**Advanced Dynamics Class Project**

**Matt Earl**

**Due: 12/6/2021**

**Introduction:**  Watching objects spin unstably about their intermediate axis is quite fascinating. This chaotic tumbling is known as the Dzhanibekov effect. The description of this motion is likewise fascinating. In this report I will examine this phenomenon by deriving Euler’s equations, applying these equations to a dynamics textbook, solving these equations numerically using Simulink, and presenting an analysis of the results. I have also included a detailed analytical solution to the problem and a link to a fun Python application which animates this phenomenon. To start let us begin by deriving Euler’s equations.

**Derivation:**

Start from Newton’s Second Law:

Note the derivative operator is absolute with respect to an inertial frame. Recall that when we consider a frame rotating with an angular velocity , this absolute derivative operator can be expressed in (2).

Where denotes the time derivative with respect to the rotating frame. Now from (2) we conclude:

Substituting this expression (3) into (1):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Notice how the local derivative operator does not act on or directions as these change with the rotation of the frame. Although these change with respect to the inertial frame, they do not change within the rotating frame so only the change in magnitude of the angular momentum, not its direction, is relevant in this description. We can expand the cross product to get clearer expressions in (4).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Notice that (4’) is equation 7/19 in the textbook which is the most general form of these equations. To reduce this to Euler’s equations we make three assumptions (requirements) for the motion: first this describes a rigid body, second the rotating frame is attached to the body, secondly this rotating frame is aligned with principal directions of inertia of the rigid body.

First, we note that we are considering the motion of a rigid body rotating with an angular velocity . In general, angular momentum is defined by (5).

Under this rigid body assumption, we note that the inertia tensor is constant hence:

Second, we note that the rotating frame is attached to the body so naturally . This assumption allows us to write (4’) as:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

The third assumption, that the principal directions of inertia are aligned with the rotating base vectors and , allows us to write (5) and (6) in terms of the principal inertias , , and .

Substituting these expressions into (7) we arrive at Euler’s equations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Addendum: we can arrive at this result quickly if we rely on the indicial notation. Starting with the indicial form of (4).

Apply the first assumption and

Apply the second assumption

Apply the third assumption

The expression (12) is in fact equivalent to (8).

**Applying Euler’s Equation to a Spinning Dynamics Textbook:**

Consider the dynamics textbook to be a rigid body freely spinning. If we attach a set of axes at the mass center aligned with the principal directions of inertia, the free body diagram is as shown in Figure 1.

Diagram

Description automatically generated

***Figure 1:*** *A free body diagram of a spinning book.*

Notice how the force of gravity acts through the mass center of the book. This means there are no moments acting on the book hence . Notice also the inertial frame and the rotating frame . The frame (which is attached to the book) rotates with angular velocity (not shown in Figure 1). Lastly, we denote the principal inertias of the book as , , and . Under this setup we can apply Euler’s equations (8) with the problem specifics yielding the three equations of motion below (12).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | (13.1) |
|  |  |  |  | (13.2) |
|  |  |  |  | (13.3) |

Given the dimensions shown above and assuming the mass of the book to be kg, I can calculate the values of inertia , and . A sample calculation for is shown in (14)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | (14) |
|  |  |  |  |  |
|  |  |  |  |  |

The principal moments of inertia for the book are calculated to be.

Note: The initial angular velocity of the book is denoted by and has not been defined yet.

Note: The inertias were intentionally named to preserve the ordering .

**Numerical Solution:**

For simplicity we define the constants , and as:

From the ordering , it should be noted that is negative while and are both positive. With these constants defined the equations (13) can be written as (14):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | (14.1) |
|  |  |  |  | (14.2) |
|  |  |  |  | (14.3) |

This is a system of coupled nonlinear differential equations. We can solve this problem numerically by using Simulink. The model of these equations is shown in Figure 2 below with the paths labelled.

Diagram

Description automatically generated

***Figure 2:*** *Simulink Model of the system of equations (14).*

A simple MATLAB script is enough to define the parameters and initial conditions of the integrator blocks. This MATLAB script is shown below in Figure 3.

Text

Description automatically generated

***Figure 3:*** *MATLAB script which defines Simulink system parameters.*

The magic of the Dzhanibekov effect really comes when we consider the stability of the system for various initial conditions. For this reason, I have run the system under three initial conditions. The first condition corresponds to an angular velocity primarily about the 1-axis, the second primarily about the 2-axis, and lastly the third primarily about the 3-axis. These initial conditions used are written below for clarity.

A slight perturbation is included on the other axes for a touch of realism. One cannot perfectly spin a body like a book about one axis without parting any angular velocity to the others. These perturbations are also added to really demonstrate the stability (or lack thereof) of this motion.

The Simulink results for the angular velocity calculated from each of these initial conditions is shown below in Figure 4. I have allowed the model to integrate the equations for a period of 14 seconds.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

***Figure 4:*** *Angular velocity components plotted as functions of time for each initial condition: The topmost plot for , the middle plot for , and the bottommost plot for*

Figure 4 is quite fascinating. Clearly when the rotation is about the 1-axis (the maximum inertia) and the 3-axis (the minimum inertia) the motion is stable. Slight oscillations resulting from the perturbations can be see about the other axes. Some very strange behavior is seen when rotating about the 2-axis (the intermediate inertia). The rate of rotation about the intermediate axis remains stable at for a little while before quite rapidly dropping until it is rotating backwards around . This oscillation of persists throughout the cycle. The other angular velocities and remain primarily at their small, perturbed values but spike up and come back down when flips.

This phenomenon is usually summed up by saying that rotations about the intermediate axes are unstable. This is not the most accurate description in my opinion. Rotation about the intermediate axis is an oscillation about two marginally stable states. At the beginning the book is in a marginally stable state. The perturbations about the other axes will grow in magnitude until the book flips. When this occurs, the perturbations shrink in size reaching the next marginally stable state before growing again.

It is hard for me to clearly state why this phenomenon occurs, but I have some pieces of understanding. As there are no moments acting on this body, the angular momentum and kinetic energy of the body are conserved. This means we have some constraints on the possible values of the angular velocity. I will explore these constraints below.

This conservation of energy tells us that the kinetic energy is equal to the initial kinetic energy.

The conservation of angular momentum likewise:

If we expand the kinetic energy equation above, we get:

Similarly expanding the angular momentum equation, we get:

Both equations (15) and (16) are the equations of ellipsoids in 3D space. Any valid solution to (14) must exist at the intersections of these curves. Using the data for this problem I have plotted these angular velocity ellipsoids for rotation about the stable axis (1-axis and 3-axis) in Figure 5 and the ellipsoids for rotation about the unstable axis in Figure 6.

Diagram

Description automatically generatedA picture containing text, accessory, umbrella

Description automatically generated

***Figure 5:*** *Representative angular velocity ellipsoids for stable rotations. Initial condition (left) shows the energy ellipsoid (blue) contained in the angular momentum ellipsoid (red) except for very small intersections near the ends. This intersection is shown in detail on the right for the case of initial condition* *Note: the angular momentum ellipsoid is now contained in the energy ellipsoid.*

*A picture containing text, sky, line

Description automatically generated*

***Figure 6:*** *Intersection of angular velocity ellipsoids for rotation about the intermediate axis corresponding to initial condition . Note: the intersection path wraps between the marginally stable points of .*

Any real solution of (14) must be some time parametrized journey along these intersection curves. The initial conditions corresponding to unstable rotations must have this separatrix feature in Figure 6 to wobble between marginally states.

**Angular Displacement:**

**Method 1:** Directly integrating velocity

With the angular velocity calculated, we can integrate the velocity to get the orientation angles of the book , and . Here we assume the initial orientation angles of the book are .

To do this numerically we simply put the angular velocity outputs of Simulink through another integrator as shown in Figure 7.

Diagram, schematic

Description automatically generated

***Figure 7****: Extension to Simulink model to calculate the orientation angles.*

With this slight addition, we get the time plots of angular displacement angles for each initial condition as shown in Figure 8.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

***Figure 8:*** *Angular displacement angles plotted as functions of time for each initial condition: The topmost plot for , the middle plot for , and the bottommost plot for*

The displacement angles for stable rotations (1-axis and 3-axis) grow very unsurprisingly. The rotation about the intermediate axis (2-axis) is a bit more interesting with some oscillation from and while the other angle steps up incrementally over time. This is not entirely surprising as the angular velocity never goes negative as seen in Figure 4 above.

**Method 2:** Non-commutative integration

Here is where I am confused. I really don’t think that anything in method 1 is correct at all even though it is what we were shown and told to do. You cannot integrate the angular velocity vector to get the orientation. This is because rotations are not commutative. Similarly, rotation matrices are not commutative. If we define a 1-rotation by , a 2-rotation by  and a 3-rotation by as shown below.

A compound rotation  will produce an orientation totally different than say . This comes from the non-commutativity of matrix multiplication. Once again, we cannot integrate angular velocity to get the orientation because the orientation is not a scalar or a vector but a matrix.

Note: For planar rotation (about only one axis) we CAN integrate with respect to time to get the orientation . We can do this because of the definition of angular velocity (18) and because there are no other components with which to commute.

We do not have quite as clear of a definition for the angular velocity vector in terms of a rotation matrix. We must walk more carefully around this problem. A fascinating relationship between the angular velocity and a rotation matrix is shown in (19). I found this resource here [1] that explains how to derive (19).

Where the angular velocity matrix is defined by:

We thus can rewrite (19) as a homogeneous MATRIX differential equation!

We are now working with matrices so everything is substantially more complex and our normal solution techniques have to be adjusted. The equation (20) has a solution in terms of ordered exponentials as discussed here [2] and here [3]. The solution is shown below and is really beyond my level of understanding right now.

A similar approach can be taken using quaternions too but that is REALLY beyond my level of understanding (this is discussed here [4] and here [5]).

**An Analytical Solution to (14)**

Disregarding the confusion of calculating the angular displacement we can use the constraints of equations (15) and (16) to solve equations (14) analytically. First, use elimination and consider To get expressed in terms of .

If we consider we can write in terms of .

With these expressions for in (20) and in (21), we can uncouple (14.2).

Where I have introduced the convenience notation:

The differential equation (22) is in fact separable!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  | (23) |

We can now integrate the left-hand side. We will get three very different answers depending on the ordering of and . We need to consider three cases i) , ii) , and iii) .

Case i):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  | (24) |

Substituting this into (23) yields the solution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  | (25.2) |

Substituting this expression into yields:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | (25.3) |

Similarly, plugging (25.2) into (21) yields the solution

Collecting these pieces we arrive at the solution of (14) under the condition .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | (25.1) |
|  |  |  |  | (25.2) |
|  |  |  |  | (25.3) |

Case ii):

Under this case the integral is much more problematic, and we must make use of the inverse Jacobi elliptic functions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | (26) |

Substituting (26) into (23) yields

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  | (27.2) |

By substituting this expression (27.2) back into (20) and (21) while making use of some of the properties of the Jacobi elliptic functions namely (and )

We arrive at the solutions (27) for case ii).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | (27.1) |
|  |  |  |  | (27.2) |
|  |  |  |  | (27.3) |

Without going into detail, the solution for case iii) is shown in (28)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | (28.1) |
|  |  |  |  | (28.2) |
|  |  |  |  | (28.3) |

There are a lot of helpful resources to understand what the Jacobi Elliptic functions are I learned about them here [6] and here [7]. Also, I would like to note that I did not come up with this solution strategy I learned about that here [8].

Results for each initial condition are shown below in Figure 9. The MATLAB code used to generate these figures is attached in the appendix.

Chart

Description automatically generatedDiagram

Description automatically generatedChart

Description automatically generated

***Figure 9:*** *Analytic Solutions of (14). Notice that apart from a sign change these are identical to the results calculated numerically in Figure 4.*

**Concluding Remarks:**

In this report I have successfully derived Euler’s equations and applied them to a spinning book. I also presented a numerical and analytical solution to the problem of angular velocity. I am not confident that I have calculated or even know how to calculate the orientation angles properly. Hopefully my discussion of the Dzhanibekov effect has been satisfactory and enjoyable.

Note: If you want to check out the Python application find it on my Github (<https://github.com/mearlearl>)

**References**

[1] <https://shiyuzhao.wordpress.com/2011/06/08/rotation-matrix-angle-axis-angular-velocity/>

[2] <https://en.wikipedia.org/wiki/Ordered_exponential>

[3] <https://en.wikipedia.org/wiki/Product_integral>

[4] <https://arxiv.org/pdf/1604.08139.pdf>

[5] <https://link.springer.com/book/10.1007/978-1-4471-7509-4>

[6] <https://mathworld.wolfram.com/JacobiEllipticFunctions.html>

[7] *Modern Analysis* E.T. Whittaker and G.N. Watson Cambridge University Press; 4th edition

[8] <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1053.3621&rep=rep1&type=pdf>

**Appendix**

%%%% JACOBI FUNCTION Solution %%%%

clear;

clc;

% Define parameters:

I1 = 0.01203;

I2 = 0.0074;

I3 = 0.0048;

Itensor = [I1 0 0; 0 I2 0 ; 0 0 I3];

omegavec = [0.01 0.01 10 ]';

Hvec = Itensor\*omegavec;

H0 = norm(Hvec);

T0 = 0.5\*dot(omegavec,Hvec);

lambda1 = (I2-I3)/I1;

lambda2 = (I3-I1)/I2;

lambda3 = (I1-I2)/I3;

a = sqrt((H0^2-2\*T0\*I1)/(I2\*(I2-I1)));

b = sqrt((H0^2-2\*T0\*I3)/(I2\*(I2-I3)));

L = sqrt(lambda3\*lambda1);

t = linspace(0,14,1000);

if a < b

k = a/b;

w2 = a\*jacobiSN(b\*L\*t,k);

w3 = a\*sqrt(-lambda3/lambda2)\*jacobiCN(b\*L\*t,k);

w1 = b\*sqrt(-lambda1/lambda2)\*jacobiDN(b\*L\*t,k);

elseif a > b

k = b/a;

w2 = b\*jacobiSN(a\*L\*t,k);

w3 = a\*sqrt(-lambda3/lambda2)\*jacobiDN(a\*L\*t,k);

w1 = b\*sqrt(-lambda1/lambda2)\*jacobiCN(a\*L\*t,k);

else

w1 = a\*sqrt(-lambda1/lambda2)\*sech(a\*L\*t);

w2 = a\*tanh(a\*L\*t);

w3 = a\*sqrt(-lambda3/lambda1)\*sech(a\*L\*t);

end

plot(t,w1,t,w2,t,w3)