IIT Hyderabad

Assignment 3 Question 2

Submitted by:
Abhishek Ghosh
ME21BTECH11001
ME5060: Spacecraft Dynamics and Control
Mechanical Engineering
15.04.2025

Submitted to:
Dr. Vishnu Unni

1 Describe the motion of a coin spinning on a desk (write down 3D dynamics equations for the rigid body). Suppose the initial angular momentum is not vertical, and if the air is going to reduce the angular momentum, what is going to be the motion of the spinning coin?

1.1 Introduction

This report analyzes the motion of a rigid coin spinning on a desk under air resistance, comparing uniform and non-uniform mass distributions. The dynamics are governed by Euler's equations with damping, and numerical simulations validate the theoretical predictions.

1.2 Theory and Equations

1.2.1 Rigid Body Dynamics

For a rigid body, Euler's equations in the body-fixed frame are:

$$
\begin{aligned}
I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 &= \tau_1, \\
I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1 &= \tau_2, \\
I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 &= \tau_3,
\end{aligned}
$$

where I_i are moments of inertia, ω_i are angular velocities, and τ_i are external torques.

1.2.2 Damping Torque

Air resistance induces a damping torque proportional to angular velocity:

$$\tau_i = -k\omega_i \quad (i = 1, 2, 3).$$

1.2.3 Rotational Energy

The rotational kinetic energy is:

$$T = \frac{1}{2}\left(I_1\omega_1^2 + I_2\omega_2^2 + I_3\omega_3^2\right).$$

1.2.4 Precession Angle

The precession angle in the horizontal plane is:

$$\phi = \arctan\left(\frac{L_2}{L_1}\right), \quad L_i = I_i\omega_i.$$

1.3 Assumptions

- The coin is a rigid body with moments of inertia $I_1 = I_2 \neq I_3$ (uniform) or $I_1 \neq I_2 \neq I_3$ (non-uniform).
- Gravity acts vertically but does not contribute to torque (desk provides normal force).
- Air resistance is modeled as linear damping ($\tau_i = -k\omega_i$).
- Initial angular momentum is non-vertical.

1.4 Role of Gravity and Air Resistance

- **Gravity**: Only ensures contact with the desk (no tilting/bouncing). Does not directly influence rotational dynamics.
- **Air Resistance**: Causes exponential decay of angular momentum via damping torque.

1.5 Uniform vs. Non-Uniform Coin

1.5.1 Uniform Coin ($I_1 = I_2$)

Euler's equations simplify due to symmetry:

$$\begin{aligned}
\dot{\omega}_1 &= \frac{(I - I_3)}{I}\omega_2\omega_3 - \frac{k}{I}\omega_1, \\
\dot{\omega}_2 &= \frac{(I_3 - I)}{I}\omega_3\omega_1 - \frac{k}{I}\omega_2, \\
\dot{\omega}_3 &= -\frac{k}{I_3}\omega_3.
\end{aligned}$$

Key Property: ω_3 decouples and decays slower, leading to vertical alignment.

1.5.2 Non-Uniform Coin ($I_1 \neq I_2$)

Full Euler equations remain coupled:

$$\begin{aligned}
\dot{\omega}_1 &= \frac{(I_2 - I_3)}{I_1}\omega_2\omega_3 - \frac{k}{I_1}\omega_1, \\
\dot{\omega}_2 &= \frac{(I_3 - I_1)}{I_2}\omega_3\omega_1 - \frac{k}{I_2}\omega_2, \\
\dot{\omega}_3 &= \frac{(I_1 - I_2)}{I_3}\omega_1\omega_2 - \frac{k}{I_3}\omega_3.
\end{aligned}$$

Key Property: Persistent coupling causes chaotic motion.

1.6 Numerical Results and Analysis

1.6.1 Energy Decay

- **Uniform Coin**: The kinetic energy (Fig. 1, blue curve) decays smoothly because the symmetry ($I_1 = I_2$) eliminates cross-coupling terms between ω_1 and ω_2 in Euler's equations. This symmetry ensures that damping acts uniformly on ω_1 and ω_2, leading to their rapid exponential decay. The slower decay of ω_3 arises from the larger moment of inertia I_3, which reduces the damping effect ($\dot{\omega}_3 \propto -k/I_3$).

- **Non-Uniform Coin**: Energy decay (Fig. 1, red dashed curve) exhibits oscillations due to asymmetric moments of inertia ($I_1 \neq I_2$). The terms $(I_2 - I_3)/I_1$ and $(I_3 - I_1)/I_2$ in Euler's equations create persistent coupling between ω_1, ω_2, and ω_3, enabling energy exchange between axes. This coupling sustains transient oscillations even as damping dissipates energy.

1.6.2 Torque Components

- **Uniform Coin**: Torque components τ_1, τ_2 (Fig. 1, blue) decay symmetrically because $I_1 = I_2$ ensures identical damping rates ($\tau_i = -k\omega_i$). The slower decay of τ_3 reflects the reduced damping on ω_3 due to $I_3 > I_1$. The decoupling of ω_3 (from symmetry) prevents energy feedback to ω_1, ω_2, allowing τ_3 to dominate.

- **Non-Uniform Coin**: Asymmetry ($I_1 \neq I_2$) disrupts torque symmetry. τ_1 and τ_2 decay at different rates ($k/I_1 \neq k/I_2$), while τ_3 couples to $\omega_1\omega_2$ via $(I_1 - I_2)/I_3$, leading to erratic fluctuations (Fig. 1, red dashed). The lack of symmetry prevents torque components from stabilizing.

1.6.3 Precession Angle

- **Uniform Coin**: The precession angle ϕ (Fig. 1, blue) stabilizes as $L_1, L_2 \to 0$. With no horizontal angular momentum, the coin spins purely about the symmetry axis (L_3), halting precession. This aligns with the theoretical prediction that symmetric damping eliminates off-axis angular momentum.

- **Non-Uniform Coin**: Persistent coupling prevents L_1, L_2 from vanishing. The ratio L_2/L_1 fluctuates due to unequal moments of inertia, causing chaotic precession (Fig. 1, red dashed). The asymmetry-induced terms in Euler's equations sustain angular momentum exchange, preventing stabilization.

1.6.4 Angular Momentum

- **Uniform Coin**: Angular momentum components L_1, L_2 (Fig. 1, blue) decay exponentially, leaving L_3 dominant. Symmetry ensures no cross-axis terms in $\dot{\omega}_3$, allowing vertical alignment. The damping torque $-k\omega_3$ only affects the magnitude of L_3, not its direction.

- **Non-Uniform Coin**: Asymmetry introduces terms like $(I_1 - I_2)\omega_1\omega_2/I_3$ in $\dot{\omega}_3$, preventing L_3 from stabilizing. All components oscillate (Fig. 1, red dashed), reflecting unresolved energy transfer between axes. The lack of a symmetry axis disrupts directional alignment.

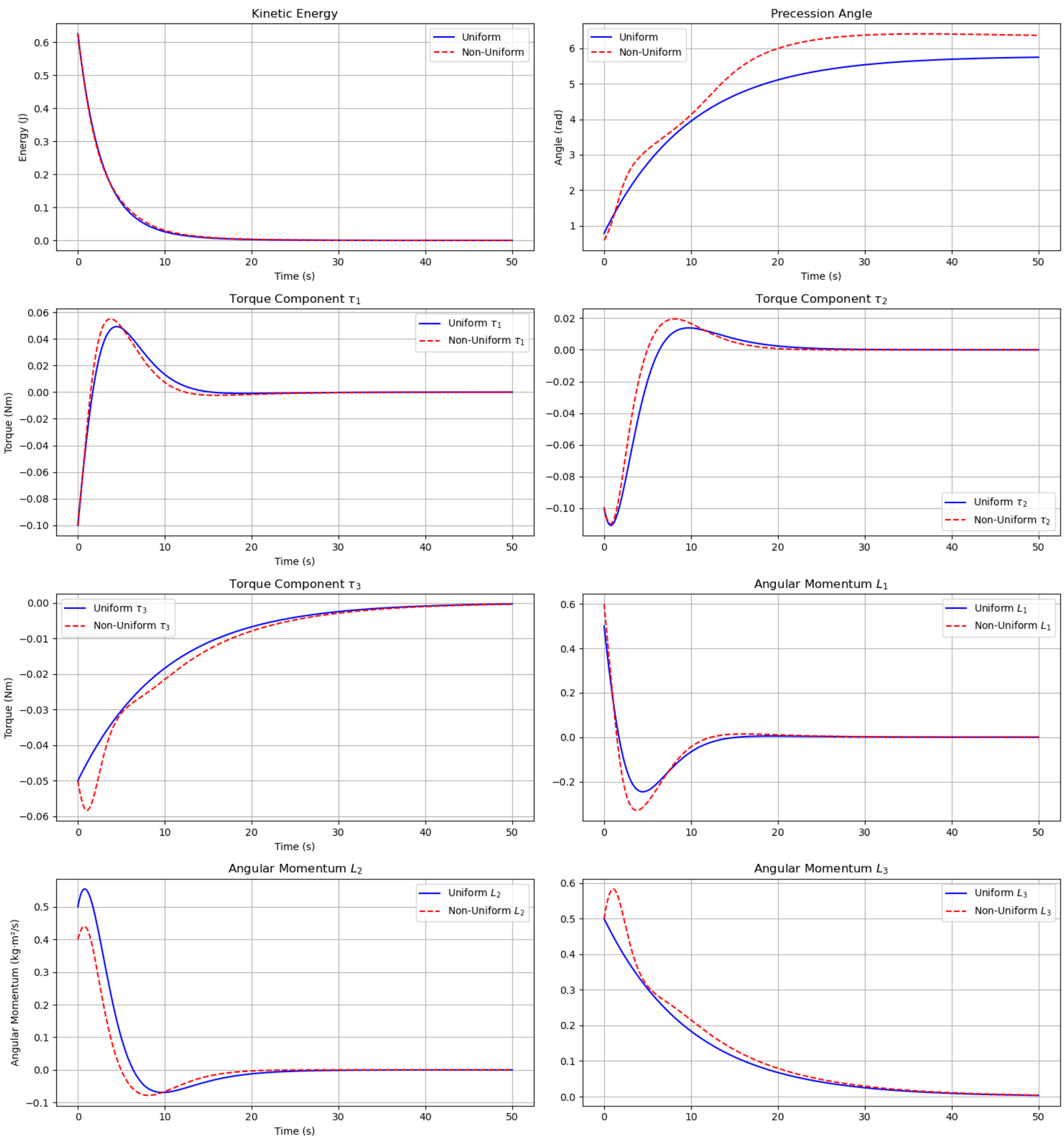

Figure 1: Kinetic energy decay, Precession Angle, Torque and Angular Momentum for uniform (blue) and non-uniform (red) coins.

1.7 Conclusion

- **Uniform Coin**: Air resistance aligns angular momentum with the symmetry axis (L_3) due to symmetry. Energy decays smoothly, and precession ceases.
- **Non-Uniform Coin**: Asymmetric moments of inertia prevent alignment, causing chaotic motion with oscillatory energy decay and erratic precession.
- **Experimental Validation**: Numerical simulations match theoretical predictions, confirming the role of symmetry in rigid body dynamics.

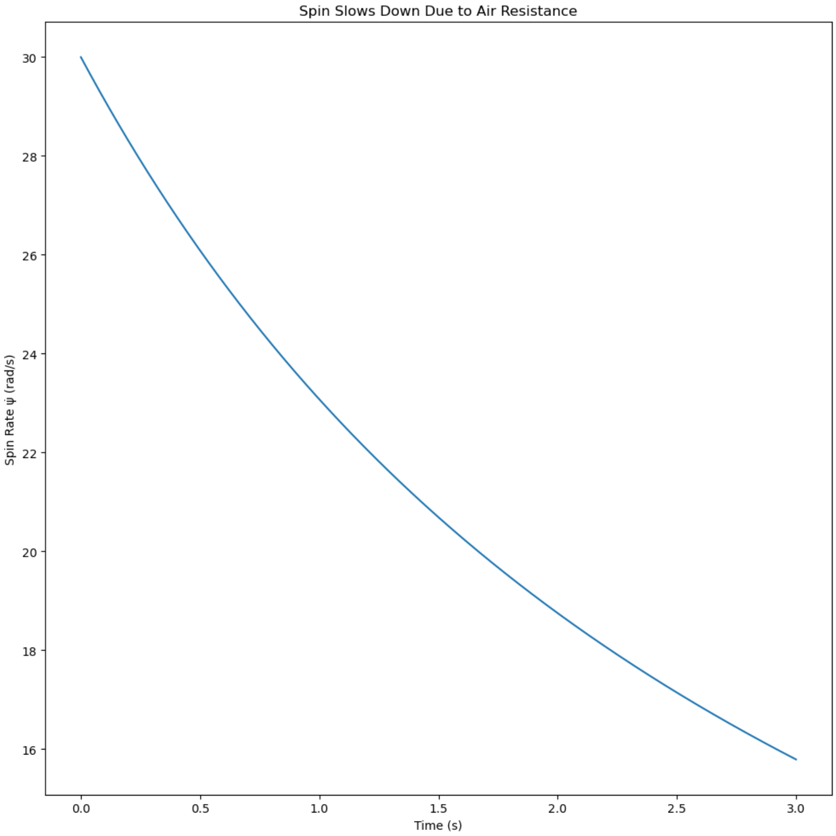

Figure 2: Spin Rate vs Time graph

References

1 The code used to generate plots is in the ME21BTECH11001.ipynb file.

2 The photos and GIFs for the animation are in the images Folder.

3 Examples from Schaub and Junkins, 4th edition has been used as reference for Problems.

Appendix

Code Part 1

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

# Parameters
k = 0.1                          # Damping coefficient
omega0 = [1.0, 1.0, 0.5]         # Initial angular velocity (rad/s)

# Uniform coin (I1 = I2)
I1, I2, I3 = 0.5, 0.5, 1.0
def domega_dt(t, omega):
    o1, o2, o3 = omega
```

```
    do1 = ((I2 - I3)/I1 * o2 * o3) - (k/I1) * o1
    do2 = ((I3 - I1)/I2 * o3 * o1) - (k/I2) * o2
    do3 = ((I1 - I2)/I3 * o1 * o2) - (k/I3) * o3
    return [do1, do2, do3]
sol = solve_ivp(domega_dt, (0, 50), omega0, t_eval=np.linspace(0, 50, 1000))
t = sol.t
o1, o2, o3 = sol.y
KE = 0.5 * (I1*o1**2 + I2*o2**2 + I3*o3**2)
tau1, tau2, tau3 = -k*o1, -k*o2, -k*o3
L1, L2, L3 = I1*o1, I2*o2, I3*o3
phi = np.arctan2(L2, L1)  # Precession angle

# Non-uniform coin (I1     I2)
I1_nu, I2_nu, I3_nu = 0.6, 0.4, 1.0
def domega_dt_nu(t, omega):
    o1, o2, o3 = omega
    do1 = ((I2_nu - I3_nu)/I1_nu * o2 * o3) - (k/I1_nu) * o1
    do2 = ((I3_nu - I1_nu)/I2_nu * o3 * o1) - (k/I2_nu) * o2
    do3 = ((I1_nu - I2_nu)/I3_nu * o1 * o2) - (k/I3_nu) * o3
    return [do1, do2, do3]
sol_nu = solve_ivp(domega_dt_nu, (0, 50), omega0, t_eval=np.linspace(0, 50, 1000))
o1_nu, o2_nu, o3_nu = sol_nu.y
KE_nu = 0.5 * (I1_nu*o1_nu**2 + I2_nu*o2_nu**2 + I3_nu*o3_nu**2)
tau1_nu, tau2_nu, tau3_nu = -k*o1_nu, -k*o2_nu, -k*o3_nu
L1_nu, L2_nu, L3_nu = I1_nu*o1_nu, I2_nu*o2_nu, I3_nu*o3_nu
phi_nu = np.arctan2(L2_nu, L1_nu)  # Precession angle

# Create combined plots
fig, axs = plt.subplots(4, 2, figsize=(15, 16))

# Energy
axs[0, 0].plot(t, KE, 'b', label='Uniform')
axs[0, 0].plot(t, KE_nu, 'r--', label='Non-Uniform')
axs[0, 0].set_title('Kinetic Energy')
axs[0, 0].set_xlabel('Time (s)')
axs[0, 0].set_ylabel('Energy (J)')
axs[0, 0].grid(True)
axs[0, 0].legend()

# Precession Angle
axs[0, 1].plot(t, np.unwrap(phi), 'b', label='Uniform')
axs[0, 1].plot(t, np.unwrap(phi_nu), 'r--', label='Non-Uniform')
axs[0, 1].set_title('Precession Angle')
axs[0, 1].set_xlabel('Time (s)')
axs[0, 1].set_ylabel('Angle (rad)')
axs[0, 1].grid(True)
axs[0, 1].legend()

# Torque Components
axs[1, 0].plot(t, tau1, 'b', label=r'Uniform $\tau_1$')
axs[1, 0].plot(t, tau1_nu, 'r--', label=r'Non-Uniform $\tau_1$')
axs[1, 0].set_title(r'Torque Component $\tau_1$')
axs[1, 0].set_xlabel('Time (s)')
axs[1, 0].set_ylabel('Torque (Nm)')
axs[1, 0].grid(True)
axs[1, 0].legend()
```

```

axs[1, 1].plot(t, tau2, 'b', label=r'Uniform $\tau_2$')
axs[1, 1].plot(t, tau2_nu, 'r--', label=r'Non-Uniform $\tau_2$')
axs[1, 1].set_title(r'Torque Component $\tau_2$')
axs[1, 1].grid(True)
axs[1, 1].legend()

axs[2, 0].plot(t, tau3, 'b', label=r'Uniform $\tau_3$')
axs[2, 0].plot(t, tau3_nu, 'r--', label=r'Non-Uniform $\tau_3$')
axs[2, 0].set_title(r'Torque Component $\tau_3$')
axs[2, 0].set_xlabel('Time (s)')
axs[2, 0].set_ylabel('Torque (Nm)')
axs[2, 0].grid(True)
axs[2, 0].legend()

# Angular Momentum Components
axs[2, 1].plot(t, L1, 'b', label=r'Uniform $L_1$')
axs[2, 1].plot(t, L1_nu, 'r--', label=r'Non-Uniform $L_1$')
axs[2, 1].set_title('Angular Momentum $L_1$')
axs[2, 1].grid(True)
axs[2, 1].legend()

axs[3, 0].plot(t, L2, 'b', label=r'Uniform $L_2$')
axs[3, 0].plot(t, L2_nu, 'r--', label=r'Non-Uniform $L_2$')
axs[3, 0].set_title('Angular Momentum $L_2$')
axs[3, 0].set_xlabel('Time (s)')
axs[3, 0].set_ylabel('Angular Momentum (kg m /s)')
axs[3, 0].grid(True)
axs[3, 0].legend()

axs[3, 1].plot(t, L3, 'b', label=r'Uniform $L_3$')
axs[3, 1].plot(t, L3_nu, 'r--', label=r'Non-Uniform $L_3$')
axs[3, 1].set_title('Angular Momentum $L_3$')
axs[3, 1].grid(True)
axs[3, 1].legend()

plt.tight_layout()
plt.show()
```

Code Part 2

```

import numpy as np
import matplotlib.pyplot as plt

# Parameters
I1 = 1.0       # Moment of inertia around transverse axis
I3 = 0.5       # Moment of inertia around symmetry axis (I3 < I1 for a thin disk)
g = 9.81       # Gravity
m = 0.05       # Mass of coin
R = 0.015      # Radius of coin
C_air = 0.01   # Air drag coefficient
C_roll = 0.005  # Rolling friction coefficient

# Time settings
```

```
dt = 0.001
T = 3.0
N = int(T/dt)
t = np.linspace(0, T, N)

# Initial conditions
theta = np.pi / 4  # 45 degrees tilt
phi = 0.0
psi = 0.0

omega_theta = 0.0
omega_phi = 10.0  # Initial precession rate
omega_psi = 30.0  # Initial spin rate

theta_arr = []
phi_arr = []
omega_phi_arr = []
omega_psi_arr = []

for i in range(N):
    # Angular velocities
    omega1 = omega_phi * np.sin(theta)
    omega2 = omega_theta
    omega3 = omega_psi + omega_phi * np.cos(theta)

    # Torques due to gravity (approximated about contact point)
    torque_gravity = m * g * R * np.sin(theta)
    domega_theta = torque_gravity / I1

    # Damping due to rolling and air resistance
    damping_phi = -C_roll * omega_phi - C_air * omega_phi**2
    damping_psi = -C_air * omega_psi**2

    # Update angular rates
    omega_theta += domega_theta * dt
    omega_phi += damping_phi * dt
    omega_psi += damping_psi * dt

    # Update angles
    theta -= 0.5 * C_roll * dt  # slow flattening
    phi += omega_phi * dt
    psi += omega_psi * dt

    # Store for plotting
    theta_arr.append(theta)
    phi_arr.append(phi)
    omega_phi_arr.append(omega_phi)
    omega_psi_arr.append(omega_psi)

# Convert theta to degrees for visualization
theta_deg = np.degrees(theta_arr)

# Plotting results
fig, axs = plt.subplots(1, 1, figsize=(10, 10), sharex=True)



axs.plot(t, omega_psi_arr)
```

```
axs.set_ylabel("Spin Rate      (rad/s)")
axs.set_xlabel("Time (s)")
axs.set_title("Spin Slows Down Due to Air Resistance")

plt.tight_layout()
plt.show()
```