# Modeling the Magnetron Toy

Chris DuBois

May 9, 2005

## Abstract

Many real-world phenomena are often too complex for us to capture entirely using mathematical tools, but a careful approach enables an exploration of that complexity and may provide insight into the phenomena. The author found a simple desk toy, the "Magnetron," to be a perfect target for exploration: while simply designed and constructed, one quickly realizes the system has very complex dynamics. The toy features two rotors, spaced slightly apart, with magnets mounted at the end of each arm; manually spinning one rotor causes the other rotor to spin due to the interactions between the magnetic fields. In this paper we present a mathematical model that successfully reproduces several of the toy's behaviors. In the future we hope to explore the transitions between various equilibria of the system.


## The Magnetron

The Magnetron toy has two plastic rotors mounted on a wooden block by metal spindles that serve as axles. Each rotor has three evenly spaced arms. At the end of each arm, there is a small bar magnet mounted inside a plastic housing with its North pole oriented away from the rotor center. The two rotors are spaced such that they nearly touch when they are closest to each other. Since dipoles are subject to forces due to external magnetic fields, the fields produced by the magnets in one rotor affect the rotation of the opposite rotor and vice versa.

## Introduction

There are several interesting aspects of this physical system that become apparent after a week-long obsession with this toy. First, the system has a stable equilibrium when two arms are pointing towards the middle but not directly at each other. This can be explained by the magnets wanting to align themselves in the other magnet's field. The system has two unstable equilibria: when the rotor arms are as far away from each other as possible and when two arms are positioned directly towards each other. Additionally, there is a steady state when one rotor is moving fast and the other is such that one arm points directly away from the system center.

Aside from equilibria, the toy shows some cool behavior such as "momentum transfer" - where one rotor begins with high angular velocity but suddenly stops while the second rotor begins fast rotation from a standstill. It is quite fascinating to watch how the two transfer energy to each other so efficiently without physically touching. Another neat effect is "velocity matching," where the two rotors are spinning in opposite directions such that they keep a nearly constant velocity. Also, there is a bouncing effect that is important for understanding the system since it often happens when the system is winding down to equilibrium.

To the author's knowledge, there is only one other attempt at modeling this particular system [1]. Whereas this paper also models the magnets as point dipoles, I use the system's potential energy to find the equations of motion instead of calculating the individual forces present. This model improves on previous models by incorporating a frictional force (and by successfully modeling the dynamics of the toy!).

# Assumptions of The Model

## 2 Dimensional

Because of the way the two rotors are mounted, rotational motion is restricted a single plane parallel to the base plate. Also, the magnetic fields in the vertical direction are symmetric with respect to this plane, so they do not play an important role in the dynamics of the two rotors. Both of these considerations make a 2D model reasonable.

## Energy

We consider the total energy of the system to be the sum of the kinetic and potential energy. Using fundamental Newtonian principles, we calculate the rotational kinetic energy from the inertia and angular velocity of each rotor. Using equations from electromagnetic theory, we can calculate the potential energy present from the interacting magnetic fields of the dipoles. Approaching the state of the system in terms of total energy allows us to use a Lagrangian formulation of the model to describe the progression of the system.

## Rotational Friction

We make the assumption that the axle provides a rotational frictional force on the rotors that is proportional to each rotor's angular velocity. Since it is impossible that both axles act upon the rotors in exactly the same amount, the coefficient of friction must be slightly different for each rotor. Employing a rotational drag term is a reasonable choice since it is impossible for the axles to be perfectly frictionless and an important choice since it provides an avenue for energy loss. (We know the toy exhibits energy loss since, in practice, the two rotors eventually find a stable, motionless equilibrium after initially having angular velocity.)

## Higher Order Frictions

We ignore the effects of air resistance and other higher order drags to clarify the effects of the first order terms. And after considering the geometry of the rotor arms and the angular velocities achieved, it is likely that the effects of air resistance are minimal.

## Mass

We will consider the magnets to each have equal mass. We will consider the rotor structures to be massless so that calculating moments of inertia of each rotor is easier. This is a good simplification since we can vary the moments of inertia by just changing the mass of the magnets rather than needing to consider the rotor structure to have mass.

## The Magnets As Point-Dipoles

Each of the six magnets is a cylindrical bar magnet made of a ferromagnetic material (iron most likely). Bar magnets have two parts, designated as the North pole and the South pole, and thus fall under the category of "dipoles." In our notation, each magnet will be represented by its "magnetic moment vector" whose direction will be from the magnet's South pole to its North pole and whose magnitude will represent the magnet's strength. Most importantly, we will consider each magnet to be a "point dipole," which means we will consider them to have an infinitesimally small size.

The advantages of this are primarily computational: the equations that govern three-dimensional dipoles are complicated, requiring integrals of vector potentials over given volumes. To make matters worse, we would need to compute the potential of a three-dimensional dipole in the field of a three-dimensional dipole for each timestep.

In reality, these bar magnets are comprised of domains of atoms that are magnetically aligned; the magnet's properties come from the fact that some domains are more dominant than others and thus provide a dominant direction to the overall external field [3]. In the future, each magnet will be described as many closely-packed point dipoles to account for the physical size of the bar magnets. This is a reasonable approach since the magnetic fields of 3D dipoles are, in a rough sense, calculated by integrating vector potentials of individual point dipoles.

## Dipoles and Potential Energy

The total energy of the system, due to the assumptions above, will simply be the sum of the kinetic and potential energies of the magnets. The kinetic energy can be found from the rotational velocity of the rotors and the masses of the magnets. The potential energy of the system can be found using principles of electromagnetism. It is a fact that magnetic dipoles have a potential energy when in an external magnetic field. From experience, we know that two bar magnets arranged in a particular manner experience attractive or repulsive forces; in other arrangements, it is clear they experience a torque - a desire to twist to align themselves. Technically, this is an interaction between the moment of one magnet and the external field of the other magnet. We can calculate this potential energy, and in this manner find the potential energy of each magnet interacting with the fields of the other five magnets, summing these to find the total potential energy.

## Theory

It is helpful to consider the model in terms of vector quanities. Throughout the following description, it may be helpful to refer to Figure 1. Let the center of Rotor A (with magnets 1, 2, and 3) be the origin, and let the center of Rotor B (with magnets 4, 5, and 6) be located at $\mathbf{L}$. Let $\mathbf{m}_i$ represent the magnetic moment vector of magnet $i$, where $i = 1, 2, 3, 4, 5, 6$ represents each magnet. Let $\mathbf{r}_i$ be the position of magnet $i$ with respect to the origin, and let $\mathbf{r}_i^* = \mathbf{r}_i - \mathbf{L}$ be the position of magnet $i$ with respect to the center of the second rotor. This is important because we need to know the relative position between magnets: let the vector from magnet $j$ to magnet $i$ be denoted $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j = \mathbf{r}_i - \mathbf{r}_j^* - \mathbf{L}$. Therefore, the distance between magnets $i$ and $j$ is $(\mathbf{r}_{ij} \cdot \mathbf{r}_{ij})^{1/2}$.

The current angular position of the left and right rotor will be denoted $\theta$ and $\phi$ respectively, measuring the position of rotors 1 and 4 relative to the line connecting the two rotors. Thus, $\mathbf{r}_i = R[\cos(\theta + \frac{2\pi}{3}(i-1)), \sin(\theta + \frac{2\pi}{3}(i-1))]$ and $\mathbf{r}_j^* = R[-\cos(\phi + \frac{2\pi}{3}(j-1)), \sin(\phi + \frac{2\pi}{3}(j-1))]$. For example, $\mathbf{r}_5^* = R[-\cos(\phi + \frac{2\pi}{3}), \sin(\phi + \frac{2\pi}{3})]$, as expected.

Similarly, $\mathbf{m}_i$ can be written in terms of $\theta$ and $\phi$. Since all the dipoles point away from the center of the rotors, $\mathbf{m}_i/|\mathbf{m}_i| = \mathbf{r}_i/|\mathbf{r}_i|$ (ie. they have the same orientation). Therefore we quickly see that $\mathbf{m}_i = m[\cos(\theta + \frac{2\pi}{3}(i-1)), \sin(\theta + \frac{2\pi}{3}(i-1))]$.

For a point dipole at $\mathbf{r}_i$ in an external magnetic field, the potential energy is

$$U(\theta, \phi) = -\mathbf{m}_i \cdot \mathbf{B}_i \tag{1}$$

where $\mathbf{m}_i$ is the magnetic moment of dipole $i$ and $\mathbf{B}_i$ is the total external magnetic field vector at $\mathbf{r}_i$ (Equation 15.4, [2]). The external magnetic field of a dipole can be expressed as a vector field that

Table 1.

Relevant constants, variables, and parameters.

| Situational Constants | Description | Value |
|---|---|---|
| $\mathbf{L}$ | position of Rotor A relative to Rotor B | $(2R+\epsilon,0)$ |
| $R$ | length of each rotor arm | 1 |
| $\epsilon$ | distance between magnets at closest approach | .4 |
| $M$ | mass of each magnet | 2 |
| $m$ | $\lvert\mathbf{m}\rvert$=magnitude of each dipole moment | 3 |
| $I$ | moment of inertia of each rotor | 6 |
| $\beta_1$ | coefficient of friction for Rotor A | .2 |
| $\beta_2$ | coefficient of friction for Rotor B | .3 |

| Dynamic Variables | Description |
|---|---|
| $i, j$ | represents a particular magnet. Rotor A=$1, 2, 3$, Rotor B=$4, 5, 6$. |
| $\mathbf{r}_i$ | position of magnet $i$ relative to Rotor A |
| $\mathbf{r}_i^*$ | position of magnet $i$ relative to Rotor B |
| $\mathbf{r}_{ij}$ | the vector with tail $\mathbf{r}_j$ and head $\mathbf{r}_i$ |
| $\mathbf{m}_i$ | the magnetic moment vector for dipole $i$, oriented in the direction from S-N |
| $\theta$ | angular position of magnet 1 relative to $\mathbf{L}$ |
| $\phi$ | angular position of magnet 4 relative to $\mathbf{L}$ |
| $T$ | the total kinetic energy of the system |
| $U$ | the total potential energy of the system |
| $\mathbf{B}_i$ | the total external magnetic field vector for magnet $i$ |
| $\mathbf{B}(\mathbf{m}_j, \mathbf{r}_{ij})$ | the magnetic field vector at $\mathbf{r}_{ij}$ relative to its source $\mathbf{m}_j$ |

depends on the dipole's moment. For a dipole with moment $\mathbf{m}_j$, the field at location $\mathbf{r}_{ij}$ relative to magnet $j$ is given, in cgs units, by (Equation 2, [5])

$$\mathbf{B}(\mathbf{m}_j, \mathbf{r}_{ij}) = \frac{3(\mathbf{m}_j \cdot \mathbf{r}_{ij})(\mathbf{r}_j) - \mathbf{m}_j|\mathbf{r}_{ij}|^2}{|\mathbf{r}_{ij}|^5} \tag{2}$$

which is a result of (Equation 14.36 [2]). Now we can find the total external magnetic field vector on magnet $i$ by summing the fields due to each dipole:

$$\mathbf{B}_i = \sum_{j=1}^{6} \mathbf{B}(\mathbf{m}_j, \mathbf{r}_{ij}) \tag{3}$$

Using Equations 1, 2, and 3, one can calculate the total potential of the system as a function of $\theta$ and $\phi$.

From (Equation 19.4 [2]) the rotational kinetic energy can be written as $T = \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}I\dot{\phi}^2$, which we can plug into the Lagrangian [4] to get

$$\mathcal{L} = T - U(\theta, \phi) = \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}I\dot{\phi}^2 - U(\theta, \phi). \tag{4}$$

The Euler-Lagrange equations are [6]

$$\frac{\partial \mathcal{L}}{\partial \theta} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) = 0 \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \phi} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) = 0 \tag{6}$$

Therefore we have the system of ODE's $I\ddot{\theta} = -\partial U(\theta, \phi)/\partial\theta$ and $I\ddot{\phi} = -\partial U(\theta, \phi)/\partial\phi$. From here, we can solve numerically for solutions to $\theta$ and $\phi$.

# Implementation

The total potential energy of the system was calculated as a function of $\theta$ and $\phi$ using the formulae detailed above and MATLAB's Symbolic Math Toolbox (refer to Appendix A for the code). Similarly, this software was used to find the partial derivatives of $U$ with respect to $\theta$ and $\phi$ (Appendix B). To solve the ODE, we use a MATLAB ODE solver named rk4.m that implements 4th order Runge-Kutta; we use a stepsize of .01.

The software's symbolic output for $U(\theta, \phi)$ included several uses of conj($\theta$) which is the MATLAB command for the complex conjugate of $\theta$. Since all of our angle measurements are real, we replaced conj($\theta$) and conj($\phi$) with $\theta$ and $\phi$ respectively.

# Experimental Setup

To get a glimpse at the progression of the real system, I took data on the angular positions and velocities of one arm on each rotors over approximately 5 second intervals after giving the left rotor an initial impulse. I filmed the toy with a digital camcorder, first manually spinning a rotor then filming until the toy came to equilibrium. The chosen video clips featured several specific behaviors of the Magnetron that I hope my model will replicate in test simulations (e.g. equilibrium points, “momentum transfer”, etc).

The Magnetron toy was provided by Dr. Ami Radunskaya. The equipment used to take data included a Sony digital camcorder connected via FireWire to an Apple PowerBook laptop with iMovie and VideoPoint software. This setup allowed for a capture rate up to 30 fps. I marked one arm on each rotor with chalk for a visual cue so that using DataPoint would be more precise. Analysis in VideoPoint was performed as accurately as possible with the project’s time constraints.

## Discussion

It is insightful to examine the total potential energy of the system in different situations. Consider the left rotor fixed at angles 0, $\pi/6$, and $\pi/3$. We can solve for the total potential energy of the system for various values of $\phi$ ranging from 0 to $2\pi/3$ (the location of the next rotor arm). As depicted in Figure 4a, the potential energy is greatest when $\theta = \phi = 0$ as one would expect: two dipoles are aligned antiparallel with a small separation between them.

By fixing one of the rotors at various angles and solving the ODE, one observes several phenomena that are similar to those of the real toy. First, when the field from Rotor A puts a positive torque on Rotor B and thus some angular acceleration; once one of the magnets comes within a certain distance, Rotor B’s angular position oscillates until it reaches an equilibrium position with no angular velocity (Figure 4b, top). With an initial velocity, we see the second rotor has two possible equilibria symmetric to $\phi = 0$ (Figure 4b, bottom).

If we allow both rotors to move freely, we should consider the potential for values of $\theta$ and $\phi$ ranging from 0 to $2\pi/3$, as depicted by the contour plot in Figure 5a. In simulations using this potential, we begin to see the complex behavior of the toy and in particular we can see behavior that reflects the experimental results for similar initial conditions. For example, Figure 5b illustrates the same “momentum transfer” phenomena we saw previously (Figure 3a): the toy is in a state where the rotors quickly and repeatedly trade having a velocity and having no velocity. Equally important, Figure 5c reveals very similar oscillations in angular position and velocity as Figure 3b where the toy tends toward its stable equilibrium, after starting just shy of the unstable equilibrium at $\phi = \pi/3$. To see this similar relationship between rotors, compare frames 15 to 40 in Fibure 3b and timesteps 90 to 170 in Figure 5c. This shows that the “bouncing” phenomena we found so interesting in the toy has been well reflected by the model.

Lastly, Figure 5d illustrates the progression from one of the steady states that arises often in the toy, where one rotor is motionless and the other rotor is spinning very quickly. The motionless rotor begins to oscillate from side to side until reaching a point where there is not enough kinetic energy to move past the rotor and shows the common pre-equilibrium behavior (as described above).

## Sensitivity Analysis

While a thorough sensitivity analysis was not performed, preliminary experimentation revealed the importance of several variables. First, two variables in particular control the nature of the total potential: $\epsilon$ is the distance between the two rotors at their closest approach and $m$ is the strength of each dipole’s magnetic moment. Roughly speaking, the author found that increasing the value of $\epsilon$ makes the peaks of the potential wider; increasing the value of $m$ makes the peaks taller.

Changing the mass of the magnets allows one to manipulate how fast the rotors respond to a given potential. This is as one might expect, since a lower moment of inertia means the rotor is easier to rotate.

The chosen ODE solver and its respective stepsize appears to have significant effects on the results, but their exact influence is still unclear. The author does know that with high velocities and too large a stepsize, rotors often pass by each other when they would normally "bounce" away in the opposite direction. Further investigation might provide an optimal combination of realism and computational time.

## Conclusion

This paper presents a model for the Magnetron toy that uses a Lagrangian formulation by solving for the total energy of the system, considering the potential energies of dipoles in the fields of other dipoles. Preliminary solutions show that the model successfully captures several of the phenomena of the real toy: "momentum transfers," "bouncing," as well as the correct equilibria and steady states. Some behavior has not been observed in the model, however. The most notable behavior missing often accompanies the velocity matching behavior, where rotors transition from both having positive angular velocity to both having negative angular velocity.

Intuitively, most would consider the toy's dynamics complex, perhaps even describe it as chaotic. In this vein, future work will determine if there is a dependence on initial conditions and approach the solutions from a more analytic standpoint in the hopes of affirming our intuition.

# Bibliography

[1] Jimmy Corno. Chaotic Magnetic Spinner Toy Dynamics (2003). *http://chaosbook.org/projects/Corno/Corno.pdf* (last accessed 5/3/2005).

[2] Richard P. Feynman. *Feynman Lectures on Physics.* 1989.

[3] R. Nave. Magnetic Domains and Ferromagnetism. *http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html* (last accessed 5/9/2005).

[4] Eric W. Weisstein. "Lagrangian." From ScienceWorld –A Wolfram Web Resource. *http://scienceworld.wolfram.com/physics/Lagrangian.html* (last accessed 5/9/2005).

[5] Eric W. Weisstein. "Magnetic Dipole." From ScienceWorld –A Wolfram Web Resource. *http://mathworld.wolfram.com/Euler-LagrangeDifferentialEquation.html* (last accessed 5/3/2005).

[6] Eric W. Weisstein. "Euler-Lagrange Differential Equation." From MathWorld–A Wolfram Web Resource. *http://scienceworld.wolfram.com/physics/MagneticDipole.html* (last accessed 5/3/2005).

# Appendix A

```
% find_p.m
% MATLAB code for Magnetron Project - Chris DuBois
% Symbolically find U as a function of theta and phi.

clear;
syms theta phi L C R epsilon m m1 m2 m3 m4 m5 m6 r r1 r2 r3 r4 r5 r6;
syms U U1 U2 U3 U4 U5 U6;
syms B1 B2 B3 B4 B5 B6;
syms B14 B15 B16 B24 B25 B26 B34 B35 B36 B41 B42 B43 B51 B52 B53 B61 B62 B63;
syms thetadotdot phidotdot I;

% Find mi and ri in vector form.
L=2*R+epsilon;
m1=m*[cos(theta);sin(theta)];
m2=m*[cos(theta+2*pi/3);sin(theta+2*pi/3)];
m3=m*[cos(theta+4*pi/3);sin(theta+4*pi/3)];
m4=m*[-cos(phi);sin(phi)];
m5=m*[-cos(phi+2*pi/3);sin(phi+2*pi/3)];
m6=m*[-cos(phi+4*pi/3);sin(phi+4*pi/3)];
r1=[cos(theta);sin(theta)];
r2=[cos(theta+2*pi/3);sin(theta+2*pi/3)];
r3=[cos(theta+4*pi/3);sin(theta+4*pi/3)];
r4=[L-cos(phi);sin(phi)];
r5=[L-cos(phi+2*pi/3);sin(phi+2*pi/3)];
r6=[L-cos(phi+4*pi/3);sin(phi+4*pi/3)];

% Magnetic Field on Rotor 1
ri=r1;
rj=r4;
mi=m1;
mj=m4;
r=ri-rj;
rm=dot(r,r)^.5;
B14=(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r1;
rj=r5;
mi=m1;
mj=m5;
r=ri-rj;
rm=dot(r,r)^.5;
B15=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B15=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;

ri=r1;
rj=r6;
mi=m1;
```

```
mj=m6;
r=ri-rj;
rm=dot(r,r)^.5;
B16=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B16=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;

B1=B14+B15+B16;

% Magnetic Field on Rotor 2
ri=r2;
rj=r4;
mi=m2;
mj=m4;
r=ri-rj;
rm=dot(r,r)^.5;
B24=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B24=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r2;
rj=r5;
mi=m2;
mj=m5;
r=ri-rj;
rm=dot(r,r)^.5;
B25=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B25=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r2;
rj=r6;
mi=m2;
mj=m6;
r=ri-rj;
rm=dot(r,r)^.5;
B26=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B26=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;

B2=B24+B25+B26;

% Magnetic Field on Rotor 3
ri=r3;
rj=r4;
mi=m3;
mj=m4;
r=ri-rj;
rm=dot(r,r)^.5;
B34=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B34=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r3;
rj=r5;
mi=m3;
```

```
mj=m5;
r=ri-rj;
rm=dot(r,r)^.5;
B35=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B35=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r3;
rj=r6;
mi=m3;
mj=m6;
r=ri-rj;
rm=dot(r,r)^.5;
B36=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B36=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;

B3=B34+B35+B36;

% Magnetic Field on Rotor 4
ri=r4;
rj=r1;
mi=m4;
mj=m1;
r=ri-rj;
rm=dot(r,r)^.5;
B41=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B41=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r4;
rj=r2;
mi=m4;
mj=m2;
r=ri-rj;
rm=dot(r,r)^.5;
B42=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B42=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r4;
rj=r3;
mi=m4;
mj=m3;
r=ri-rj;
rm=dot(r,r)^.5;
B43=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B43=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;

B4=B41+B42+B43;

% Magnetic Field on Rotor 5
ri=r5;
rj=r1;
mi=m5;
```

```
mj=m1;
r=ri-rj;
rm=dot(r,r)^.5;
B51=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B51=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r5;
rj=r2;
mi=m5;
mj=m2;
r=ri-rj;
rm=dot(r,r)^.5;
B52=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B52=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r5;
rj=r3;
mi=m5;
mj=m3;
r=ri-rj;
rm=dot(r,r)^.5;
B53=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B53=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;

B5=B51+B52+B53;

% Magnetic Field on Rotor 6
ri=r6;
rj=r1;
mi=m6;
mj=m1;
r=ri-rj;
rm=dot(r,r)^.5;
B61=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B61=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r6;
rj=r2;
mi=m6;
mj=m2;
r=ri-rj;
rm=dot(r,r)^.5;
B62=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
B62=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;
ri=r6;
rj=r3;
mi=m6;
mj=m3;
r=ri-rj;
rm=dot(r,r)^.5;
B63=(3*(ri-rj)*dot(mj,ri-rj)-mj*dot(ri-rj,ri-rj)^2)/dot(ri-rj,ri-rj)^5;
```

```
B63=C*(3*r*dot(mj,r)-mj*rm^2)/rm^5;

B6=B61+B62+B63;

% Total Potential Energies
U1=-dot(m1,B1);
U2=-dot(m2,B2);
U3=-dot(m3,B3);
U4=-dot(m4,B4);
U5=-dot(m5,B5);
U6=-dot(m6,B6);

Utotal=U1+U2+U3+U4+U5+U6;
save Utotal;
```

# Appendix B

```
% U.m
% MATLAB code for Magnetron Project - Chris DuBois
% Here, we show MATLAB's solution for Utotal found by find_p.m
% Unew is Utotal without any 'conj' functions.
% We also find dUnew/dtheta and dUnew/dphi symbolically and save them
% for use in other code.

Unew=-(m*cos(theta))*(((-(m*cos(phi))*(cos(theta)-2*R-epsilon+cos(phi))+
(m*sin(phi))*(sin(theta)-sin(phi)))*(3*cos(theta)-6*R-3*epsilon+3*cos(phi))
+((cos(theta)-2*R-epsilon+cos(phi))*(cos(theta)-2*R-epsilon+cos(phi))+
(sin(theta)-sin(phi))*(sin(theta)-sin(phi)))*m*cos(phi))/((cos(theta)-
2*R-epsilon+cos(phi))*(cos(theta)-2*R-epsilon+cos(phi))+(sin(theta)-sin(phi))*
(sin(theta)-sin(phi)))^(5/2)+C*(((m*sin(phi+1/6*pi))*(cos(theta)-2*R-epsilon-
sin(phi+1/6*pi))+(m*cos(phi+1/6*pi))*(sin(theta)-cos(phi+1/6*pi)))*
(3*cos(theta)-6*R-3*epsilon-3*sin(phi+1/6*pi))-((cos(theta)-2*R-epsilon-
sin(phi+1/6*pi))*(cos(theta)-2*R-epsilon-sin(phi+1/6*pi))+(sin(theta)-
cos(phi+1/6*pi))*(sin(theta)-cos(phi+1/6*pi)))*m*sin(phi+1/6*pi))/
((cos(theta)-2*R-epsilon-sin(phi+1/6*pi))*(cos(theta)-2*R-epsilon-sin(phi+
1/6*pi))+(sin(theta)-cos(phi+1/6*pi))*(sin(theta)-cos(phi+1/6*pi)))^(5/2)+
C*(((m*cos(phi+1/3*pi))*(cos(theta)-2*R-epsilon-cos(phi+1/3*pi))-(m*
sin(phi+1/3*pi))*(sin(theta)+sin(phi+1/3*pi)))*(3*cos(theta)-6*R-3*epsilon-
3*cos(phi+1/3*pi))-((cos(theta)-2*R-epsilon-cos(phi+1/3*pi))*(cos(theta)-2*R
-epsilon-cos(phi+1/3*pi))+(sin(theta)+sin(phi+1/3*pi))*(sin(theta)+sin(phi+1/3
*pi)))*m*cos(phi+1/3*pi))/((cos(theta)-2*R-epsilon-cos(phi+1/3*pi))*
(cos(theta)-2*R-epsilon-cos(phi+1/3*pi))+(sin(theta)+sin(phi+1/3*pi))*
(sin(theta)+sin(phi+1/3*pi)))^(5/2))-(m*sin(theta))*(((-(m*cos(phi))*
(cos(theta)-2*R-epsilon+cos(phi))+(m*sin(phi))*(sin(theta)-sin(phi)))*
(3*sin(theta)-3*sin(phi))-((cos(theta)-2*R-epsilon+cos(phi))*(cos(theta)
-2*R-epsilon+cos(phi))+(sin(theta)-sin(phi))*(sin(theta)-sin(phi)))*m*
sin(phi))/((cos(theta)-2*R-epsilon+cos(phi))*(cos(theta)-2*R-epsilon+
```

```
cos(phi))+(sin(theta)-sin(phi))*(sin(theta)-sin(phi)))^(5/2)+C*(((m*
sin(phi+1/6*pi))*(cos(theta)-2*R-epsilon-sin(phi+1/6*pi))+(m*cos(phi+
1/6*pi))*(sin(theta)-cos(phi+1/6*pi)))*(3*sin(theta)-3*cos(phi+1/6*pi))-
((cos(theta)-2*R-epsilon-sin(phi+1/6*pi))*(cos(theta)-2*R-epsilon-sin(phi+
1/6*pi))+(sin(theta)-cos(phi+1/6*pi))*(sin(theta)-cos(phi+1/6*pi)))*m*
cos(phi+1/6*pi))/((cos(theta)-2*R-epsilon-sin(phi+1/6*pi))*(cos(theta)-2*
R-epsilon-sin(phi+1/6*pi))+(sin(theta)-cos(phi+1/6*pi))*(sin(theta)-
cos(phi+1/6*pi)))^(5/2)+C*(((m*cos(phi+1/3*pi))*(cos(theta)-2*R-epsilon-
cos(phi+1/3*pi))-(m*sin(phi+1/3*pi))*(sin(theta)+sin(phi+1/3*pi)))*(3*
sin(theta)+3*sin(phi+1/3*pi))+((cos(theta)-2*R-epsilon-cos(phi+1/3*pi))*
(cos(theta)-2*R-epsilon-cos(phi+1/3*pi))+(sin(theta)+sin(phi+1/3*pi))*
(sin(theta)+sin(phi+1/3*pi)))*m*sin(phi+1/3*pi))/((cos(theta)-2*R-epsilon
-cos(phi+1/3*pi))*(cos(theta)-2*R-epsilon-cos(phi+1/3*pi))+(sin(theta)+
sin(phi+1/3*pi))*(sin(theta)+sin(phi+1/3*pi)))^(5/2))+(m*sin(theta+1/6*
pi))*(C*((-(m*cos(phi))*(-sin(theta+1/6*pi)-2*R-epsilon+cos(phi))+(m*
sin(phi))*(cos(theta+1/6*pi)-sin(phi)))*(-3*sin(theta+1/6*pi)-6*R-3*epsilon+
3*cos(phi))+((-sin(theta+1/6*pi)-2*R-epsilon+cos(phi))*(-sin(theta+1/6*pi)-
2*R-epsilon+cos(phi))+(cos(theta+1/6*pi)-sin(phi))*(cos(theta+1/6*pi)-
sin(phi)))*m*cos(phi))/((-sin(theta+1/6*pi)-2*R-epsilon+cos(phi))*(-sin(theta+
1/6*pi)-2*R-epsilon+cos(phi))+(cos(theta+1/6*pi)-sin(phi))*(cos(theta+1/6*pi)-
sin(phi)))^(5/2)+C*(((m*sin(phi+1/6*pi))*(-sin(theta+1/6*pi)-2*R-epsilon-
sin(phi+1/6*pi))+(m*cos(phi+1/6*pi))*(cos(theta+1/6*pi)-cos(phi+1/6*pi)))*
(-3*sin(theta+1/6*pi)-6*R-3*epsilon-3*sin(phi+1/6*pi))-((-sin(theta+1/6*pi)-
2*R-epsilon-sin(phi+1/6*pi))*(-sin(theta+1/6*pi)-2*R-epsilon-sin(phi+1/6*pi))+
(cos(theta+1/6*pi)-cos(phi+1/6*pi))*(cos(theta+1/6*pi)-cos(phi+1/6*pi)))*m*
sin(phi+1/6*pi))/((-sin(theta+1/6*pi)-2*R-epsilon-sin(phi+1/6*pi))*(-sin(theta+
1/6*pi)-2*R-epsilon-sin(phi+1/6*pi))+(cos(theta+1/6*pi)-cos(phi+1/6*pi))*
(cos(theta+1/6*pi)-cos(phi+1/6*pi)))^(5/2)+C*(((m*cos(phi+1/3*pi))*(-sin(theta+
1/6*pi)-2*R-epsilon-cos(phi+1/3*pi))-(m*sin(phi+1/3*pi))*(cos(theta+1/6*pi)+
sin(phi+1/3*pi)))*(-3*sin(theta+1/6*pi)-6*R-3*epsilon-3*cos(phi+1/3*pi))-
((-sin(theta+1/6*pi)-2*R-epsilon-cos(phi+1/3*pi))*(-sin(theta+1/6*pi)-2*R-
epsilon-cos(phi+1/3*pi))+(cos(theta+1/6*pi)+sin(phi+1/3*pi))*(cos(theta+1/6*
pi)+sin(phi+1/3*pi)))*m*cos(phi+1/3*pi))/((-sin(theta+1/6*pi)-2*R-epsilon-
cos(phi+1/3*pi))*(-sin(theta+1/6*pi)-2*R-epsilon-cos(phi+1/3*pi))+(cos(theta+
1/6*pi)+sin(phi+1/3*pi))*(cos(theta+1/6*pi)+sin(phi+1/3*pi)))^(5/2))-(m*cos
(theta+1/6*pi))*(C*((-(m*cos(phi))*(-sin(theta+1/6*pi)-2*R-epsilon+cos(phi))+
(m*sin(phi))*(cos(theta+1/6*pi)-sin(phi)))*(3*cos(theta+1/6*pi)-3*sin(phi))-
((-sin(theta+1/6*pi)-2*R-epsilon+cos(phi))*(-sin(theta+1/6*pi)-2*R-epsilon+
cos(phi))+(cos(theta+1/6*pi)-sin(phi))*(cos(theta+1/6*pi)-sin(phi)))*m*
sin(phi))/((-sin(theta+1/6*pi)-2*R-epsilon+cos(phi))*(-sin(theta+1/6*pi)-
2*R-epsilon+cos(phi))+(cos(theta+1/6*pi)-sin(phi))*(cos(theta+1/6*pi)-
sin(phi)))^(5/2)+C*(((m*sin(phi+1/6*pi))*(-sin(theta+1/6*pi)-2*R-epsilon-
sin(phi+1/6*pi))+(m*cos(phi+1/6*pi))*(cos(theta+1/6*pi)-cos(phi+1/6*pi)))*
(3*cos(theta+1/6*pi)-3*cos(phi+1/6*pi))-((-sin(theta+1/6*pi)-2*R-epsilon-
sin(phi+1/6*pi))*(-sin(theta+1/6*pi)-2*R-epsilon-sin(phi+1/6*pi))+
(cos(theta+1/6*pi)-cos(phi+1/6*pi))*(cos(theta+1/6*pi)-cos(phi+1/6*pi)))*
m*cos(phi+1/6*pi))/((-sin(theta+1/6*pi)-2*R-epsilon-sin(phi+1/6*pi))*
(-sin(theta+1/6*pi)-2*R-epsilon-sin(phi+1/6*pi))+(cos(theta+1/6*pi)-
```

```
cos(phi+1/6*pi))*(cos(theta+1/6*pi)-cos(phi+1/6*pi)))^(5/2)+C*(((m*
cos(phi+1/3*pi))*(-sin(theta+1/6*pi)-2*R-epsilon-cos(phi+1/3*pi))-(m*
sin(phi+1/3*pi))*(cos(theta+1/6*pi)+sin(phi+1/3*pi)))*(3*cos(theta+1/6*pi)+
3*sin(phi+1/3*pi))+((-sin(theta+1/6*pi)-2*R-epsilon-cos(phi+1/3*pi))*
(-sin(theta+1/6*pi)-2*R-epsilon-cos(phi+1/3*pi))+(cos(theta+1/6*pi)+
sin(phi+1/3*pi))*(cos(theta+1/6*pi)+sin(phi+1/3*pi)))*m*sin(phi+1/3*pi))/
((-sin(theta+1/6*pi)-2*R-epsilon-cos(phi+1/3*pi))*(-sin(theta+1/6*pi)-
2*R-epsilon-cos(phi+1/3*pi))+(cos(theta+1/6*pi)+sin(phi+1/3*pi))*
(cos(theta+1/6*pi)+sin(phi+1/3*pi)))^(5/2))+(m*cos(theta+1/3*pi))*
(C*((-(m*cos(phi))*(-cos(theta+1/3*pi)-2*R-epsilon+cos(phi))+(m*sin(phi))*
(-sin(theta+1/3*pi)-sin(phi)))*(-3*cos(theta+1/3*pi)-6*R-3*epsilon+3*
cos(phi))+((-cos(theta+1/3*pi)-2*R-epsilon+cos(phi))*(-cos(theta+1/3*pi)-
2*R-epsilon+cos(phi))+(-sin(theta+1/3*pi)-sin(phi))*(-sin(theta+1/3*pi)-
sin(phi)))*m*cos(phi))/((-cos(theta+1/3*pi)-2*R-epsilon+cos(phi))*(-cos(theta+
1/3*pi)-2*R-epsilon+cos(phi))+(-sin(theta+1/3*pi)-sin(phi))*(-sin(theta+1/3*pi)
-sin(phi)))^(5/2)+C*(((m*sin(phi+1/6*pi))*(-cos(theta+1/3*pi)-2*R-epsilon-
sin(phi+1/6*pi))+(m*cos(phi+1/6*pi))*(-sin(theta+1/3*pi)-cos(phi+1/6*pi)))*
(-3*cos(theta+1/3*pi)-6*R-3*epsilon-3*sin(phi+1/6*pi))-((-cos(theta+1/3*pi)-
2*R-epsilon-sin(phi+1/6*pi))*(-cos(theta+1/3*pi)-2*R-epsilon-sin(phi+1/6*pi))+
(-sin(theta+1/3*pi)-cos(phi+1/6*pi))*(-sin(theta+1/3*pi)-cos(phi+1/6*pi)))*
m*sin(phi+1/6*pi))/((-cos(theta+1/3*pi)-2*R-epsilon-sin(phi+1/6*pi))*(-
cos(theta+1/3*pi)-2*R-epsilon-sin(phi+1/6*pi))+(-sin(theta+1/3*pi)-cos(phi+
1/6*pi))*(-sin(theta+1/3*pi)-cos(phi+1/6*pi)))^(5/2)+C*(((m*cos(phi+1/3*pi))*
(-cos(theta+1/3*pi)-2*R-epsilon-cos(phi+1/3*pi))-(m*sin(phi+1/3*pi))*(-
sin(theta+1/3*pi)+sin(phi+1/3*pi)))*(-3*cos(theta+1/3*pi)-6*R-3*epsilon-3*
cos(phi+1/3*pi))-((-cos(theta+1/3*pi)-2*R-epsilon-cos(phi+1/3*pi))*(-
cos(theta+1/3*pi)-2*R-epsilon-cos(phi+1/3*pi))+(-sin(theta+1/3*pi)+sin(phi+
1/3*pi))*(-sin(theta+1/3*pi)+sin(phi+1/3*pi)))*m*cos(phi+1/3*pi))/((-
cos(theta+1/3*pi)-2*R-epsilon-cos(phi+1/3*pi))*(-cos(theta+1/3*pi)-2*R-
epsilon-cos(phi+1/3*pi))+(-sin(theta+1/3*pi)+sin(phi+1/3*pi))*(-sin(theta+
1/3*pi)+sin(phi+1/3*pi)))^(5/2))+(m*sin(theta+1/3*pi))*(C*((-(m*cos(phi))*
(-cos(theta+1/3*pi)-2*R-epsilon+cos(phi))+(m*sin(phi))*(-sin(theta+1/3*pi)-
sin(phi)))*(-3*sin(theta+1/3*pi)-3*sin(phi))-((-cos(theta+1/3*pi)-2*R-
epsilon+cos(phi))*(-cos(theta+1/3*pi)-2*R-epsilon+cos(phi))+(-sin(theta+
1/3*pi)-sin(phi))*(-sin(theta+1/3*pi)-sin(phi)))*m*sin(phi))/((-cos(theta+
1/3*pi)-2*R-epsilon+cos(phi))*(-cos(theta+1/3*pi)-2*R-epsilon+cos(phi))+(-
sin(theta+1/3*pi)-sin(phi))*(-sin(theta+1/3*pi)-sin(phi)))^(5/2)+C*(((m*
sin(phi+1/6*pi))*(-cos(theta+1/3*pi)-2*R-epsilon-sin(phi+1/6*pi))+(m*
cos(phi+1/6*pi))*(-sin(theta+1/3*pi)-cos(phi+1/6*pi)))*(-3*sin(theta+1/3*pi)-
3*cos(phi+1/6*pi))-((-cos(theta+1/3*pi)-2*R-epsilon-sin(phi+1/6*pi))*(-
cos(theta+1/3*pi)-2*R-epsilon-sin(phi+1/6*pi))+(-sin(theta+1/3*pi)-cos(phi+
1/6*pi))*(-sin(theta+1/3*pi)-cos(phi+1/6*pi)))*m*cos(phi+1/6*pi))/((-
cos(theta+1/3*pi)-2*R-epsilon-sin(phi+1/6*pi))*(-cos(theta+1/3*pi)-2*R-
epsilon-sin(phi+1/6*pi))+(-sin(theta+1/3*pi)-cos(phi+1/6*pi))*(-sin(theta+
1/3*pi)-cos(phi+1/6*pi)))^(5/2)+C*(((m*cos(phi+1/3*pi))*(-cos(theta+1/3*pi)-
2*R-epsilon-cos(phi+1/3*pi))-(m*sin(phi+1/3*pi))*(-sin(theta+1/3*pi)+
sin(phi+1/3*pi)))*(-3*sin(theta+1/3*pi)+3*sin(phi+1/3*pi))+((-cos(theta+
1/3*pi)-2*R-epsilon-cos(phi+1/3*pi))*(-cos(theta+1/3*pi)-2*R-epsilon-
```

cos(phi+1/3*pi))+(-sin(theta+1/3*pi)+sin(phi+1/3*pi))*(-sin(theta+1/3*pi)+
sin(phi+1/3*pi)))*m*sin(phi+1/3*pi))/((-cos(theta+1/3*pi)-2*R-epsilon-
cos(phi+1/3*pi))*(-cos(theta+1/3*pi)-2*R-epsilon-cos(phi+1/3*pi))+(-sin(theta+
1/3*pi)+sin(phi+1/3*pi))*(-sin(theta+1/3*pi)+sin(phi+1/3*pi)))^(5/2))+(m*
cos(phi))*(C*(((m*cos(theta))*(2*R+epsilon-cos(phi)-cos(theta))+(m*sin(theta))*
(sin(phi)-sin(theta)))*(6*R+3*epsilon-3*cos(phi)-3*cos(theta))-((2*R+epsilon-
cos(phi)-cos(theta))*(2*R+epsilon-cos(phi)-cos(theta))+(sin(phi)-sin(theta))*
(sin(phi)-sin(theta)))*m*cos(theta))/((2*R+epsilon-cos(phi)-cos(theta))*(2*R+
epsilon-cos(phi)-cos(theta))+(sin(phi)-sin(theta))*(sin(phi)-sin(theta)))^
(5/2)+C*((-(m*sin(theta+1/6*pi))*(2*R+epsilon-cos(phi)+sin(theta+1/6*pi))+
(m*cos(theta+1/6*pi))*(sin(phi)-cos(theta+1/6*pi)))*(6*R+3*epsilon-3*cos(phi)+
3*sin(theta+1/6*pi))+((2*R+epsilon-cos(phi)+sin(theta+1/6*pi))*(2*R+epsilon-
cos(phi)+sin(theta+1/6*pi))+(sin(phi)-cos(theta+1/6*pi))*(sin(phi)-cos(theta+
1/6*pi)))*m*sin(theta+1/6*pi))/((2*R+epsilon-cos(phi)+sin(theta+1/6*pi))*
(2*R+epsilon-cos(phi)+sin(theta+1/6*pi))+(sin(phi)-cos(theta+1/6*pi))*
(sin(phi)-cos(theta+1/6*pi)))^(5/2)+C*((-(m*cos(theta+1/3*pi))*(2*R+epsilon-
cos(phi)+cos(theta+1/3*pi))-(m*sin(theta+1/3*pi))*(sin(phi)+sin(theta+
1/3*pi)))*(6*R+3*epsilon-3*cos(phi)+3*cos(theta+1/3*pi))+((2*R+epsilon-
cos(phi)+cos(theta+1/3*pi))*(2*R+epsilon-cos(phi)+cos(theta+1/3*pi))+
(sin(phi)+sin(theta+1/3*pi))*(sin(phi)+sin(theta+1/3*pi)))*m*cos(theta+1/3*
pi))/((2*R+epsilon-cos(phi)+cos(theta+1/3*pi))*(2*R+epsilon-cos(phi)+cos(theta+
1/3*pi))+(sin(phi)+sin(theta+1/3*pi))*(sin(phi)+sin(theta+1/3*pi)))^(5/2))-(m*
sin(phi))*(C*(((m*cos(theta))*(2*R+epsilon-cos(phi)-cos(theta))+(m*sin(theta))*
(sin(phi)-sin(theta)))*(3*sin(phi)-3*sin(theta))-((2*R+epsilon-cos(phi)-
cos(theta))*(2*R+epsilon-cos(phi)-cos(theta))+(sin(phi)-sin(theta))*(sin(phi)-
sin(theta)))*m*sin(theta))/((2*R+epsilon-cos(phi)-cos(theta))*(2*R+epsilon-
cos(phi)-cos(theta))+(sin(phi)-sin(theta))*(sin(phi)-sin(theta)))^(5/2)+C*
((-(m*sin(theta+1/6*pi))*(2*R+epsilon-cos(phi)+sin(theta+1/6*pi))+(m*cos(theta+
1/6*pi))*(sin(phi)-cos(theta+1/6*pi)))*(3*sin(phi)-3*cos(theta+1/6*pi))-((2*R+
epsilon-cos(phi)+sin(theta+1/6*pi))*(2*R+epsilon-cos(phi)+sin(theta+1/6*pi))+
(sin(phi)-cos(theta+1/6*pi))*(sin(phi)-cos(theta+1/6*pi)))*m*cos(theta+1/6*
pi))/((2*R+epsilon-cos(phi)+sin(theta+1/6*pi))*(2*R+epsilon-cos(phi)+sin(theta+
1/6*pi))+(sin(phi)-cos(theta+1/6*pi))*(sin(phi)-cos(theta+1/6*pi)))^(5/2)+
C*((-(m*cos(theta+1/3*pi))*(2*R+epsilon-cos(phi)+cos(theta+1/3*pi))-(m*
sin(theta+1/3*pi))*(sin(phi)+sin(theta+1/3*pi)))*(3*sin(phi)+3*sin(theta+1/3*
pi))+((2*R+epsilon-cos(phi)+cos(theta+1/3*pi))*(2*R+epsilon-cos(phi)+cos(theta+
1/3*pi))+(sin(phi)+sin(theta+1/3*pi))*(sin(phi)+sin(theta+1/3*pi)))*m*
sin(theta+1/3*pi))/((2*R+epsilon-cos(phi)+cos(theta+1/3*pi))*(2*R+epsilon-
cos(phi)+cos(theta+1/3*pi))+(sin(phi)+sin(theta+1/3*pi))*(sin(phi)+sin(theta+
1/3*pi)))^(5/2))-(m*sin(phi+1/6*pi))*(C*(((m*cos(theta))*(2*R+epsilon+sin(phi+
1/6*pi)-cos(theta))+(m*sin(theta))*(cos(phi+1/6*pi)-sin(theta)))*(6*R+3*
epsilon+3*sin(phi+1/6*pi)-3*cos(theta))-((2*R+epsilon+sin(phi+1/6*pi)-
cos(theta))*(2*R+epsilon+sin(phi+1/6*pi)-cos(theta))+(cos(phi+1/6*pi)-
sin(theta))*(cos(phi+1/6*pi)-sin(theta)))*m*cos(theta))/((2*R+epsilon+sin(phi+
1/6*pi)-cos(theta))*(2*R+epsilon+sin(phi+1/6*pi)-cos(theta))+(cos(phi+1/6*pi)-
sin(theta))*(cos(phi+1/6*pi)-sin(theta)))^(5/2)+C*((-(m*sin(theta+1/6*pi))*(2*
R+epsilon+sin(phi+1/6*pi)+sin(theta+1/6*pi))+(m*cos(theta+1/6*pi))*(cos(phi+
1/6*pi)-cos(theta+1/6*pi)))*(6*R+3*epsilon+3*sin(phi+1/6*pi)+3*sin(theta+1/6*

```
pi))+((2*R+epsilon+sin(phi+1/6*pi)+sin(theta+1/6*pi))*(2*R+epsilon+sin(phi+1/6*
pi)+sin(theta+1/6*pi))+(cos(phi+1/6*pi)-cos(theta+1/6*pi))*(cos(phi+1/6*pi)-
cos(theta+1/6*pi)))*m*sin(theta+1/6*pi))/((2*R+epsilon+sin(phi+1/6*pi)+
sin(theta+1/6*pi))*(2*R+epsilon+sin(phi+1/6*pi)+sin(theta+1/6*pi))+(cos(phi+
1/6*pi)-cos(theta+1/6*pi))*(cos(phi+1/6*pi)-cos(theta+1/6*pi)))^(5/2)+C*((-(m*
cos(theta+1/3*pi))*(2*R+epsilon+sin(phi+1/6*pi)+cos(theta+1/3*pi))-(m*
sin(theta+1/3*pi))*(cos(phi+1/6*pi)+sin(theta+1/3*pi)))*(6*R+3*epsilon+3*
sin(phi+1/6*pi)+3*cos(theta+1/3*pi))+((2*R+epsilon+sin(phi+1/6*pi)+cos(theta+
1/3*pi))*(2*R+epsilon+sin(phi+1/6*pi)+cos(theta+1/3*pi))+(cos(phi+1/6*pi)+
sin(theta+1/3*pi))*(cos(phi+1/6*pi)+sin(theta+1/3*pi)))*m*cos(theta+1/3*pi))/
((2*R+epsilon+sin(phi+1/6*pi)+cos(theta+1/3*pi))*(2*R+epsilon+sin(phi+1/6*pi)+
cos(theta+1/3*pi))+(cos(phi+1/6*pi)+sin(theta+1/3*pi))*(cos(phi+1/6*pi)+
sin(theta+1/3*pi)))^(5/2))-(m*cos(phi+1/6*pi))*(C*(((m*cos(theta))*(2*R+
epsilon+sin(phi+1/6*pi)-cos(theta))+(m*sin(theta))*(cos(phi+1/6*pi)-
sin(theta)))*(3*cos(phi+1/6*pi)-3*sin(theta))-((2*R+epsilon+sin(phi+1/6*pi)-
cos(theta))*(2*R+epsilon+sin(phi+1/6*pi)-cos(theta))+(cos(phi+1/6*pi)-
sin(theta))*(cos(phi+1/6*pi)-sin(theta)))*m*sin(theta))/((2*R+epsilon+sin(phi+
1/6*pi)-cos(theta))*(2*R+epsilon+sin(phi+1/6*pi)-cos(theta))+(cos(phi+1/6*pi)-
sin(theta))*(cos(phi+1/6*pi)-sin(theta)))^(5/2)+C*((-(m*sin(theta+1/6*pi))*
(2*R+epsilon+sin(phi+1/6*pi)+sin(theta+1/6*pi))+(m*cos(theta+1/6*pi))*(cos(phi+
1/6*pi)-cos(theta+1/6*pi)))*(3*cos(phi+1/6*pi)-3*cos(theta+1/6*pi))-((2*R+
epsilon+sin(phi+1/6*pi)+sin(theta+1/6*pi))*(2*R+epsilon+sin(phi+1/6*pi)+
sin(theta+1/6*pi))+(cos(phi+1/6*pi)-cos(theta+1/6*pi))*(cos(phi+1/6*pi)-
cos(theta+1/6*pi)))*m*cos(theta+1/6*pi))/((2*R+epsilon+sin(phi+1/6*pi)+
sin(theta+1/6*pi))*(2*R+epsilon+sin(phi+1/6*pi)+sin(theta+1/6*pi))+(cos(phi+
1/6*pi)-cos(theta+1/6*pi))*(cos(phi+1/6*pi)-cos(theta+1/6*pi)))^(5/2)+C*((-(m*
cos(theta+1/3*pi))*(2*R+epsilon+sin(phi+1/6*pi)+cos(theta+1/3*pi))-(m*
sin(theta+1/3*pi))*(cos(phi+1/6*pi)+sin(theta+1/3*pi)))*(3*cos(phi+1/6*pi)+3*
sin(theta+1/3*pi))+((2*R+epsilon+sin(phi+1/6*pi)+cos(theta+1/3*pi))*(2*R+
epsilon+sin(phi+1/6*pi)+cos(theta+1/3*pi))+(cos(phi+1/6*pi)+sin(theta+1/3*pi))*
(cos(phi+1/6*pi)+sin(theta+1/3*pi)))*m*sin(theta+1/3*pi))/((2*R+epsilon+
sin(phi+1/6*pi)+cos(theta+1/3*pi))*(2*R+epsilon+sin(phi+1/6*pi)+cos(theta+1/3*
pi))+(cos(phi+1/6*pi)+sin(theta+1/3*pi))*(cos(phi+1/6*pi)+sin(theta+1/3*pi)))^
(5/2))-(m*cos(phi+1/3*pi))*(C*(((m*cos(theta))*(2*R+epsilon+cos(phi+1/3*pi)-
cos(theta))+(m*sin(theta))*(-sin(phi+1/3*pi)-sin(theta)))*(6*R+3*epsilon+3*
cos(phi+1/3*pi)-3*cos(theta))-((2*R+epsilon+cos(phi+1/3*pi)-cos(theta))*(2*R+
epsilon+cos(phi+1/3*pi)-cos(theta))+(-sin(phi+1/3*pi)-sin(theta))*(-sin(phi+
1/3*pi)-sin(theta)))*m*cos(theta))/((2*R+epsilon+cos(phi+1/3*pi)-cos(theta))*
(2*R+epsilon+cos(phi+1/3*pi)-cos(theta))+(-sin(phi+1/3*pi)-sin(theta))*(-
sin(phi+1/3*pi)-sin(theta)))^(5/2)+C*((-(m*sin(theta+1/6*pi))*(2*R+epsilon+
cos(phi+1/3*pi)+sin(theta+1/6*pi))+(m*cos(theta+1/6*pi))*(-sin(phi+1/3*pi)-
cos(theta+1/6*pi)))*(6*R+3*epsilon+3*cos(phi+1/3*pi)+3*sin(theta+1/6*pi))+
((2*R+epsilon+cos(phi+1/3*pi)+sin(theta+1/6*pi))*(2*R+epsilon+cos(phi+1/3*pi)+
sin(theta+1/6*pi))+(-sin(phi+1/3*pi)-cos(theta+1/6*pi))*(-sin(phi+1/3*pi)-
cos(theta+1/6*pi)))*m*sin(theta+1/6*pi))/((2*R+epsilon+cos(phi+1/3*pi)+
sin(theta+1/6*pi))*(2*R+epsilon+cos(phi+1/3*pi)+sin(theta+1/6*pi))+(-sin(phi+
1/3*pi)-cos(theta+1/6*pi))*(-sin(phi+1/3*pi)-cos(theta+1/6*pi)))^(5/2)+C*((-
(m*cos(theta+1/3*pi))*(2*R+epsilon+cos(phi+1/3*pi)+cos(theta+1/3*pi))-(m*
```

```
sin(theta+1/3*pi))*(-sin(phi+1/3*pi)+sin(theta+1/3*pi)))*(6*R+3*epsilon+3*
cos(phi+1/3*pi)+3*cos(theta+1/3*pi))+((2*R+epsilon+cos(phi+1/3*pi)+cos(theta+
1/3*pi))*(2*R+epsilon+cos(phi+1/3*pi)+cos(theta+1/3*pi))+(-sin(phi+1/3*pi)+
sin(theta+1/3*pi))*(-sin(phi+1/3*pi)+sin(theta+1/3*pi)))*m*cos(theta+1/3*pi))/
((2*R+epsilon+cos(phi+1/3*pi)+cos(theta+1/3*pi))*(2*R+epsilon+cos(phi+1/3*pi)+
cos(theta+1/3*pi))+(-sin(phi+1/3*pi)+sin(theta+1/3*pi))*(-sin(phi+1/3*pi)+
sin(theta+1/3*pi)))^(5/2))+(m*sin(phi+1/3*pi))*(C*(((m*cos(theta))*(2*R+
epsilon+cos(phi+1/3*pi)-cos(theta))+(m*sin(theta))*(-sin(phi+1/3*pi)-
sin(theta)))*(-3*sin(phi+1/3*pi)-3*sin(theta))-((2*R+epsilon+cos(phi+1/3*pi)-
cos(theta))*(2*R+epsilon+cos(phi+1/3*pi)-cos(theta))+(-sin(phi+1/3*pi)-
sin(theta))*(-sin(phi+1/3*pi)-sin(theta)))*m*sin(theta))/((2*R+epsilon+cos(phi+
1/3*pi)-cos(theta))*(2*R+epsilon+cos(phi+1/3*pi)-cos(theta))+(-sin(phi+1/3*pi)-
sin(theta))*(-sin(phi+1/3*pi)-sin(theta)))^(5/2)+C*((-(m*sin(theta+1/6*pi))*
(2*R+epsilon+cos(phi+1/3*pi)+sin(theta+1/6*pi))+(m*cos(theta+1/6*pi))*
(-sin(phi+1/3*pi)-cos(theta+1/6*pi)))*(-3*sin(phi+1/3*pi)-3*cos(theta+1/6*pi))-
((2*R+epsilon+cos(phi+1/3*pi)+sin(theta+1/6*pi))*(2*R+epsilon+cos(phi+1/3*pi)+
sin(theta+1/6*pi))+(-sin(phi+1/3*pi)-cos(theta+1/6*pi))*(-sin(phi+1/3*pi)-
cos(theta+1/6*pi)))*m*cos(theta+1/6*pi))/((2*R+epsilon+cos(phi+1/3*pi)+
sin(theta+1/6*pi))*(2*R+epsilon+cos(phi+1/3*pi)+sin(theta+1/6*pi))+(-sin(phi+
1/3*pi)-cos(theta+1/6*pi))*(-sin(phi+1/3*pi)-cos(theta+1/6*pi)))^(5/2)+C*
((-(m*cos(theta+1/3*pi))*(2*R+epsilon+cos(phi+1/3*pi)+cos(theta+1/3*pi))-
(m*sin(theta+1/3*pi))*(-sin(phi+1/3*pi)+sin(theta+1/3*pi)))*(-3*sin(phi+
1/3*pi)+3*sin(theta+1/3*pi))+((2*R+epsilon+cos(phi+1/3*pi)+cos(theta+1/3*pi))*
(2*R+epsilon+cos(phi+1/3*pi)+cos(theta+1/3*pi))+(-sin(phi+1/3*pi)+sin(theta+
1/3*pi))*(-sin(phi+1/3*pi)+sin(theta+1/3*pi)))*m*sin(theta+1/3*pi))/((2*R+
epsilon+cos(phi+1/3*pi)+cos(theta+1/3*pi))*(2*R+epsilon+cos(phi+1/3*pi)+
cos(theta+1/3*pi))+(-sin(phi+1/3*pi)+sin(theta+1/3*pi))*(-sin(phi+1/3*pi)+
sin(theta+1/3*pi)))^(5/2))

dUdTheta=diff(Unew,'theta');
dUdPhi=diff(Unew,'phi');
save Unew;
save dUdTheta;
save dUdPhi;
```

# Appendix C

```
% graph_p.m
% MATLAB code for Magnetron Project - Chris DuBois
% Script for plotting potentials

load Utotal;

%%% CONSTANTS
R=1;
epsilon=.4;
M=2;
```

```
m=3;
I=3*M*R^2;
beta1=.2;
beta2=.3;
C=1; % C=\mu_0/4pi

% Step size
d=2*pi/3/500;

% Potential Energy Plot for
% theta=(0,2pi/3), phi=(0,2pi/3)
P=0;
i=1;
for theta=0:d:2*pi/3
    j=1;
    for phi=0:d:2*pi/3
        P(i,j)=eval(U);
        j=j+1;
    end
    i=i+1;
end

% Potential Energy Plot for
% Theta Fixed, Phi (0,2pi/3)

theta=0;
i=1;
for theta=0:pi/6:pi/3
    j=1;
    for phi=0:d:2*pi/3
        P(i,j)=eval(Utotal);
        j=j+1;
    end
    i=i+1;
end
hold off;
plot(P(1,:)); hold on;
plot(P(2,:));
plot(P(3,:));
```

# Appendix D

```
% main.m
% MATLAB code for Magnetron Project - Chris DuBois
% Solve the ODEs using rk4

global dUdTheta dUdPhi;
```

```
load dUdTheta;
load dUdPhi;

hold off; clf;
[t, y] = rk4('projectFun', [0 20], [-pi/6;0;pi/3;30],.005);
save run036 y;
```

## Appendix E

```
% projectFun.m
% MATLAB code for Magnetron Project - Chris DuBois
% Provides the ODEs to solve

function Yprime = projectFun(t,y)

%%% CONSTANTS
R=1;
epsilon=.4;
M=2;
m=3;
I=3*M*R^2;
beta1=.2;
beta2=.3;
C=1; % C=\mu_0/4pi

Theta    = y(1);
ThetaDot = y(2);
Phi      = y(3);
PhiDot   = y(4);

theta=mod(Theta,2*pi);
phi=mod(Phi,2*pi);

global dUdTheta dUdPhi;

dUdT=eval(dUdTheta);
dUdP=eval(dUdPhi);

ThetaDotDot=-dUdT/I;
PhiDotDot=-dUdP/I;

Yprime=[ThetaDot;
        ThetaDotDot-beta1*ThetaDot;
        PhiDot;
        PhiDotDot-beta2*PhiDot];

% % Uncomment for fixed theta
```

```
% Yprime=[Theta;
%         ThetaDot;
%         PhiDot;
%         PhiDotDot-beta2*PhiDot];
```