CAT FLIPPING

ARDEN, SEAN, AND WILL

1. Moment of Inertia of the Cat

We model a falling cat with two cylinders A and B connected by a spherical joint (which we assume to be massless) with A and B separated by angle θ.

Claim 1. About the center of mass, the y axis of the cat is principle. Further, we can write $\vec{L} = (2I_{yy})\omega_y \hat{y}$. For an I_{yy} dependent on the geometry of the cylinders and θ.

Proof. Let the moment of inertia tensors of cylinders A and B about the center of mass of the cat as a whole be denoted I_A and I_B. Recall the definition of the moment of inertia tensor:

$$I = \begin{pmatrix} \sum(y^2+z^2) & \sum xy & \sum xz \\ \sum yx & \sum(x^2+z^2) & \sum yz \\ \sum zx & \sum zy & \sum(x^2+y^2) \end{pmatrix}$$

Then, by the reflectional symmetry of bodies A and B, the calculations for I_A and I_B are identical except for the replacement of y by $-y$. So, we can safely write

$$I_A = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \text{ and } I_B = \begin{pmatrix} I_{xx} & -I_{xy} & I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ I_{zx} & -I_{zy} & I_{zz} \end{pmatrix},$$

And so the moment of inertia tensor I of the cat as a whole is of the form

$$I = I_A + I_B = \begin{pmatrix} 2I_{xx} & 0 & 2I_{xz} \\ 0 & 2I_{yy} & 0 \\ 2I_{zx} & 0 & 2I_{zz} \end{pmatrix}$$

Then $\vec{y}$ is a clear eigenvector of I confirming that the y axis is a principle axis. □

Note that the above argument can be generalized to any two bodies with reflectional symmetry through the x, y plane.

Claim 2. The I_{yy} in claim 1 can be written $I_{y'y'}\cos^2(\theta/2) + I_{z'z'}\sin^2(\theta/2)$ where $I_{y'y'}$ and $I_{z'z'}$ are the moment of inertia elements corresponding to the principle axes.

Proof. Let x', y', z' denote the principle axes of cylinder A about the center of mass with z' going down the length of the cylinder, x' parallel to x, and y' in the corresponding location for a right-handed orthogonal coordinate system. Then, with this coordinate system corresponding to the principle axes, we can write the moment of inertia tenor I'_A of cylinder A about the center of mass of A.

$$I_A = \begin{pmatrix} I_{x'x'} & 0 & 0 \\ 0 & I_{y'y'} & 0 \\ 0 & 0 & I_{z'z'} \end{pmatrix}$$

Now, we adjust to the x, y, z coordinate system established above and solve for I_{yy}. Note $\hat{y}$ is given by $(0, \cos(\theta/2), \sin(\theta/2))$ in x', y', z' coordinates. Thus by adjusting coordinates we have

$$\begin{aligned} I_{yy} &= \begin{pmatrix} 0 & \cos(\theta/2) & \sin(\theta/2) \end{pmatrix} \begin{pmatrix} I_{x'x'} & 0 & 0 \\ 0 & I_{y'y'} & 0 \\ 0 & 0 & I_{z'z'} \end{pmatrix} \begin{pmatrix} 0 \\ \cos(\theta/2) \\ \sin(\theta/2) \end{pmatrix} \\ &= I_{y'y'} \cos^2(\theta/2) + I_{z'z'} \sin^2(\theta/2) \end{aligned}$$

The above gives the value of I_{yy} about the center of mass of A, but the center of mass of the cat as a whole shares identical x and z values. So, by $I_{yy} = \sum x^2 + z^2$, the value of I_{yy} holds about the center of mass the cat.

□

Again, note the above argument is generalizable given the principle axes.

Then overall we have a simple relationship between the angular momentum of the cat and an angular velocity ω of the cat in the $\vec{y}$ direction.

$$\vec{L} = (I_{y'y'} \cos^2(\theta/2) + I_{z'z'} \sin^2(\theta/2))\omega\hat{y} \tag{1}$$

In particular, if A and B are two solid cylinders of equal density with length L and radius R, we have

$$\vec{L} = (/ * */) \tag{2}$$

2. ????

3. Simulation

To verify our computations, we constructed a software simulation, that would be easily configurable, to allow rapid development. For the simulation we chose to use the `Python` programming language, and to make use of the `pyBullet` physics simulator. `pyBullet` is provided using python's integrated package manager `pip`.

3.1. URDF File Format.

`pyBullet` makes use of the `URDF` file format, so all models in the simulation must be loaded from a urdf file. This format has some specific constraints that are explained here, and the cat model that we used is included.

3.2. Simulation Setup.

`pyBullet` provides all of the physics simulations that are necessary for most common applications, but first it must be initialized in the code. To do this one must use the following commands before any physics simulation is possible.

```
import pybullet as p
physicsClient = p.connect(p.GUI)
p.setAdditionalSearchPath(pybullet_data.getDataPath())
```

These commands initialize the graphical interface for the simulation, and includes a path for some standard `pyBullet` models.

The next step is to set global constants that are persistent for the entirety of the simulation. In this case we will only set the gravitational constant.

```
p.setGravity(0,0,-9.8)
```

This function sets the gravitational force in the x, y, and z directions.

The final step before beginning the simulation is to load any models that will be used in the simulation. For our purposes, we load the standard `plane.urdf`, and we load our constructed `cat.urdf`.

```
planeId = p.loadURDF("plane.urdf")
startPos = [0,0,10]
startOrientation = p.getQuaternionFromEuler([0,0,0])
boxId = p.loadURDF("cat.urdf",startPos, startOrientation)
```

These functions load the plane model positioned at the origin, and loads the cat model positioned at $(0, 0, 10)$, with the default orientation.