

# Relatório 6 - Manobras Orbitais (AGA0521)

Kethelin Parra Ramos - 9898349

## I. INTRODUÇÃO

Para verificar a estabilidade do movimento de um corpo rígido sem torque, devemos calcular o momento de inércia em cada eixo (x,y,z).

1. O eixo que apresentar maior momento de inércia será o de maior estabilidade.
2. Se não houver dissipação da energia de rotação, a rotação será instável para o eixo que apresentar momento de inércia intermediário.
3. Se houver dissipação da energia de rotação, a rotação será instável para o eixo que apresentar menor momento de inércia

Nos exercícios desta semana foram considerados casos em que não houvesse dissipação de energia, portanto, apenas os itens (1) e (2) foram verificados.

## II. MÉTODOS

Para resolver o problema do torque livre nos dois casos do enunciado, foi criada a função `torque_livre()` que seguia os passos mostrados em aula (código abaixo). Todos os dados utilizados na atividade foram retirados do slide do enunciado.

```
1  #Bibliotecas
2  import numpy as np
3
4  """                               Funções                               """
5  #Função que resolve o problema do torque livre
6  def torque_livre(mc,mh,rc,lc,lh):
7      print("\nResultados intermediarios:")
8
```

```

9      #Momentos de inercia de cada eixo do cilindro B
10     I_cilx = (1/4)*mc*(rb**2) + (1/12)*mc*lc**2
11     I_cily = I_cilx
12     I_cilz = (1/2)*mc*rb**2
13     print(" - Momentos de inércia do cilindro (kg m^2):")
14     print(' I_cx = {:.f}; I_cy = {:.f}; I_cz = {:.f}'.format(I_cilx,I_cily,I_cilz))
15
16     #Momentos de inercia da haste
17     I_hx = (1/12)*mh*lh**2
18     I_hy = 0
19     I_hz = I_hx
20
21     print(" - Momentos de inércia da haste (kg m^2):")
22     print(' I_hx = {:.f}; I_hy = {:.f}; I_hz = {:.f}'.format(I_hx,I_hy,I_hz))
23
24     #Calculando os eixos
25     Ix = I_cilx + I_hx
26     Iy = I_cily + I_hy
27     Iz = I_cilz + I_hz
28
29     #Verificando o eixo estável/instável (considerando que não há dissipação de energia)
30     if Ix > Iy and Ix > Iz:      #Verificando se for é o maior valor
31         x = 'estável'
32     else:                      #Verificar se é o valor intermediario ou o menor
33         if (Ix > Iy and Ix < Iz) or (Ix > Iz and Ix < Iy):
34             x = 'instável'
35         else:
36             x = ' - '
37     if Iy > Ix and Iy > Iz:      #Verificando se for é o maior valor
38         y = 'estável'
39     else:                      #Verificar se é o valor intermediario ou o menor
40         if (Iy > Ix and Iy < Iz) or (Iy > Iz and Iy < Ix):
41             y = 'instável'
42         else:
43             y = ' - '
44     if Iz > Ix and Iz > Iy:      #Verificando se for é o maior valor
45         z = 'estável'
46     else:                      #Verificar se é o valor intermediario ou o menor

```

```

47     if (Iz > Ix and Iz < Iy) or (Iz > Iy and Iz < Ix):
48         z = 'instável'
49     else:
50         z = ' - '
51     return [Ix,x],[Iy,y],[Iz,z]

```

### III. RESULTADOS

Abaixo se encontra o código para resolver a atividade 6a (reprodução do exemplo de aula) e seu respectivo resultado (intermediário e final).

```

1  #Exemplo 2 - Torque livre
2  mb = 300. #Massa do cilindro B (kg)
3  mh = 30.  #Massa da haste (kg)
4  rb = 0.5  #Raio do cilindro B (m)
5  lb = 2*0.5 #Comprimento do cilindro (m)
6  lh = 2.0  #Comprimento da haste (m)
7
8  x,y,z = torque_livre(mb,mh,rb,lb,lh)
9
10 print("\nResultado final - Atividade 6a:")
11 print(' Ix = {:f} --> {}'.format(x[0],x[1]))
12 print(' Iy = {:f} --> {}'.format(y[0],y[1]))
13 print(' Iz = {:f} --> {}'.format(z[0],z[1]))

```

```

Resultados intermediarios:
- Momentos de inércia do cilindro (kg m^2):
I_cx = 43.750000; I_cy = 43.750000; I_cz = 37.500000
- Momentos de inércia da haste (kg m^2):
I_hx = 10.000000; I_hy = 0.000000; I_hz = 10.000000

Resultado final - Atividade 6a:
Ix = 53.750000 kg m^2 --> estável
Iy = 43.750000 kg m^2 --> -
Iz = 47.500000 kg m^2 --> instável

```

Figura 1: Saídas da atividade 6a.

Abaixo se encontra o código para resolver a atividade 6b (cilindro+haste similar ao do exercício anterior, mas com dimensões diferentes) e seu respectivo resultado (intermediário e final).

```

1  #Atividade 6b - Torque livre
2  mb = 400.  #Massa do cilindro (kg)
3  mh = 60.   #Massa da haste (kg)
4  rb = 0.75  #Raio do cilindro (m)
5  lb = 2*0.5 #Comprimento do cilindro (m)
6  lh = 4.    #Comprimento da haste (m)
7
8  x,y,z = torque_livre(mb,mh,rb,lb,lh)
9
10 print("\nResultado final - Atividade 6b:")
11 print(' Ix = {:.f} --> {}'.format(x[0],x[1]))
12 print(' Iy = {:.f} --> {}'.format(y[0],y[1]))
13 print(' Iz = {:.f} --> {}'.format(z[0],z[1]))

```

```

Resultados intermediarios:
- Momentos de inércia do cilindro (kg m^2):
I_cx = 89.583333; I_cy = 89.583333; I_cz = 112.500000
- Momentos de inércia da haste (kg m^2):
I_hx = 80.000000; I_hy = 0.000000; I_hz = 80.000000

Resultado final - Atividade 6b:
Ix = 169.583333 kg m^2 --> instável
Iy = 89.583333 kg m^2 --> -
Iz = 192.500000 kg m^2 --> estável
>>> _

```

Figura 2: Saídas da atividade 6b.

#### IV. CONCLUSÃO

Como a reprodução do exemplo apresentado em aula foi bem sucedida, o programa está funcionando como o esperado.