Asta ruotante

Eugenio Barbieri Viale

4 marzo 2024

Il centro di massa del sistema assa-proiettile si muove di moto rettilineo uniforme parallelamente alla direzione y. Ha quindi legge oraria:

$$y(t) = v_{cm}t$$

La distanza percorsa dal centro di massa in un periodo di rotazione T è quindi:

$$\Delta y = v_{cm}T$$

Fissando l'origine del sistema di riferimento inerziale nel centro geometrico dell'asta prima dell'urto, il centro di massa ha coordinata x_{cm} e velocità v_{cm}:

$$x_{cm} = \frac{m}{2(m+M)}l \qquad v_{cm} = \frac{m}{m+M}v_0$$

Se β è il rapporto tra la massa dell'asta M e la massa del proiettile m, allora si ottiene che $M = \beta m$. Sostituendo risulta che:

$$x_{cm} = \frac{1}{2(\beta+1)}l \qquad v_{cm} = \frac{1}{\beta+1}v_0$$

La distanza percorsa dal CM in un periodo di rivoluzione dell'asta è perciò:

$$\Delta y = \frac{1}{\beta+1}v_0T$$

Ora, sapendo che il sistema assa-proiettile è un sistema isolato, poichè la risultante dei momenti esterni è nulla (le forze tra asta e proiettile sono interne), il momento angolare si conserva. Inoltre, per il teorema di König:

$$L_0 = L_f \qquad \rightarrow \qquad L_0 = L_{trasl} + L_{rot}$$

L_0 e L_{trasl} hanno come polo l'orgine del sistema di riferimento, mentre L_{rot} è calcolato rispetto al CM. Dato che all'inizio l'asta è ferma:

$$mv_0\frac{l}{2} = (m+M)v_{cm}x_{cm} + I_{cm}\omega$$

Ora è necessario calcolare il momento d'inerzia del sistema rispetto al suo centro di massa. Applicando il teorema di Steiner:

$$\begin{aligned}
I_{cm} &= \frac{1}{12}Ml^2 + Mx_{cm}^2 + m(\frac{1}{2}l - x_{cm})^2 && (1)\\
&= \frac{\beta}{12}ml^2 + \beta ml^2\frac{1}{4(\beta+1)^2} + \frac{1}{4}ml^2(1 - \frac{1}{\beta+1}) && (2)\\
&= ml^2\left(\frac{\beta}{12} + \frac{\beta}{4(\beta+1)^2} + \frac{1}{4(\beta+1)^2} - \frac{1}{2(\beta+1)} + \frac{1}{4}\right) && (3)\\
&= ml^2\left(\frac{\beta}{12} + \frac{\beta+1}{4(\beta+1)^2} - \frac{1}{2(\beta+1)} + \frac{1}{4}\right) && (4)\\
&= ml^2\left(\frac{\beta}{12} + \frac{1}{4(\beta+1)} - \frac{1}{2(\beta+1)} + \frac{1}{4}\right) && (5)\\
&= ml^2\left(\frac{\beta^2+\beta+3-6+3\beta+3}{12(\beta+1)}\right) && (6)\\
&= ml^2\frac{\beta^2+4\beta}{12(\beta+1)} && (7)
\end{aligned}$$

Facendo tutte le sostituzioni necessarie nell'equazione dei momenti angolari:

$$\begin{aligned}
\frac{1}{2}mlv_0 &= (1+\beta)m\frac{1}{\beta+1}v_0\frac{1}{2(\beta+1)}l + ml^2\frac{\beta^2+4\beta}{12(\beta+1)}\frac{2\pi}{T} && (8)\\
\frac{1}{2}v_0 &= \frac{1}{2(\beta+1)}v_0 + \frac{2\pi l\beta(\beta+4)}{12T(\beta+1)} && (9)\\
\frac{1}{2}(\beta+1)v_0 &= \frac{1}{2}v_0 + \frac{2\pi l\beta(\beta+4)}{12T} && (10)\\
6(\beta+1)v_0 - 6v_0 &= \frac{2\pi l\beta(\beta+4)}{T} && (11)
\end{aligned}$$

Alla fine si ottiene che il periodo:

$$T = \frac{\pi l(\beta+4)}{3v_0}$$

Sostituendo nella legge oraria del CM e facendo qualche altra semplificazione, si ottiene:

$$y = \frac{\pi l(\beta+4)}{3(\beta+1)}$$

Ora calcoliamo il rapporto tra la distanza percorsa e la lunghezza l dell'asta, in modo da ottenere una quantità adimensionale:

$$y(\beta) = \frac{\pi\beta+4\pi}{3\beta+3}$$

Questa è evidentemente un'iperbole equilatera riferita agli asintoti. I suoi asintoti sono $x = -1$ e $y = \frac{\pi}{3}$. Ricordando che $\beta > 0$, poichè è il rapporto tra due masse, e anche che $y \geq 0$, dato che è una distanza, la massima distanza percorsa si trova all'intersezione con l'asse y.

Perciò la massima distanza percorsa durante una rivoluzione dell'asta si trova al tendere a 0 del rapporto β tra le due masse. Qui il codice della simulazione, che ho realizzato usando la libreria di Python "Pygame".

```python
import pygame, sys
import numpy as np

pygame.init()
clock = pygame.time.Clock()

X,Y = 1000,1000

screen = pygame.display.set_mode([X,Y])
pygame.display.set_caption("Eugenio Barbieri Viale")

# Grid propeties
x = 0
y = Y
dist_grid = 25

# Ratio of mass of the rod and mass of the bullet
beta = 2

# Half of the length of the rod
R = 230

# Position of the center of mass of the system
x_cm = (R)/(beta+1)

# Initial angles of the to ends of the rod
a1 = 0
a2 = np.pi

# Coordinates of the center of mass with respect to the top left
x0 = X//2 + x_cm
y0 = Y//2

# Distances of the ends from the center of mass
l1 = R + x_cm
l2 = R - x_cm

# The two ends and the cm
cm = pygame.math.Vector2(x0, y0)
p1 = pygame.math.Vector2(0,0)
p2 = pygame.math.Vector2(0,0)
```

```
d = R

# Initial position and velocity of the bullet
pos = pygame.math.Vector2(X//2 + d-6, y0 + 200)
vel = 2

v_cm = (vel)/(beta+1) # velocity of the center of mass
va = 3*vel/(R*(beta+4)) # angular velocity

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    screen.fill((200,200,200))

    cos_norm = abs(np.cos(a1))/np.cos(a1)
    borderX = x0 + np.cos(a1)*(d+10)*cos_norm

    if np.sin(a1) >= 0:
        if pos.y <= y0 and pos.x <= borderX:
            y += v_cm

            a1 += va
            a2 += va

            pos.x = x0 - np.cos(a2)*(d-x_cm)
            pos.y = y0 + np.sin(a2)*(d-x_cm)

        else:
            pos.y -= vel

    if np.sin(a1) < 0:
        if pos.y > y0 and pos.x <= borderX:
            y += v_cm

            a1 += va
            a2 += va

            pos.x = x0 - np.cos(a2)*(d-x_cm)
            pos.y = y0 + np.sin(a2)*(d-x_cm)

        else:
            pos.y -= vel

    p1.x = cm.x - np.cos(a1)*l1
    p1.y = cm.y + np.sin(a1)*l1
```

```
    p2.x = cm.x - np.cos(a2)*l2
    p2.y = cm.y + np.sin(a2)*l2

    # Draw grid
    yax = pygame.font.SysFont("Comic Sans MS", 20)
    for i in range(100):
        pygame.draw.line(screen, (0,0,0), (x+dist_grid*i,0),
             (x+dist_grid*i,Y), 2)
        if y > dist_grid:
            write = yax.render(str(dist_grid*i), 1, (255,255,255))
            screen.blit(write, (X//2+3,y-dist_grid*i-Y//2))
            pygame.draw.line(screen, (0,0,0), (0,y-dist_grid*i), (X,
                 y-dist_grid*i), 2)

    pygame.draw.line(screen, (255,0,0), (X//2,0), (X//2,Y), 2)
    pygame.draw.line(screen, (255,0,0), (0,y-Y//2), (X,y-Y//2), 2)

    # Rod
    pygame.draw.line(screen, (255,255,0), cm, p1, 8)
    pygame.draw.line(screen, (0,0,255), cm, p2, 8)

    # Bullet
    pygame.draw.circle(screen, (255,0,0), pos, 6)

    font = pygame.font.SysFont("Comic Sans MS", 35)
    # Conversion from pixel/frame to pixel/second (60 frames every
         second)
    write1 = font.render("Angular velocity: " + str(round(va*60,4)) + "
         rad/s", 1, (255,255,255))
    write2 = font.render("Velocity of the center of mass: " +
         str(round(v_cm*60,2)) + " px/s", 1, (255,255,255))

    screen.blit(write1, (10,10))
    screen.blit(write2, (10,30))

    pygame.display.flip()
    clock.tick(60)
    pygame.display.update()

pygame.quit()
```