Lab 10: Moments of Inertia

Jonah Spector, Alec Gray, Sarah McMahon

31 October 2024

1 Introduction

An object's moment of inertia acts as a component used to calculate torque required to accelerate the object around a selected pivot point. Across this lab we intend to apply this concept alongside the idea behind torque to determine properties of an object, and refine our understanding of how moment of inertia, the radius of an object and the position of an objects mass will impact the resulting torque required to achieve a desired angular velocity.

2 Problem 1

For this experiment we intend to complete several runs to ensure quality data is gathered.

1. Attach the mass to the end of the string, measure the radius of the axle.
2. Calculate the circumference of the axle.
3. Calculate length of the string coiled around the axle by multiplying the number of rotations by the circumference.
4. Release the mass and measure the time for the falling mass to hit the ground using a stopwatch.

2.1 Derivation

First, for the tension force of the string connecting the mass to the platter axle:

$$\sum F = T - mg = ma$$

$$T = m(g + a)$$

Next, the torque this forces exerts on the axle of radius r:

$$\tau = rT\sin(\theta) = rm(g + a)$$

Relating this to the moment of inertia of the platter and object using Newton's second law:

$$\tau = I\alpha$$

$$rm(g+a) = (I_P + I_M)\frac{a}{r}$$

Solving for the moment of inertia of the platter and object:

$$\frac{r^2m(g+a)}{a} = I_P + I_M = I$$

However, I_M is ignored for this problem since I_P is what is being found. For the experiment, we calculate acceleration from the time it takes the mass to fall a certain distance d.

$$d = \frac{1}{2}at^2$$

$$a = \frac{2d}{t^2}$$

$$\frac{r^2m(g+2d/t^2)t^2}{2d} = I_P$$

$$\frac{r^2m(gt^2+2d)}{2d} = I_P$$

$$\boxed{r^2m\left(\frac{gt^2}{2d}+1\right) = I_P}$$

We measure d using a number of wraps around the axle of rotation of radius r. The error propagation is simple for this too:

$$d = -2\pi r$$

$$\sigma_d = 2\pi\sigma_r$$

2.2 Error Propagation

We use the same method as was done in Lab 9 (see there for and explanation). The variables r, m, and d have instrument limits of error. However t is measured by hand and has and unknown human error, meaning it must be calculated from the standard error of multiple measurements:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

$$\sigma = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$$

We use the following multivariable variance formula for our variables:

$$\sigma_I = \sqrt{\left(\frac{\partial I}{\partial r}\sigma_r\right)^2 + \left(\frac{\partial I}{\partial m}\sigma_m\right)^2 + \left(\frac{\partial I}{\partial d}\sigma_d\right)^2 + \left(\frac{\partial I}{\partial t}\sigma_t\right)^2}$$

The partial derivatives are:

$$\frac{\partial I}{\partial r} = 2rm\left(\frac{gt^2}{2d} + 1\right) \qquad \frac{\partial I}{\partial m} = r^2\left(\frac{2gt}{2d}\right)$$

$$\frac{\partial I}{\partial d} = r^2m\left(1 - \frac{gt^2}{2d^2}\right) \qquad \frac{\partial I}{\partial t} = r^2m\left(\frac{2gt}{2d}\right)$$

[38]:

```
from math import *
import numpy as np

g = 9.81  # m/s^2

# Functions


def platter_inertia(r, m, d, t):
    return (r**2) * m * (((g * (t**2)) / (2 * d)) + 1)


def platter_inertia_error(r, m, d, t, e_r, e_m, e_d, e_t):

    dI_dr = 2 * r * m * (((g * (t**2)) / (2 * d)) + 1)
    dI_dm = (r**2) * ((2 * g * t) / (2 * d))
    dI_dd = (r**2) * m * (1 - ((g * (t**2)) / (2 * (d**2))))
    dI_dt = (r**2) * m * (((2 * g * t) / (2 * d)))

    parts = [dI_dr * e_r, dI_dm * e_m, dI_dd * e_d, dI_dt * e_t]

    return sqrt(sum([part**2 for part in parts]))


# Measured (ILE)
r = 0.02  # m
e_r = 0  # m
m = 0.55  # m
e_m = 0  # kg

num_rot = 4  # for calculating d
d = -2 * pi * r * num_rot  # m
```

```
e_d = 2 * pi * e_r  # m

# Measured by hand (6 trials)
t = [1.85, 1.7, 1.78, 1.73, 1.7, 1.79]  # s
e_t = np.std(t) / sqrt(len(t))  # s

p_inertia = abs(platter_inertia(r, m, d, np.mean(t)))
e_p_inertia = platter_inertia_error(r, m, d, np.mean(t), e_r, e_m, e_d, e_t)

print(f"Platter moment of inertia: {p_inertia} \u00B1 {e_p_inertia} kg m^2")
```

```
Platter moment of inertia: 0.006417342912852919 ± 0.00016637340716253436 kg m^2
```

2.3 Analysis

Through our experimentation and calculations we had managed to determine a moment of inertia for the platter of 0.0064, with an error +/- 0.00016. We feel that, though our calculations do not include friction nor air resistance, they still accurately reflect the properties of the disk.

3 Problem 2

We again are completing multiple runs of this experiment. 1. Calculate r for the new mass 2. Place the new mass on the platter 3. Attach the mass to the end of the string, measure the radius of the axle. 4. Calculate the circumference of the axle. 5. Calculate length of the string coiled around the axle by multiplying the number of rotations by the circumference. 6. Release the mass and measure the time for the falling mass to hit the ground using a stopwatch.

3.1 Derivation

Since there is now an added mass, the equation is:

$$r^2 m \left(\frac{gt^2}{2d} + 1\right) = I_P + I_M$$

$$\boxed{r^2 m \left(\frac{gt^2}{2d} + 1\right) - I_P = I_M}$$

For error propagation, the previous derivation is used with simple sum error propagation:

$$\sigma_{I_M} = \sqrt{\sigma^2 + \sigma_I^2}$$

where σ uses the same algorithm as before and σ_I is already calculated from the preivous problem. Since we are trying to caclulate the mass of the specific objects (marked by i), we need to take into account the different equations for moments of inertia:

$$I_M = c_i m_i r_i^2$$

$$\frac{I_M}{c_i r_i^2} = m_i$$

Here c is a constant dependent on the shape of the object. Error propagation is again fairly simply given a known σ_{I_M}.

$$\sigma_{m_i} = \sqrt{\left(\frac{\partial m_i}{\partial r_i}\sigma_{r_i}\right)^2 + \left(\frac{\partial m_i}{\partial I_M}\sigma_{I_M}\right)^2}$$

The partial derivatives are:

$$\frac{\partial m_i}{\partial r_i} = \frac{-2I_M}{c_i r_i^3} \qquad \frac{\partial m_i}{\partial I_M} = \frac{1}{c_i r_i^2}$$

[47]:

```
def object_mass_error(c, p_inertia, r, e_p_inertia, e_r):

    dm_dI = (-2 * p_inertia) / (c * (r**3))
    dm_dr = 1 / (c * (r**2))

    parts = [dm_dI * e_p_inertia, dm_dr * e_r]

    return sqrt(sum([part**2 for part in parts]))


# Measured (ILE is divided by two since diameter is measured)
disk_r, disk_e_r, disk_c = 0.25 / 2, 0.001 / 2, 0.5  # m
ring_r, ring_e_r, ring_c = 0.13 / 2, 0.001 / 2, 1  # m
bar_r, bar_e_r, bar_c = 0.223 / 2, 0.001 / 2, 1 / 3  # m

# Measured by hand
disk_t = np.array([2.63, 2.80, 2.70, 2.55, 2.66, 2.76])  # s
disk_e_t = np.std(disk_t) / sqrt(len(disk_t))  # s

ring_t = np.array([2.21, 2.16, 2.08, 2.10, 2.07, 2.05])  # s
ring_e_t = np.std(ring_t) / sqrt(len(ring_t))  # s

bar_t = np.array([2.13, 2.13, 2.10, 2.16, 2.16, 2.20])  # s
bar_e_t = np.std(bar_t) / sqrt(len(bar_t))  # s

# Calculate total moments of inertia - platter moment of inertia = objects'
# moment of inertiaa
disk_inertia = abs(platter_inertia(r, m, d, np.mean(disk_t)) - p_inertia)
ring_inertia = abs(platter_inertia(r, m, d, np.mean(ring_t)) - p_inertia)
bar_inertia = abs(platter_inertia(r, m, d, np.mean(bar_t)) - p_inertia)

# Error propagation for total moments of inertia
```

```
disk_e = platter_inertia_error(r, m, d, np.mean(disk_t), e_r, e_m, e_d,␣
 ↪disk_e_t)
ring_e = platter_inertia_error(r, m, d, np.mean(ring_t), e_r, e_m, e_d,␣
 ↪ring_e_t)
bar_e = platter_inertia_error(r, m, d, np.mean(bar_t), e_r, e_m, e_d, bar_e_t)

# Error propagation for objects' moments of inertia
disk_e_i = sqrt(disk_e**2 + e_p_inertia**2)
ring_e_i = sqrt(ring_e**2 + e_p_inertia**2)
bar_e_i = sqrt(bar_e**2 + e_p_inertia**2)

# Calculate object masses
disk_m = disk_inertia / (disk_c * (disk_r**2))
ring_m = ring_inertia / (ring_c * (ring_r**2))
bar_m = bar_inertia / (bar_c * (bar_r**2))

# Error propagation of object masses
disk_e_m = object_mass_error(disk_c, disk_inertia, disk_r, disk_e_i, disk_e_r)
ring_e_m = object_mass_error(ring_c, ring_inertia, ring_r, ring_e_i, ring_e_r)
bar_e_m = object_mass_error(bar_c, bar_inertia, bar_r, bar_e_i, bar_e_r)

print(
    f"Disk mass: {round(disk_m, 3)}"
    f" \u00B1 {round(disk_e_m, 3)} kg\n"
    f"Ring mass: {round(ring_m, 3)}"
    f" \u00B1 {round(ring_e_m, 3)} kg\n"
    f"Bar mass: {round(bar_m, 3)}"
    f" \u00B1 {round(bar_e_m, 3)} kg"
)
```

```
Disk mass: 2.772 ± 0.067 kg
Ring mass: 3.733 ± 0.122 kg
Bar mass: 3.883 ± 0.121 kg
```

3.2 Analysis

While the first experiment did accurately model the properties of the disk under its original conditions, we found ourselves struggling to predict the masses of the disk at 2.772 +/- 0.0kg ring at 3.733 +/- 0.122kg and the bar at 3.883 +/- 0.121kg. Unfortunately, it is likely that in our misguided efforts to debug working code, we managed to put too little focus on the experiment being run concurrently. It is likely we may have miscounted the number of coils, resulting in a bad distance value, or mistimed the fall.

4 Conclusion

It is clear through this experiment how the moment of inertia depends on the shape of objects, and further how it resists changes to momentum just like its linear equivalent, mass. Despite some set backs in figuring out issues with the code and equations (where there really were none we could find), the lab when well. Because we spend so long working on the aforementioned issues, we neglected to look more into our methods of experimentation. However, We managed to still get somewhat reasonable values (all around 3 kg) for the objects and a reasonable value for the moment of inertia of the platter (0.006 kg m^2).