,
"(")
 Call place_text2(Val(Form1.Text6.text) + 5, Val(Form1.Text7.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
 Call place_text2(Val(Form1.Text6.text) + 23, Val(Form1.Text7.text), ",")
 Call place_text2(Val(Form1.Text6.text) + 25, Val(Form1.Text7.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
 Call place_text2(Val(Form1.Text6.text) + 45, Val(Form1.Text7.text) + 1,
")")
```

```
End Sub

Sub main_6()
  Call plot_9
  Call plot_10
End Sub

Sub plot_9()
  Form1.Picture1.Cls
  Form1.Picture1.Print "Plot"
  swidth = Form1.Picture1.ScaleWidth
  sheight = Form1.Picture1.ScaleHeight
  a = Fix(Val(Form1.Text6.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  b = Fix(Val(Form1.Text7.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  c = Fix(Val(Form1.Text8.text) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  Form1.Text1.text = a
  Form1.Text2.text = b
  Form1.Text3.text = c
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line(-100, 0, 100, 0, 0, 255, 0)
  Call place_line(0, -100, 0, 100, 0, 0, 255)
  For i = -100 To 100 Step 20
    Call place_line(i, 0, i, -ymax / 20, 0, 255, 0)
  Next i
  For i = -100 To 100 Step 20
    Call place_line(0, i, -xmax / 20, i, 0, 0, 255)
  Next i
  Call place_text(3, 100, "y""")
  Call place_text(100, 0, "x""")
  Call point_set9
  Call coord_set10
  Form1.Label6.Caption = "Point's Position After Torque"
```

```
End Sub

Sub point_set9()
  Call place_line(Val(Form1.Text1.text), Val(Form1.Text2.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text2.text) + 2, 255, 0, 0)
  Call place_line(Val(Form1.Text1.text) - 2, Val(Form1.Text2.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text2.text), 255, 0, 0)
End Sub

Sub coord_set10()
  Call place_text(Val(Form1.Text6.text) + 3, Val(Form1.Text7.text) + 1, "(")
  Call place_text(Val(Form1.Text6.text) + 5, Val(Form1.Text7.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
  Call place_text(Val(Form1.Text6.text) + 23, Val(Form1.Text7.text), ",")
  Call place_text(Val(Form1.Text6.text) + 25, Val(Form1.Text7.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
  Call place_text(Val(Form1.Text6.text) + 45, Val(Form1.Text7.text) + 1,
")")
End Sub

Sub plot_10()
  Form1.Picture2.Cls
  Form1.Picture2.Print "Plot"
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Form1.Text6.text = Fix((Val(Form1.Text1.text) * Cos(Pi / 180 *
Val(Form1.Text5.text)) + Val(Form1.Text8.text) * Sin(Pi / 180 *
Val(Form1.Text5.text))) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  Form1.Text7.text = Fix(Val(Form1.Text2.text) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Text8.text = Fix((Val(Form1.Text1.text) * (-Sin(Pi / 180 *
Val(Form1.Text5.text))) + Val(Form1.Text8.text) * Cos(Pi / 180 *
```

```
Val(Form1.Text5.text))) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
 Call place_line2(-100, 0, 100, 0, 0, 255, 200)
 Call place_line2(0, -100, 0, 100, 255, 0, 200)
 For i = -100 To 100 Step 20
  Call place_line2(i, 0, i, -ymax / 20, 0, 255, 200)
 Next i
 For i = -100 To 100 Step 20
  Call place_line2(0, i, -xmax / 20, i, 255, 0, 200)
 Next i
 Call place_text2(3, 105, "z""")
 Call place_text2(100, 0, "x""")
 Call point_set10
 Call coord_set11
 Call axis_6
 Form1.Label10.Caption = "Point's Position After First Return Rotation"
 Form1.Label7.Caption = "x"-coordinate"
 Form1.Label8.Caption = "y"-coordinate"
 Form1.Label9.Caption = "z"-coordinate"
End Sub

Sub point_set10()
 Call place_line2(Val(Form1.Text1.text), Val(Form1.Text3.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text3.text) + 2, 0, 0, 255)
 Call place_line2(Val(Form1.Text1.text) - 2, Val(Form1.Text3.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text3.text), 0, 0, 255)
End Sub

Sub coord_set11()
 Call place_text2(Val(Form1.Text1.text) + 3, Val(Form1.Text3.text) + 1,
"(")
 Call place_text2(Val(Form1.Text1.text) + 5, Val(Form1.Text3.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
 Call place_text2(Val(Form1.Text1.text) + 23, Val(Form1.Text3.text), ",")
 Call place_text2(Val(Form1.Text1.text) + 25, Val(Form1.Text3.text),
Fix(Val(Form1.Text8.text) * 100) / 100)
 Call place_text2(Val(Form1.Text1.text) + 45, Val(Form1.Text3.text) + 1,
")")
End Sub
```

```
Sub axis_6()
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line2(-100 * Cos(Pi / 180 * Val(Form1.Text5.text)), -100 *
Sin(Pi / 180 * Val(Form1.Text5.text)), 100 * Cos(Pi / 180 *
Val(Form1.Text5.text)), 100 * Sin(Pi / 180 * Val(Form1.Text5.text)), 0,
255, 0)
  Call place_line2(-100 * -Sin((Pi / 180 * Val(Form1.Text5.text))), -100 *
Cos((Pi / 180 * Val(Form1.Text5.text))), 100 * -Sin((Pi / 180 *
Val(Form1.Text5.text))), 100 * Cos((Pi / 180 * Val(Form1.Text5.text))),
255, 0, 0)
  Call place_text2(100 * (-Sin(Pi / 180 * Val(Form1.Text5.text))) + 3, 100 *
Cos(Pi / 180 * Val(Form1.Text5.text)) + 5, "z"")
  Call place_text2(100 * Cos(Pi / 180 * Val(Form1.Text5.text)), 100 * Sin(Pi
/ 180 * Val(Form1.Text5.text)), "x"")
  'For i = -100 To 100 Step 20
   'Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
  'Next i
  'For i = -100 To 100 Step 20
   'Call place_line2(0, i, -xmax / 20, i, 255, 0, 0)
  'Next i
End Sub
Sub main_7()
  Call plot_11
  Call plot_12
End Sub

Sub plot_11()
  Form1.Picture1.Cls
  Form1.Picture1.Print "Plot"
  swidth = Form1.Picture1.ScaleWidth
  sheight = Form1.Picture1.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
```

```
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Form1.Text1.text = Fix(Val(Form1.Text6.text) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Text2.text = Fix(Val(Form1.Text7.text) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Text3.text = Fix(Val(Form1.Text8.text) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Label1.Caption = "x"-coordinate"
  Form1.Label2.Caption = "y"-coordinate"
  Form1.Label3.Caption = "z"-coordinate"
  Call place_line(-100, 0, 100, 0, 0, 255, 0)
  Call place_line(0, -100, 0, 100, 255, 0, 0)
  For i = -100 To 100 Step 20
    Call place_line(i, 0, i, -ymax / 20, 0, 255, 0)
  Next i
  For i = -100 To 100 Step 20
    Call place_line(0, i, -xmax / 20, i, 255, 0, 0)
  Next i
  Call place_text(3, 105, "z"")
  Call place_text(100, 0, "x"")
  Call point_set11
  Call coord_set12
  Form1.Label6.Caption = "Point's Position After First Return Rotation"
End Sub

Sub point_set11()
  Call place_line(Val(Form1.Text1.text), Val(Form1.Text3.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text3.text) + 2, 0, 0, 255)
  Call place_line(Val(Form1.Text1.text) - 2, Val(Form1.Text3.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text3.text), 0, 0, 255)
End Sub

Sub coord_set12()
  Call place_text(Val(Form1.Text6.text) + 3, Val(Form1.Text8.text) + 1, "(")
  Call place_text(Val(Form1.Text6.text) + 5, Val(Form1.Text8.text),
Fix(Val(Form1.Text1.text) * 100) / 100)
  Call place_text(Val(Form1.Text6.text) + 23, Val(Form1.Text8.text), ",")
```

```
  Call place_text(Val(Form1.Text6.text) + 25, Val(Form1.Text8.text),
Fix(Val(Form1.Text3.text) * 100) / 100)
  Call place_text(Val(Form1.Text6.text) + 45, Val(Form1.Text8.text) + 1,
")")
End Sub

Sub plot_12()
  Form1.Picture2.Cls
  Form1.Picture2.Print "Plot"
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Form1.Text6.text = Fix((Val(Form1.Text1.text) * Cos(Pi / 180 *
Val(Form1.Text4.text)) + Val(Form1.Text2.text) * -Sin(Pi / 180 *
Val(Form1.Text4.text))) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  Form1.Text7.text = Fix((Val(Form1.Text1.text) * Sin(Pi / 180 *
Val(Form1.Text4.text)) + Val(Form1.Text2.text) * Cos(Pi / 180 *
Val(Form1.Text4.text))) * 10 ^ (Val(Form1.Text13.text))) / 10 ^
(Val(Form1.Text13.text))
  Form1.Text8.text = Fix(Val(Form1.Text3.text) * 10 ^
(Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Label7.Caption = "x'-coordinate"
  Form1.Label8.Caption = "y'-coordinate"
  Form1.Label9.Caption = "z'-coordinate"
  Call place_line2(-100, 0, 100, 0, 0, 100, 0)
  Call place_line2(0, -100, 0, 100, 0, 0, 100)
  For i = -100 To 100 Step 20
    Call place_line2(i, 0, i, -ymax / 20, 0, 100, 0)
  Next i
  For i = -100 To 100 Step 20
    Call place_line2(0, i, -xmax / 20, i, 0, 0, 100)
  Next i
  Call place_text2(3, 105, "y''")
  Call place_text2(100, 0, "x''")
```

```
  Call point_set12
  Call coord_set13
  Call axis_7
  Form1.Label10.Caption = "Point's Position After Second Return Rotation"
End Sub

Sub point_set12()
  Call place_line2(Val(Form1.Text1.text), Val(Form1.Text2.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text2.text) + 2, 255, 0, 0)
  Call place_line2(Val(Form1.Text1.text) - 2, Val(Form1.Text2.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text2.text), 255, 0, 0)
End Sub

Sub coord_set13()
  Call place_text2(Val(Form1.Text1.text) + 3, Val(Form1.Text2.text) + 1,
"(")
  Call place_text2(Val(Form1.Text1.text) + 5, Val(Form1.Text2.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
  Call place_text2(Val(Form1.Text1.text) + 23, Val(Form1.Text2.text), ",")
  Call place_text2(Val(Form1.Text1.text) + 25, Val(Form1.Text2.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
  Call place_text2(Val(Form1.Text1.text) + 45, Val(Form1.Text2.text) + 1,
")")
End Sub

Sub axis_7()
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line2(-100 * Cos(Pi / 180 * Val(Form1.Text4.text)), -100 * -
Sin(Pi / 180 * Val(Form1.Text4.text)), 100 * Cos(Pi / 180 *
Val(Form1.Text4.text)), 100 * -Sin(Pi / 180 * Val(Form1.Text4.text)), 0,
255, 0)
  Call place_line2(-100 * Sin((Pi / 180 * Val(Form1.Text4.text))), -100 *
Cos((Pi / 180 * Val(Form1.Text4.text))), 100 * Sin((Pi / 180 *
```

```
Val(Form1.Text4.text))), 100 * Cos((Pi / 180 * Val(Form1.Text4.text))), 0,
0, 255)
 Call place_text2(100 * Sin(Pi / 180 * Val(Form1.Text4.text)) + 3, 100 *
Cos(Pi / 180 * Val(Form1.Text4.text)) + 5, "y'")
 Call place_text2(100 * Cos(Pi / 180 * Val(Form1.Text4.text)), 100 * -
Sin(Pi / 180 * Val(Form1.Text4.text)), "x'")
 'For i = -100 To 100 Step 20
  'Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
 'Next i
 'For i = -100 To 100 Step 20
  'Call place_line2(0, i, -xmax / 20, i, 255, 0, 0)
 'Next i
End Sub

Sub main_8()
 Call plot_13
 Call plot_14
End Sub

Sub plot_13()
 Form1.Picture1.Cls
 Form1.Picture1.Print "Plot"
 swidth = Form1.Picture1.ScaleWidth
 sheight = Form1.Picture1.ScaleHeight
 swidth30 = swidth / 30
 sheight30 = sheight / 30
 xmin = -100
 xmax = 100
 ymin = -100
 ymax = 100
 Form1.Text1.text = Form1.Text6.text
 Form1.Text2.text = Form1.Text7.text
 Form1.Text3.text = Form1.Text8.text
 Form1.Label1.Caption = "x'-coordinate"
 Form1.Label2.Caption = "y'-coordinate"
 Form1.Label3.Caption = "z'-coordinate"
 Call place_line(-100, 0, 100, 0, 0, 255, 0)
 Call place_line(0, -100, 0, 100, 0, 0, 255)
 For i = -100 To 100 Step 20
  Call place_line(i, 0, i, -ymax / 20, 0, 255, 0)
```

```
  Next i
  For i = -100 To 100 Step 20
   Call place_line(0, i, -xmax / 20, i, 0, 0, 255)
  Next i
  Call place_text(3, 105, "y'")
  Call place_text(100, 0, "x'")
  Call point_set13
  Call coord_set14
  Form1.Label6.Caption = "Point's Position After Second Return Rotation"
End Sub

Sub point_set13()
  Call place_line(Val(Form1.Text1.text), Val(Form1.Text3.text) - 2,
Val(Form1.Text1.text), Val(Form1.Text3.text) + 2, 255, 0, 0)
  Call place_line(Val(Form1.Text1.text) - 2, Val(Form1.Text3.text),
Val(Form1.Text1.text) + 2, Val(Form1.Text3.text), 255, 0, 0)
End Sub

Sub coord_set14()
  Call place_text(Val(Form1.Text1.text) + 3, Val(Form1.Text3.text) + 1, "(")
  Call place_text(Val(Form1.Text1.text) + 5, Val(Form1.Text3.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
  Call place_text(Val(Form1.Text1.text) + 23, Val(Form1.Text3.text), ",")
  Call place_text(Val(Form1.Text1.text) + 25, Val(Form1.Text3.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
  Call place_text(Val(Form1.Text1.text) + 45, Val(Form1.Text3.text) + 1,
")")
End Sub
Sub plot_14()
  Form1.Picture2.Cls
  Form1.Picture2.Print "Plot"
  Form1.Label10.Caption = "Point's Final Position"
  Form1.Text6.text = Fix((Val(Form1.Text1.text) + Val(Form1.Text10.text))
* 10 ^ (Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Text7.text = Fix((Val(Form1.Text2.text) + Val(Form1.Text11.text))
* 10 ^ (Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Text8.text = Fix((Val(Form1.Text3.text) + Val(Form1.Text12.text))
* 10 ^ (Val(Form1.Text13.text))) / 10 ^ (Val(Form1.Text13.text))
  Form1.Label7.Caption = "x-coordinate"
  Form1.Label8.Caption = "y-coordinate"
```

```
  Form1.Label9.Caption = "z-coordinate"
  swidth = Form1.Picture2.ScaleWidth
  sheight = Form1.Picture2.ScaleHeight
  swidth30 = swidth / 30
  sheight30 = sheight / 30
  xmin = -100
  xmax = 100
  ymin = -100
  ymax = 100
  Call place_line2(-100, 0, 100, 0, 0, 255, 0)
  Call place_line2(0, -100, 0, 100, 0, 0, 255)
  For i = -100 To 100 Step 20
   Call place_line2(i, 0, i, -ymax / 20, 0, 255, 0)
  Next i
  For i = -100 To 100 Step 20
   Call place_line2(0, i, -xmax / 20, i, 0, 0, 255)
  Next i
  Call place_text2(0, 110, "y")
  Call place_text2(100, 0, "x")
  Call point_set14
  Call coord_set15
  End Sub
Sub point_set14()
  Call place_line2(Val(Form1.Text6.text), Val(Form1.Text7.text) - 2,
Val(Form1.Text6.text), Val(Form1.Text7.text) + 2, 255, 0, 0)
  Call place_line2(Val(Form1.Text6.text) - 2, Val(Form1.Text7.text),
Val(Form1.Text6.text) + 2, Val(Form1.Text7.text), 255, 0, 0)
End Sub
Sub coord_set15()
  Call place_text2(Val(Form1.Text6.text) + 3, Val(Form1.Text7.text) + 1,
"(")
  Call place_text2(Val(Form1.Text6.text) + 5, Val(Form1.Text7.text),
Fix(Val(Form1.Text6.text) * 100) / 100)
  Call place_text2(Val(Form1.Text6.text) + 23, Val(Form1.Text7.text), ",")
  Call place_text2(Val(Form1.Text6.text) + 25, Val(Form1.Text7.text),
Fix(Val(Form1.Text7.text) * 100) / 100)
  Call place_text2(Val(Form1.Text6.text) + 45, Val(Form1.Text7.text) + 1,
")")
End Sub
```

B.2 Solution of Angular Velocity for $k \neq 0$

> deq4:=In*(D@@2)(theta)(t)+b*D(theta)(t)+k*theta(t)=tau;

$$deq4 := In D^{(2)}(\theta)(t) + bD(\theta)(t) + k\theta(t) = \tau$$

> deqinit:=theta(0)=0, D(theta)(0)=0;

$$deq4init := \theta(0) = 0, D(\theta)(0) = 0$$

> deq4sol:=dsolve({deq4,deq4init},theta(t));

$$deq4sol := \theta(t) = \frac{\tau}{k} - \frac{\tau}{2\%1k\sqrt{\exp\left(bt/In\right)}}\left(\cos\left(\frac{\sqrt{\%1}\,t}{2In}\right)\left[-2b^2 + 8kIn\right] + I\sin\left(\frac{\sqrt{\%1}\,t}{2In}\right)\left(Ib\sqrt{\%1}\right)\right)$$

> diff(",t);

$$\frac{\partial}{\partial t}\theta(t) = -\frac{\tau}{\%1k}\left(\frac{b}{2In}\left[b^2 e^{\frac{bt}{2In}} + \sin\left(\frac{\sqrt{\%1}t}{2In}\right)\sqrt{\%1}b + \cos\left(\frac{\sqrt{\%1}t}{2In}\right)\sqrt{\%1}\right] - 2k\left[b\,e^{\frac{bt}{2In}} + \sin\left(\frac{\sqrt{\%1}t}{2In}\right)\sqrt{\%1}\right]\right)$$

$$+ \frac{\tau b e^{-\frac{bt}{2In}}}{2\,In\%1k}\left(e^{\frac{bt}{2In}}\left[b^2 - 4kIn\right] + \cos\left[\frac{\sqrt{\%1}t}{2In}\right]\left\{kIn - b^2\right\} + \sin\left[\frac{\sqrt{\%1}t}{2In}\right]b\sqrt{\%1}\right)$$

$$\%1 = -b^2 + 4kIn$$

> Simplify(");

$$\frac{\partial}{\partial t}\theta(t) = 2\frac{\tau e^{-\frac{bt}{2In}}\sin\left(\frac{\sqrt{-b^2 + 4kInt}}{2In}\right)}{\sqrt{-b^2 + 4kIn}}$$

B.3 Code for the Calculation of Moment of Inertia

```
Sub Inert_a(inert)
 ' the moment of inertia for the cylinder is calculated
 Dim intop As Double
 Dim InCent As Double
 Dim Inbot As Double
 Dim Cst As Double
 Dim temp As Double
 Dim R_Max As Double
 Dim R_Min As Double
 Dim dL As Double
 Dim Theta_max As Double
 Dim U As Double
 U = 1 'zzz * 10 ^ (-6)
 temp = Radius_Particle(1) / Top_Radius_Cylinder(1)
 If temp >= 1# Then
  Theta_max = Pi / 2#
 Else
  Theta_max = Abs(Atn(temp / Sqr(1# - temp ^ 2)))
 End If
 Cst = Cos(Theta_max)
 R_Max = Top_Radius_Cylinder(1)
 R_Min = R_Max * Cst
 dL = Sqr((Xo_Particle(1) - Cx_Top(1)) ^ 2 + (Yo_Particle(1) - Cy_Top(1))
^ 2 + (Zo_Particle(1) - Cz_Top(1)) ^ 2)
 If R_Min > Sqr((Radius_Particle(1)) ^ 2 + (1 / 2 *
Body_Length_Cylinder(1)) ^ 2) Then
 pb.Text15.Text = dL
 intop = ((dL * U) * (2 / 3 * (dL * U) * (1 - Cst) * ((R_Max * U) ^ 3 -
(R_Min * U) ^ 3) - 1 / 2 * (Sin(Theta_max) ^ 2) * ((R_Max * U) ^ 4 -
(R_Min * U) ^ 4))) + 1 / 5 * ((R_Max * U) ^ 5 - (R_Min * U) ^ 5) * (4 / 3 -
Cst - 1 / 3 * Cst ^ 3)
 Else
 intop = ((dL * U) * (2 / 3 * (dL * U) * (1 - Cst) * ((R_Max * U) ^ 3 -
(R_Min * U) ^ 3) + 1 / 2 * (Sin(Theta_max) ^ 2) * ((R_Max * U) ^ 4 -
(R_Min * U) ^ 4))) + 1 / 5 * ((R_Max * U) ^ 5 - (R_Min * U) ^ 5) * (4 / 3 -
Cst - 1 / 3 * Cst ^ 3)
 End If
```

```
 temp = Radius_Particle(1) / Bottom_Radius_Cylinder(1)
 If temp >= 1# Then
   Theta_max = Pi / 2#
 Else
   Theta_max = Abs(Atn(temp / Sqr(1# - temp ^ 2)))
 End If
 Cst = Cos(Theta_max)
 R_Max = Bottom_Radius_Cylinder(1)
 R_Min = R_Max * Cst
 dL = Sqr((Xo_Particle(1) - Cx_Bottom(1)) ^ 2 + (Yo_Particle(1) -
Cy_Bottom(1)) ^ 2 + (Zo_Particle(1) - Cz_Bottom(1)) ^ 2)
 pb.Text15.Text = dL
 If R_Min > Sqr((Radius_Particle(1)) ^ 2 + (1 / 2 *
Body_Length_Cylinder(1)) ^ 2) Then
   pb.Text15.Text = dL
 Inbot = ((dL * U) * (2 / 3 * (dL * U) * (1 - Cst) * ((R_Max * U) ^ 3 -
(R_Min * U) ^ 3) - 1 / 2 * (Sin(Theta_max) ^ 2) * ((R_Max * U) ^ 4 -
(R_Min * U) ^ 4))) + 1 / 5 * ((R_Max * U) ^ 5 - (R_Min * U) ^ 5) * (4 / 3 -
Cst - 1 / 3 * Cst ^ 3)
 Else
 Inbot = ((dL * U) * (2 / 3 * (dL * U) * (1 - Cst) * ((R_Max * U) ^ 3 -
(R_Min * U) ^ 3) + 1 / 2 * (Sin(Theta_max) ^ 2) * ((R_Max * U) ^ 4 -
(R_Min * U) ^ 4))) + 1 / 5 * ((R_Max * U) ^ 5 - (R_Min * U) ^ 5) * (4 / 3 -
Cst - 1 / 3 * Cst ^ 3)
 End If

 InCent = (Radius_Particle(1) * U) ^ 2 * (1 / 12 *
(Body_Length_Cylinder(1) * U) ^ 3 + (1 / 4 * (Radius_Particle(1) * U) ^ 2 *
Body_Length_Cylinder(1) * U))
 inert = Pi * Density_Particle_Inside(1) * 1000# * (intop + Inbot + InCent)
* 10 ^ (-24)

End Sub
```

Appendix C

C.1 Code for Cylinder Parameter Torque Calculations

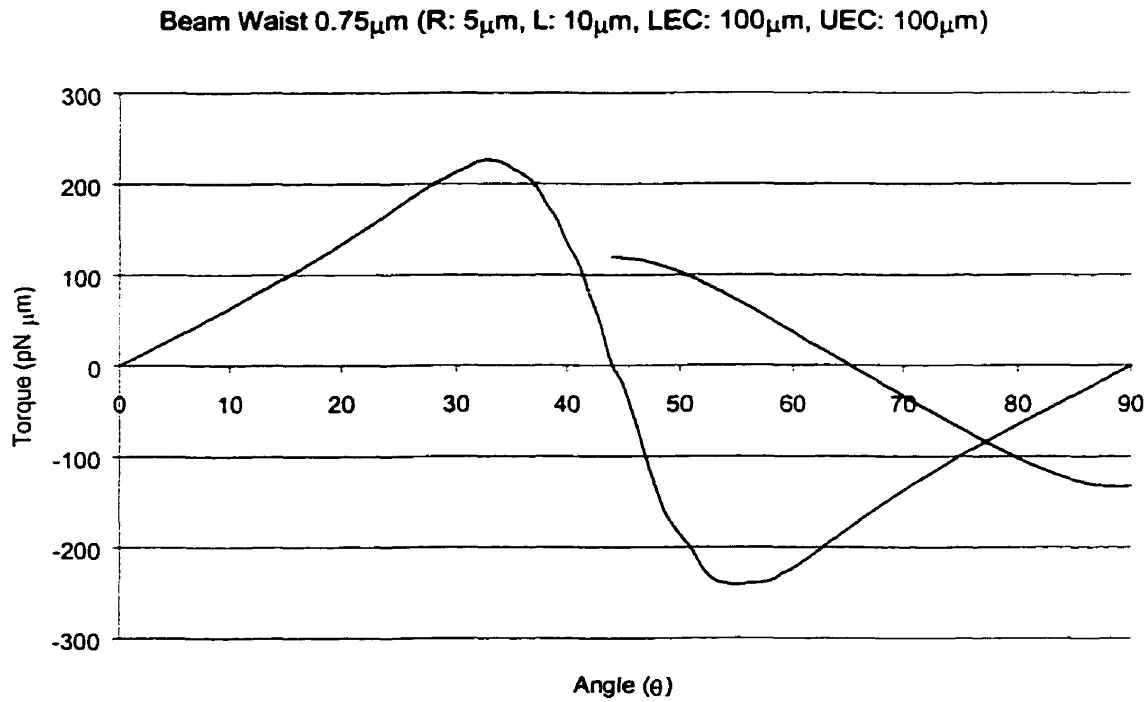
```
Sub mikes_scan()
' this subroutine is to calculate the value of the touque
' for the different parameters such as beam waist
' cylinder waist, radius of endcaps and length of cylinder
 Dim A As Double
 Dim steps As Double
 Dim B As Double
 Dim C As String
 Dim e As Double
 Dim torquevalues As String
 pb.Show
 torquevalues = "f:\mtrapping\mdata\" & Parameters.Text17.Text & ".csv"
 Open torquevalues For Output As #100  'what is the importance of #100
 Print #100, "-------,-------"
 Print #100, "Torque,Results,, File name"; torquevalues
 Print #100, "-------,-------"
 If counter < 4 Then 'check to see if the right number of boxes checked
  Parameters.Label26.Caption = "Please check off the appropriate number
of boxes."
 Else
  If Parameters.Check1.Value = 0 Then 'waist parameters (1)
   C = "Waist"
   A = Val(Parameters.Text1.Text)
   B = Val(Parameters.Text2.Text)
   steps = Val(Parameters.Text11.Text)
  Else
   Print #100, "waist, size, is,,"; Parameters.Text1.Text; " "
  End If
  If Parameters.Check2.Value = 0 Then 'radius of beam parameters (2)
   C = "Radius"
   A = Val(Parameters.Text3.Text)
   B = Val(Parameters.Text4.Text)
   steps = Val(Parameters.Text14.Text)
  Else
```

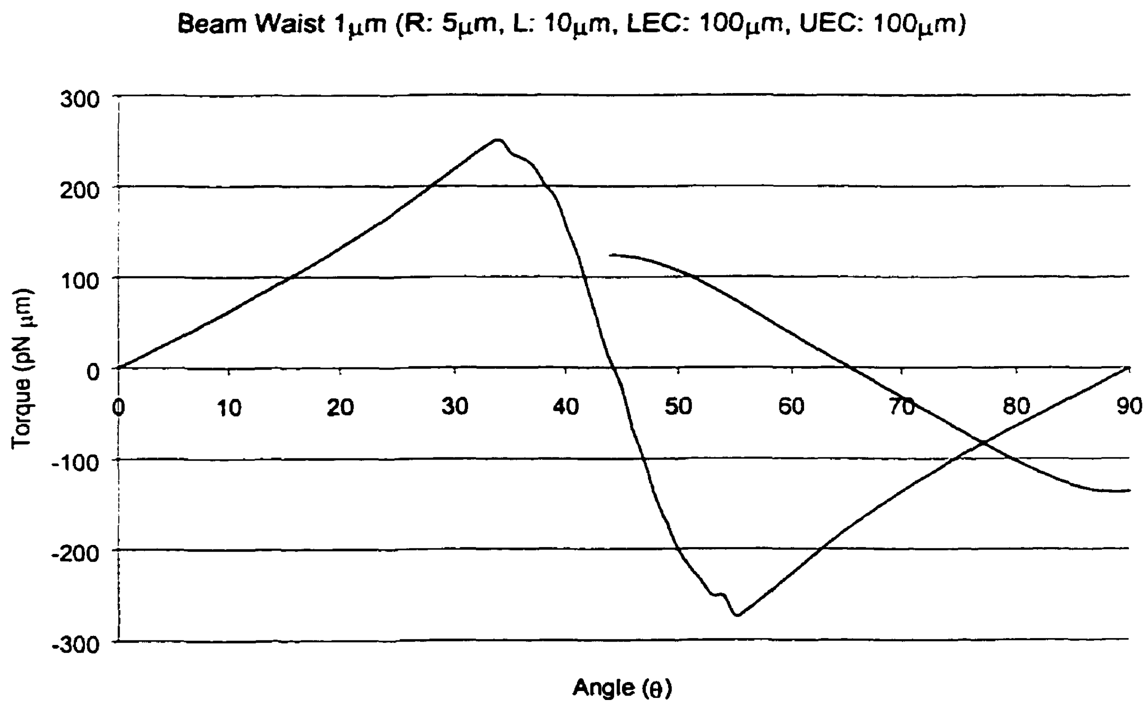
```
 Print #100, "Beam,radius,is,,"; Parameters.Text3.Text; " "
End If
If Parameters.Check3.Value = 0 Then 'length of cylinder (3)
 C = "Length"
 A = Val(Parameters.Text5.Text)
 B = Val(Parameters.Text6.Text)
 steps = Val(Parameters.Text15.Text)
Else
 Print #100, "Length,of,cylinder,is,"; Parameters.Text5.Text; " "
End If
If Parameters.Check4.Value = 0 Then 'radius of left endcap (4)
 C = "rleft"
 A = Val(Parameters.Text7.Text)
 B = Val(Parameters.Text8.Text)
 steps = Val(Parameters.Text12.Text)
Else
 Print #100, "Radius,of,left,endcap,"; Parameters.Text7.Text; " "
End If
If Parameters.Check5.Value = 0 Then 'radius of right endcap (5)
 C = "rright"
 A = Val(Parameters.Text9.Text)
 B = Val(Parameters.Text10.Text)
 steps = Val(Parameters.Text13.Text)
Else
 Print #100, "Radius,of,right,endcap,"; Parameters.Text9.Text; " "
End If
Print #100, " "
Print #100, ",Theta"; " "
Print #100, C; ","; 'this allows the first space to be unoccupied

For Theta_Particle(1) = 0# To 90# Step 1# 'this prints the values of theta
across
 Print #100, Theta_Particle(1); ",";
Next Theta_Particle(1)
Print #100, " "
For e = A To B Step steps
Call Inert_a(inert)
Print #100, e; ",";
 pb.Text21.Text = A
 pb.Text22.Text = e
```

```
pb.Text23.Text = B
pb.Text17.Text = steps
For Theta_Particle(1) = 0# To 90# Step 1#
 pb.Text16.Text = Theta_Particle(1)
 If Parameters.Check1.Value = 0 Then '***** (1) *****
  Waist_Beam(1) = e
 Else
  Waist_Beam(1) = Val(Parameters.Text1.Text)
 End If
 If Parameters.Check2.Value = 0 Then '***** (2) *****
  Radius_Particle(1) = e
 Else
  Radius_Particle(1) = Val(Parameters.Text3.Text)
 End If
 If Parameters.Check3.Value = 0 Then '***** (3) *****
  Body_Length_Cylinder(1) = e
 Else
  Body_Length_Cylinder(1) = Val(Parameters.Text5.Text)
 End If
 If Parameters.Check4.Value = 0 Then '***** (4) *****
  Bottom_Radius_Cylinder(1) = e
 Else
  Bottom_Radius_Cylinder(1) = Val(Parameters.Text7.Text)
 End If
 If Parameters.Check5.Value = 0 Then '***** (5) *****
  Top_Radius_Cylinder(1) = e
 Else
  Top_Radius_Cylinder(1) = Val(Parameters.Text9.Text)
 End If
 Lo_Particle(1) = Sin(Theta_Particle(1) * Pi / 180#)
 Mo_Particle(1) = 0#
 No_Particle(1) = Cos(Theta_Particle(1) * Pi / 180#)
 Call cylinder_top_bottom_positions(1)
 Stop_Calculation = "go"
 Call default_particle_save_dynamic
 Call default_beam_save_dynamic
 X_Y_Z_Scan_Dynamic = "go"
 Call xyzscan_initial_positions_particles
 Call static_calculations
 pb.Refresh
```

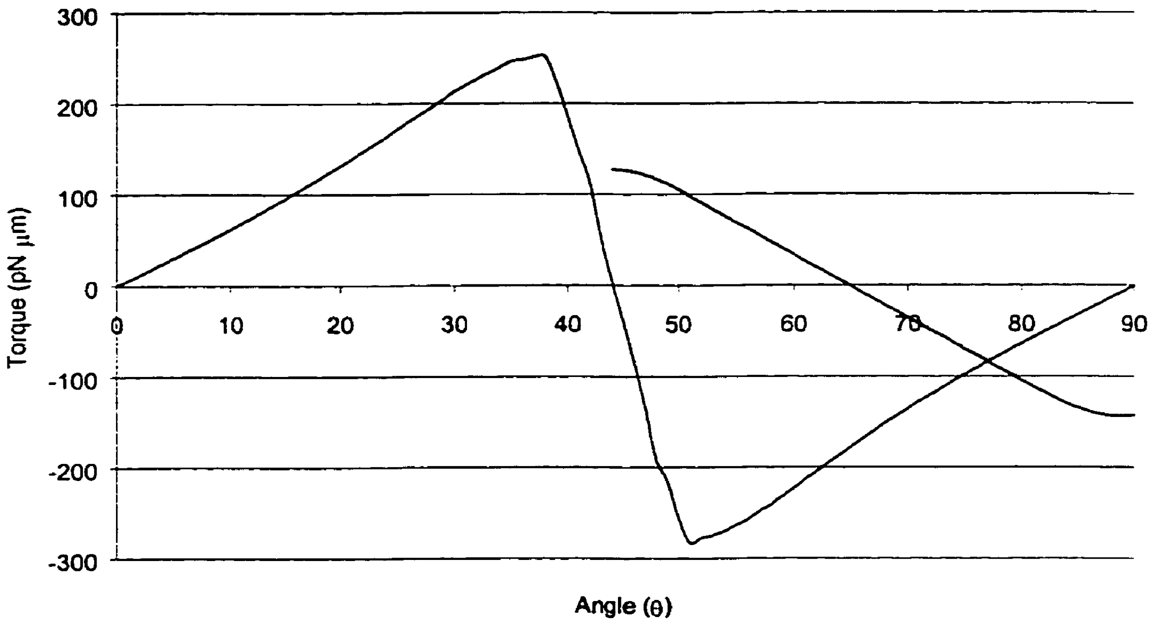
```
      Print #100, Torque_Y_Final(1) * 1000000000000#; ","; 'units of
picoNewtons*micrometers
      If Stop_Calculation <> "go" Then Exit For
      Call default_particle_load_dynamic
      Call default_beam_load_dynamic
    Next Theta_Particle(1)
    Print #100, " "  'down the first column
    If Stop_Calculation <> "go" Then Exit For
  Next e
 Print #100, "Radial,Scan,Nuber,=,"; calculate.Text1.Text; " "
 Print #100, "Radial,Scan, Waist,=,"; calculate.Text2.Text; " "
 Print #100, "Angular,Scan,step,=,"; calculate.Text3.Text; " "
 Close 100
 End If
End Sub
```

C.2 Plots of Torque versus Angle for Beam Waists

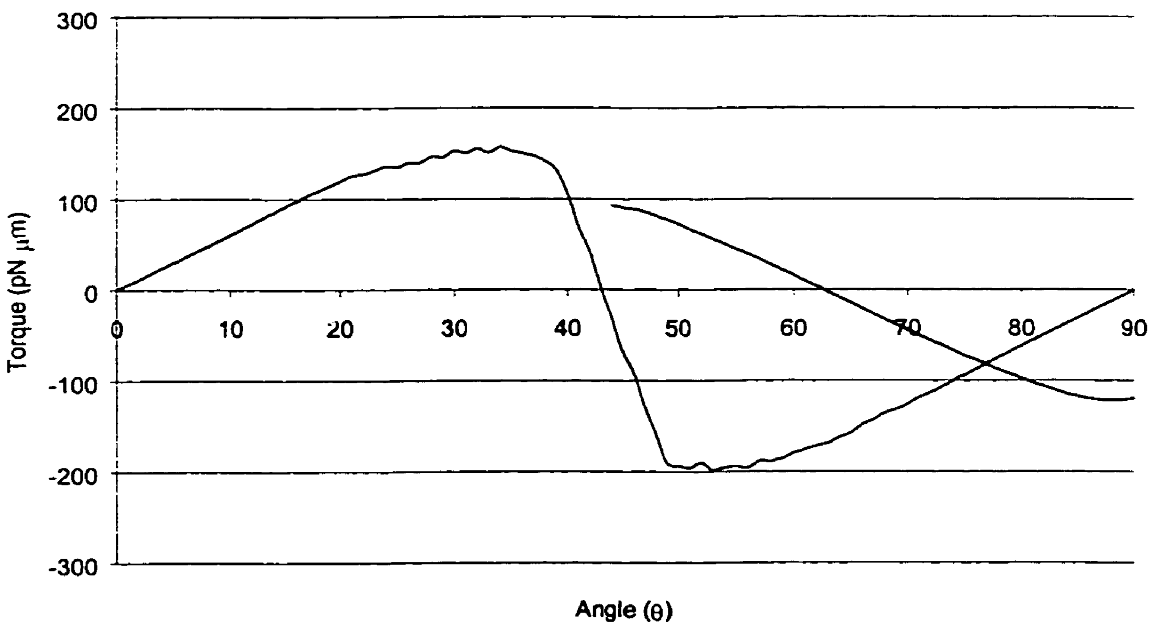




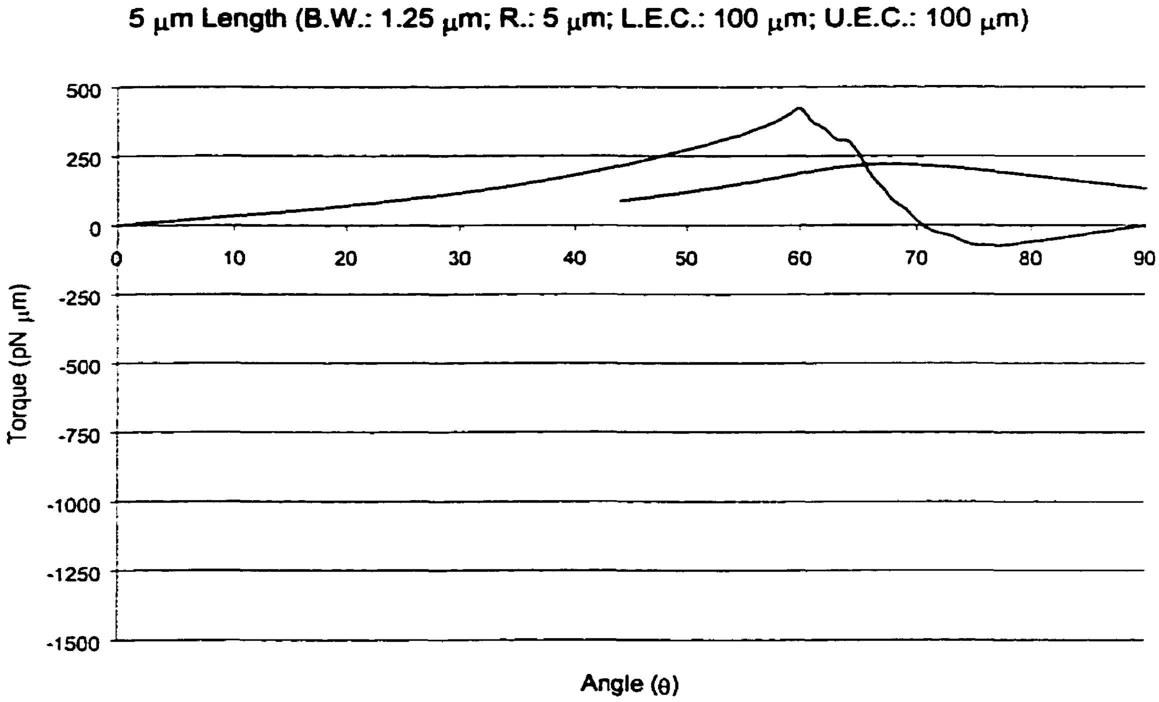
Beam Waist 1.5μm (R: 5μm, L: 10μm, LEC: 100μm, UEC: 100μm)

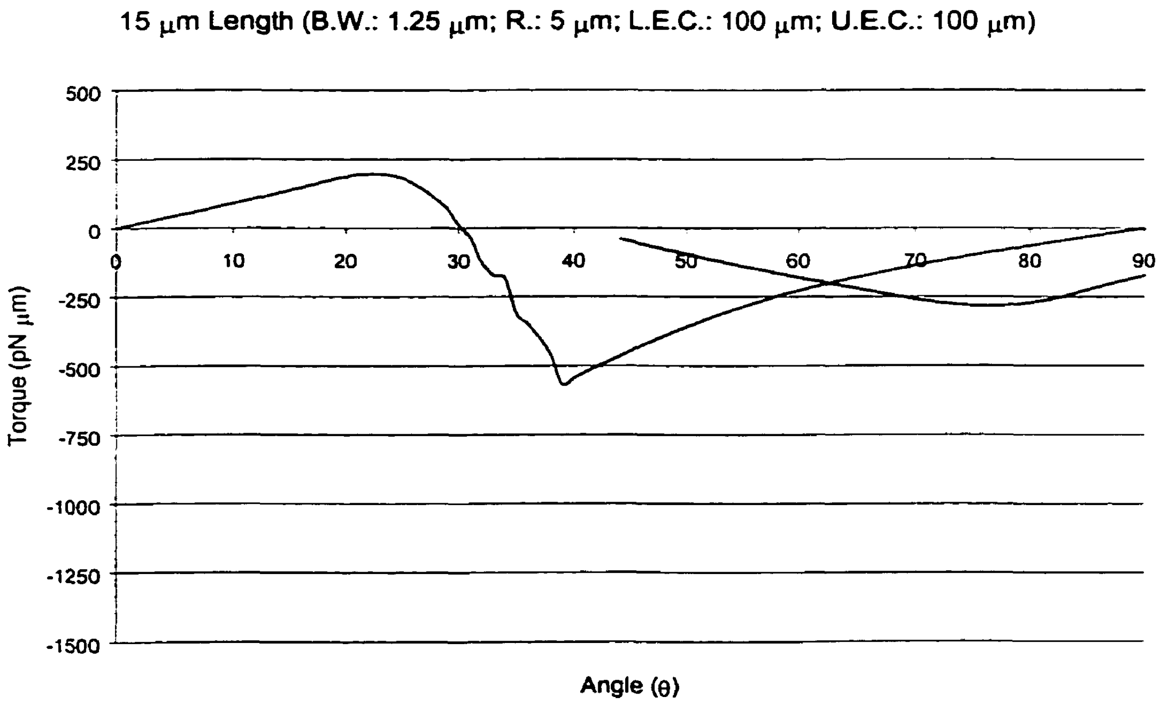


Beam Waist 2.5μm (R: 5μm, L: 10μm, LEC: 100μm, UEC: 100μm)

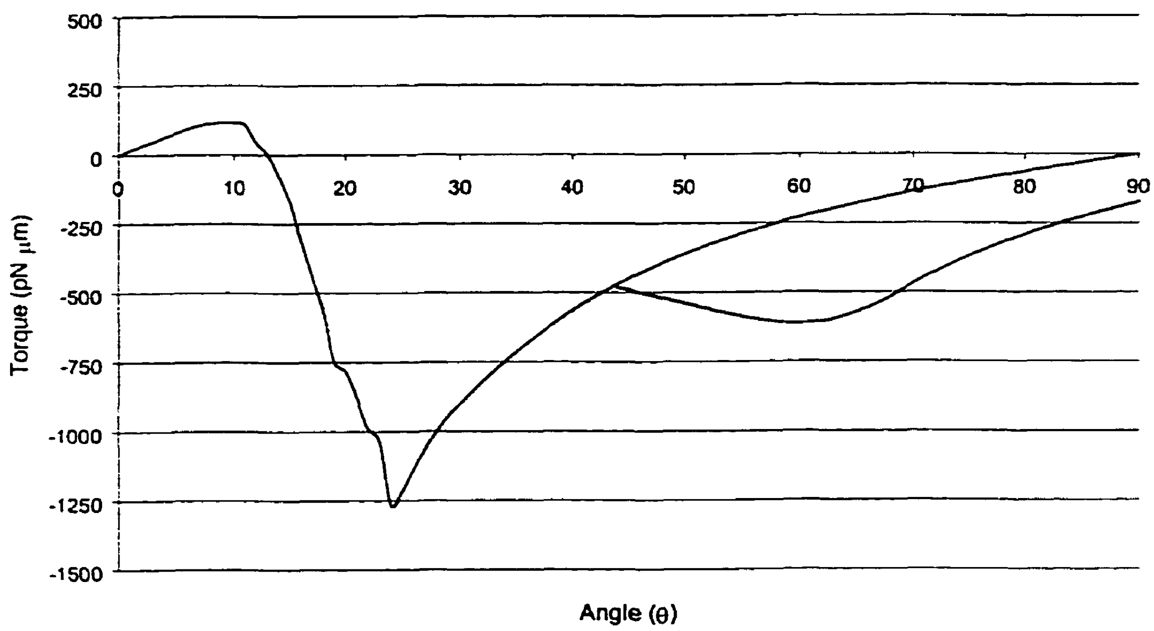


C.3 Plot of Torque versus Angle for Cylinder Length

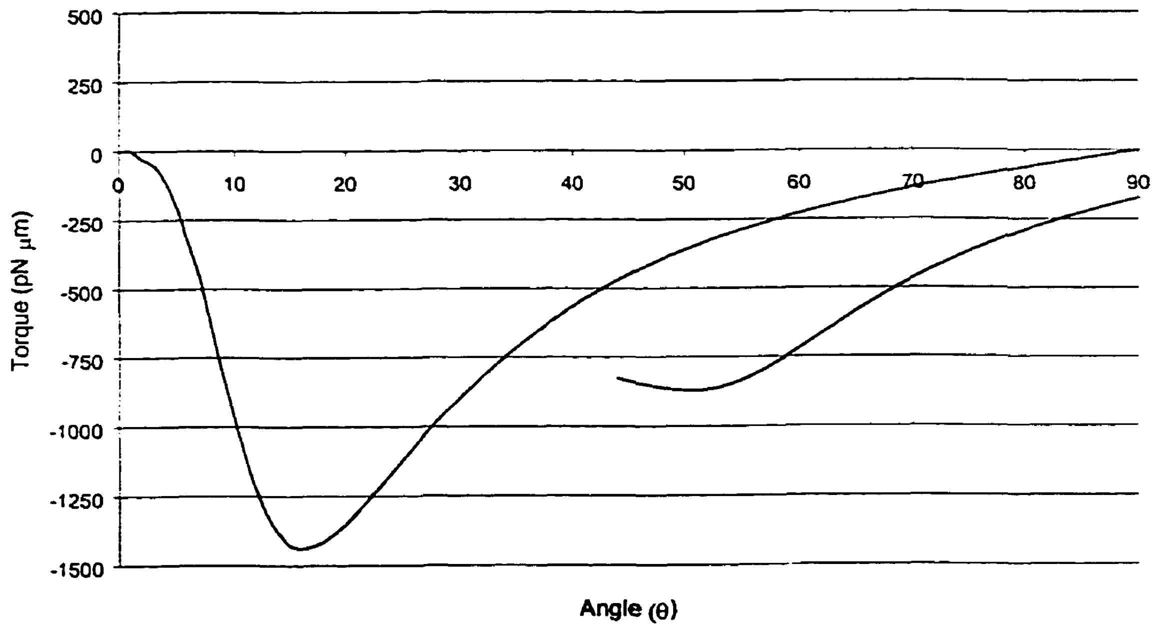




30 μm Length (B.W.: 1.25 μm; R.: 5 μm; L.E.C.: 100 μm; U.E.C.: 100 μm)

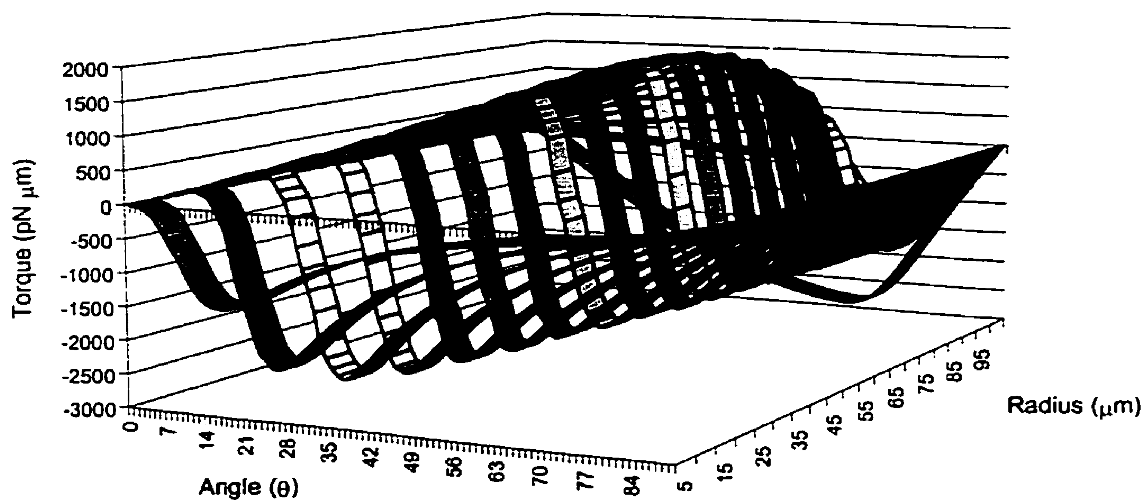


100 μm Length (B.W.: 1.25 μm; R.: 5 μm; L.E.C.: 100 μm; U.E.C.: 100 μm)

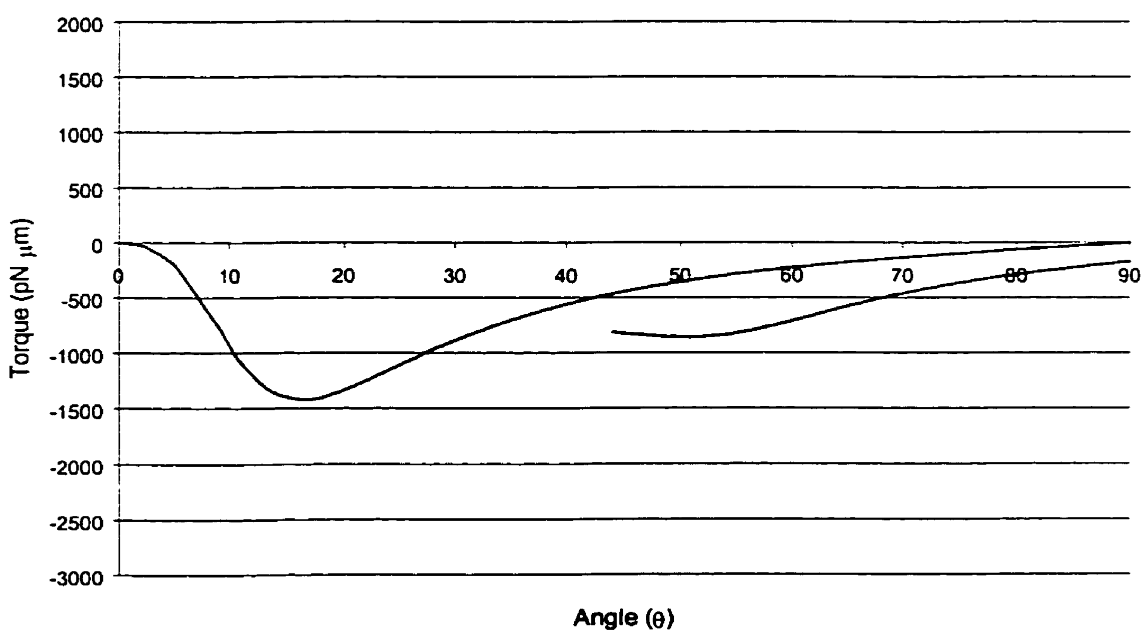


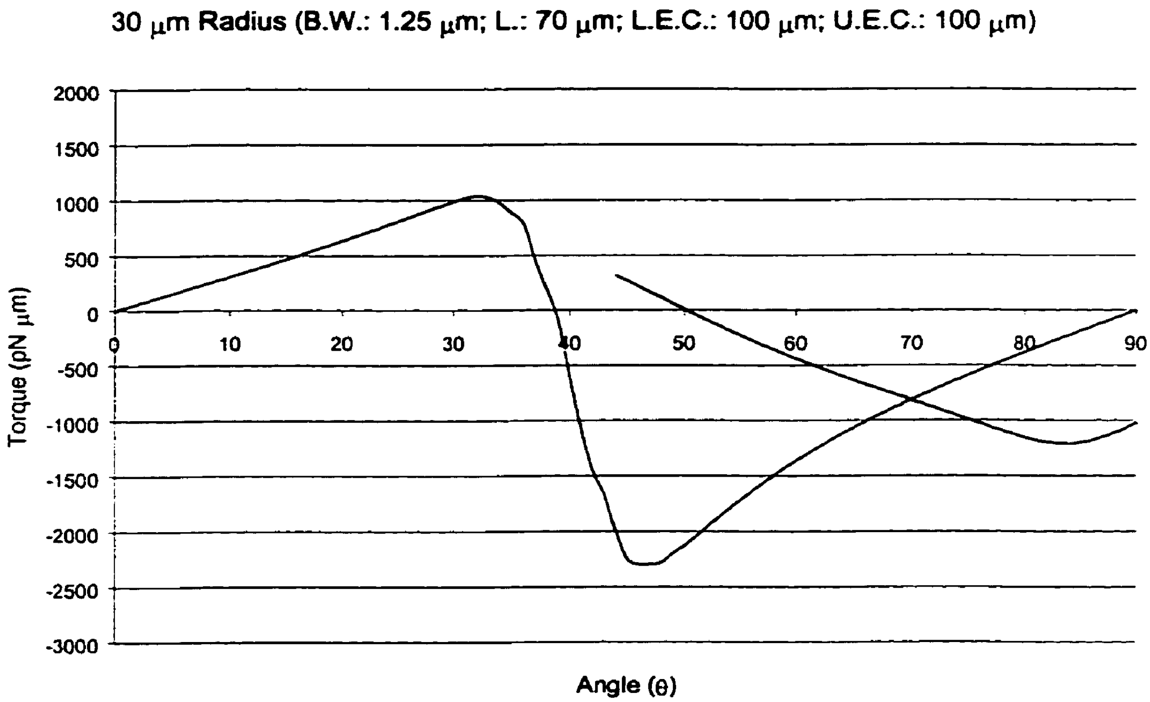
C.4 Plots of Torque versus Angle for Cylinder Radius

Radius Scan (B. W.: 1.25 μm; L.: 70 μm; L.E.C.: 100 μm; U.E.C.: 100 μm)

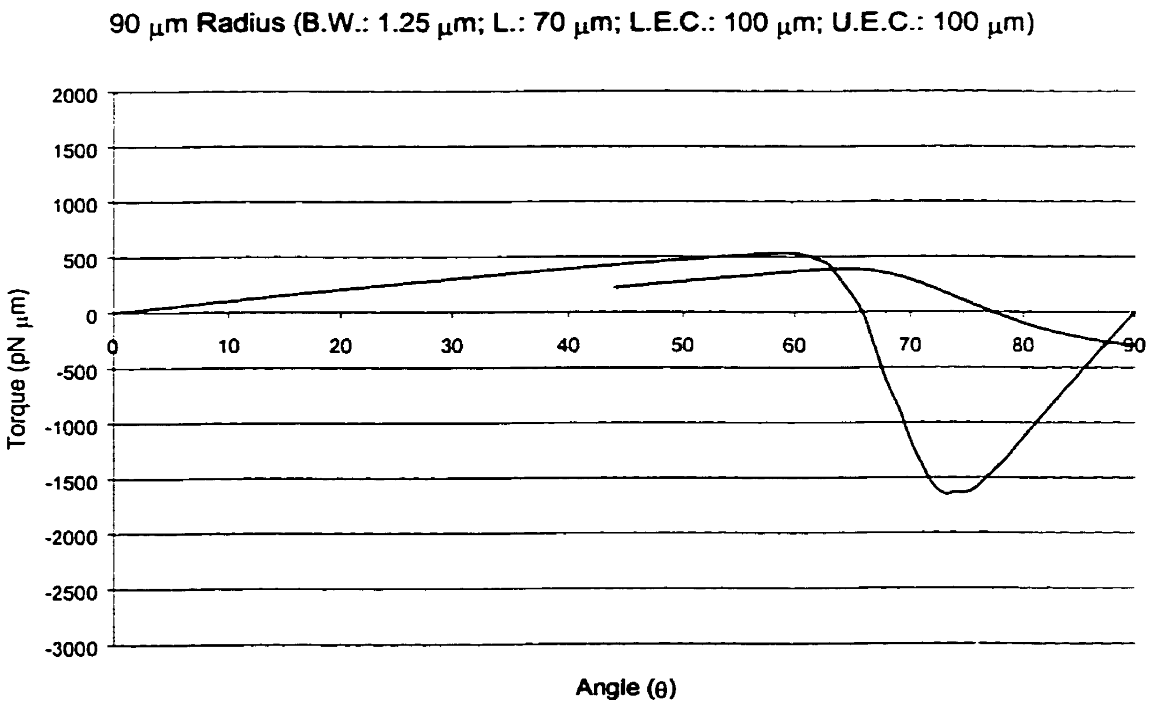


5 μm Radius (B.W.: 1.25 μm; L.: 70 μm; L.E.C.: 100 μm; U.E.C.: 100 μm)





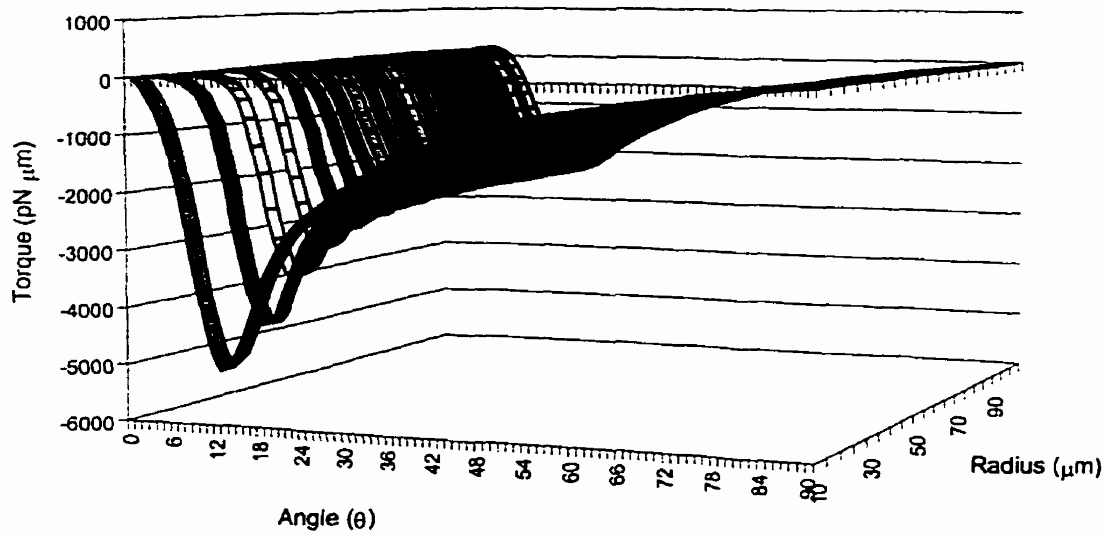
30 μm Radius (B.W.: 1.25 μm; L.: 70 μm; L.E.C.: 100 μm; U.E.C.: 100 μm)
2000
1500
1000
500
0
-500
-1000
-1500
-2000
-2500
-3000
Torque (pN μm)
0
10
20
30
40
50
60
70
80
90
Angle (θ)



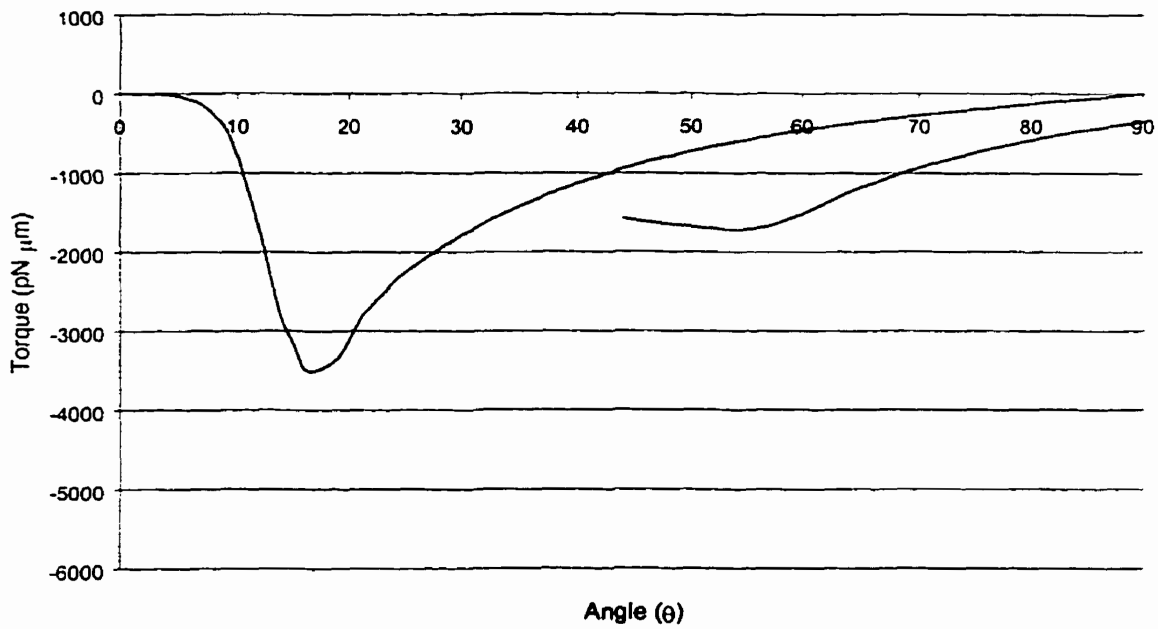
90 μm Radius (B.W.: 1.25 μm; L.: 70 μm; L.E.C.: 100 μm; U.E.C.: 100 μm)
2000
1500
1000
500
0
-500
-1000
-1500
-2000
-2500
-3000
Torque (pN μm)
0
10
20
30
40
50
60
70
80
90
Angle (θ)

C.5 Plots of Torque versus Angle for Lower End-Cap Radius

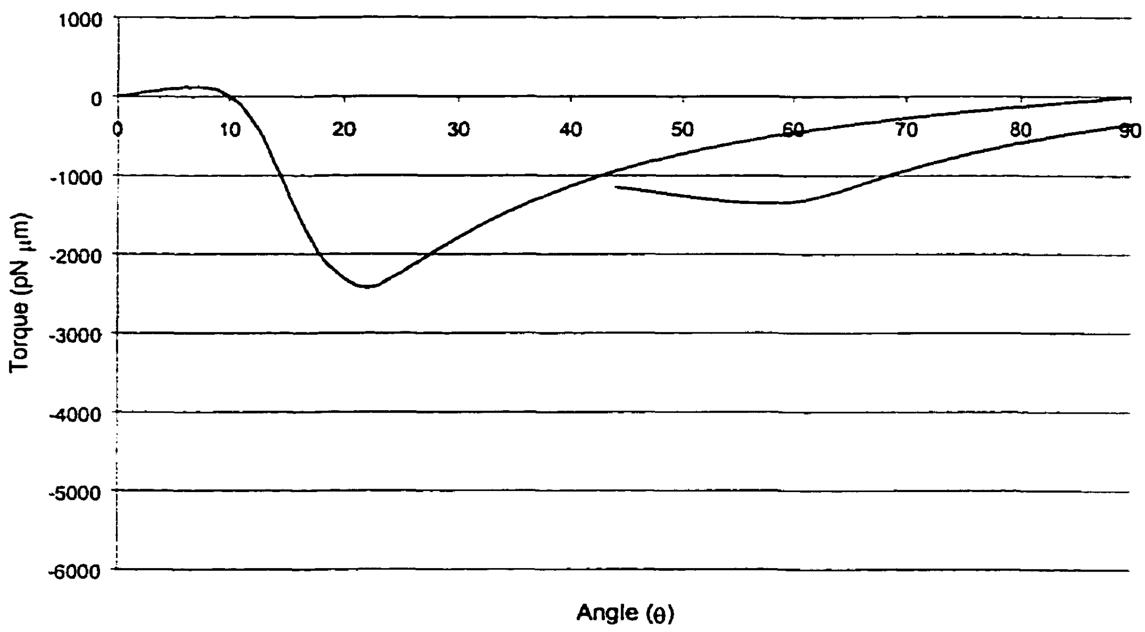
L.E.C. Scan (B. W.: 1.25 μm; L.: 70 μm; R.: 10 μm; U.E.C.: 100 μm)



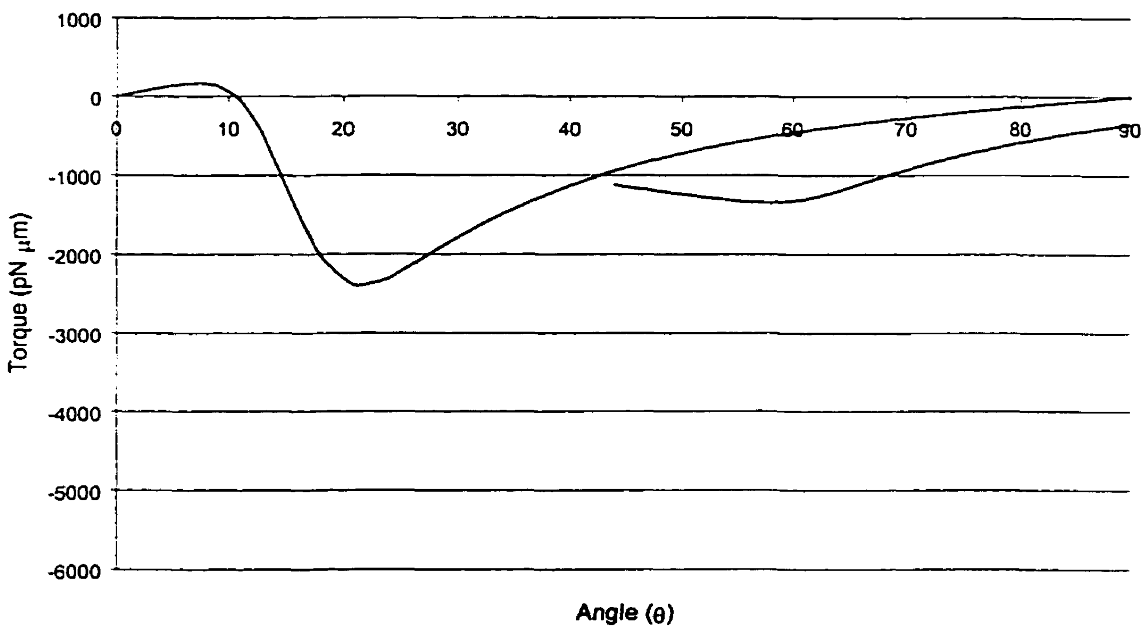
20 μm L. E. C. (B. W.: 1.25 μm; L.: 70 μm; R.: 10 μm; U. E. C. 100 μm)



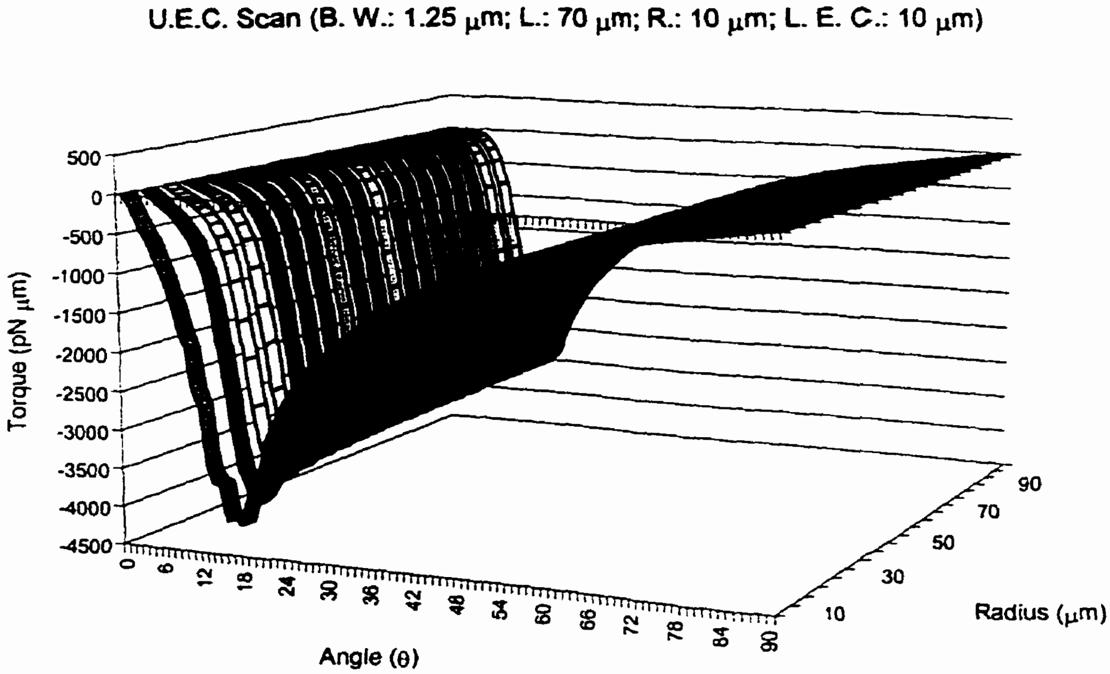
50 μm L. E. C. (B. W.: 1.25 μm; L.: 70 μm; R.: 10 μm; U. E. C. 100 μm)

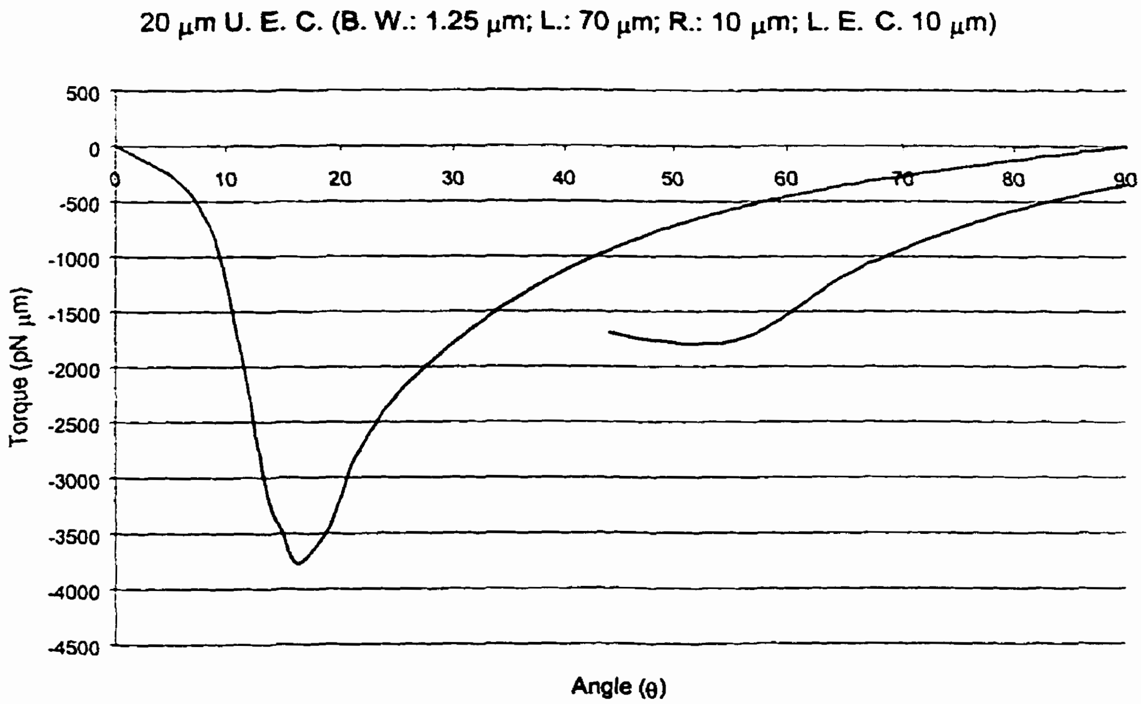


90 μm L. E. C. (B. W.: 1.25 μm; L.: 70 μm; R.: 10 μm; U. E. C. 100 μm)

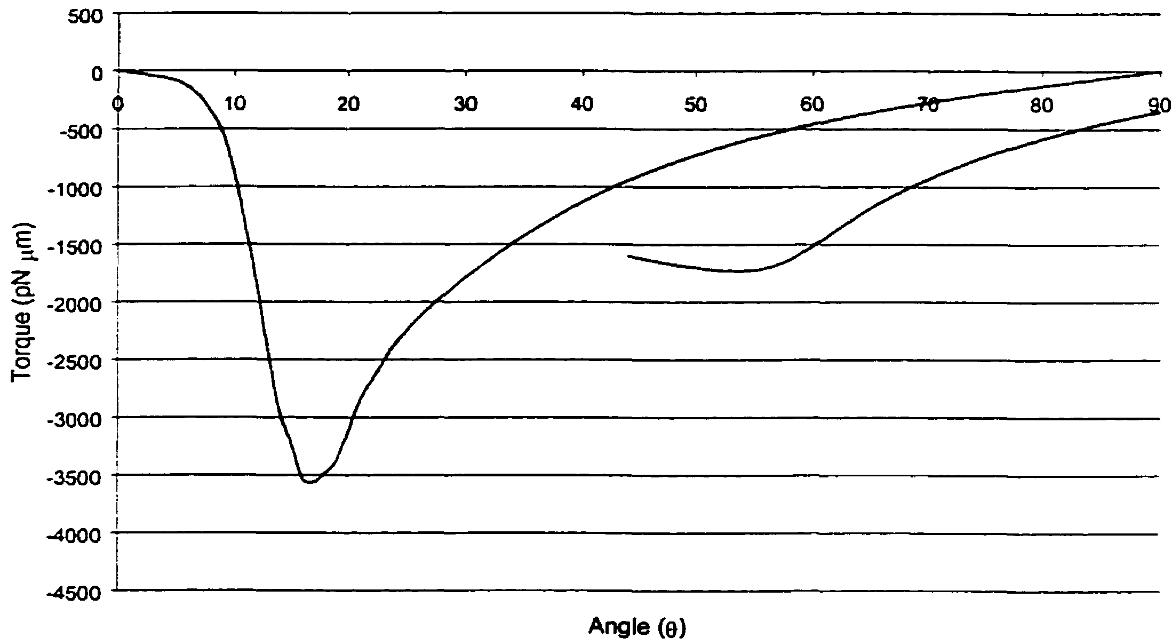


C.6 Plots of Torque versus Angle for Upper End-Cap Radius





50 μm U. E. C. (B. W.: 1.25 μm; L.: 70 μm; R.: 10 μm; L. E. C. 10 μm)



90 μm U. E. C. (B. W.: 1.25 μm; L.: 70 μm; R.: 10 μm; L. E. C. 10 μm)