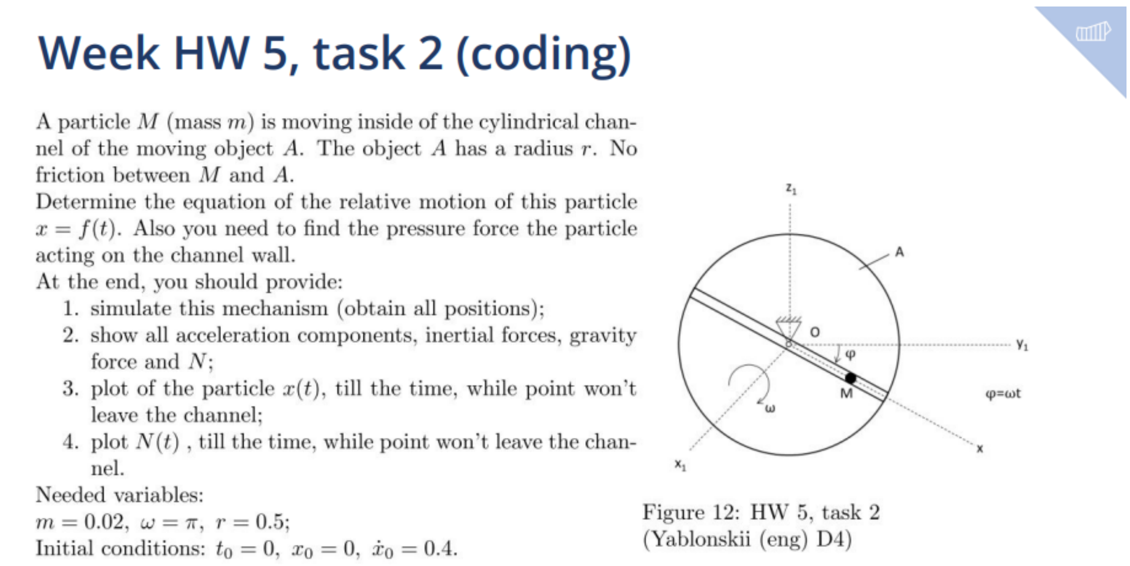
task_2

October 6, 2022

[1]:
```python
import IPython.display
import sympy as sp
from sympy.vector import CoordSys3D
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from sympy import init_printing,latex
sp.init_printing(use_latex=True)
```



Week HW 5, task 2 (coding)

A particle M (mass m) is moving inside of the cylindrical channel of the moving object A. The object A has a radius r. No friction between M and A.
Determine the equation of the relative motion of this particle $x = f(t)$. Also you need to find the pressure force the particle acting on the channel wall.
At the end, you should provide:

1. simulate this mechanism (obtain all positions);
2. show all acceleration components, inertial forces, gravity force and N;
3. plot of the particle $x(t)$, till the time, while point won't leave the channel;
4. plot $N(t)$, till the time, while point won't leave the channel.

Needed variables:
$m = 0.02,\ \omega = \pi,\ r = 0.5$;
Initial conditions: $t_0 = 0,\ x_0 = 0,\ \dot{x}_0 = 0.4$.

Figure 12: HW 5, task 2 (Yablonskii (eng) D4)

1 RO:

The system consists of two bodies: A (cylinder with a hollow tube) and M - particle # Motion: M - translatory motion A - rotation

2 Condition:

I want to describe the motion, where the new coordinate system is: $\begin{bmatrix} x \\ y \\ x \end{bmatrix} = R_\phi \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$

$$\begin{array}{ccc} initial & '2' & final \\ y_0 = 0 & y_2 = r & y_f - ? \\ \dot{y_0} = 0.4 & \dot{y_2} - ? & \dot{y_f} - ? \\ t_0 = 0 & t_2 - ? & t_f - ? \end{array}$$

Kinematics analysis: $\epsilon = 0$

$$\vec{a}_{tr} = \vec{\omega} \times (\vec{\omega} \times \vec{OM}) + \vec{\epsilon} \times \vec{OM} = \vec{\omega} \times (\vec{\omega} \times \vec{OM})$$

$$\vec{a}_{cor} = 2\vec{\omega} \times \vec{V}_{rel}$$

3 Force analysis:

$$ma_{rel} = \sum F + \Phi_{tr} + \Phi_{cor}$$

There is no friction between M and A, so the $\sum F = \vec{N} + m\vec{g}$. $\Phi_{tr} = -m\vec{a}_{tr}$ $\Phi_{cor} = -m\vec{a}_{cor}$

4 Solution:

$$ma_{rel} = \vec{N} + m\vec{g} - m\vec{a}_{tr} + -m\vec{a}_{cor}$$

Set initial conditions and constants

[2]:
```
# constants
r = 0.5
m = 0.02
omega = [[np.pi], [0], [0]]

# initial conditions
t_0 = 0
x_0 = 0
dx_0 = 0.4
phi_0 = [[0], [0], [0]]
```

[5]:
```
# init symbols for parameters
t_s, y_s, dy_s, m_s, r_s = sp.symbols("t, y, y', m, r")
```

4.1 Kinematics solution:

4.1.1 Set the coordinates of point M and O

There are two coordinate systems: Transport

$ M_{tr} (0, 0, x)$

Static $ M = R_ M_{tr}$

[6]:
```
phi_s = sp.pi * t_s

M_tr = sp.Matrix([[0], [y_s], [0]])
```

```python
rot_matrix = sp.Matrix([[1, 0, 0],
                        [0, sp.cos(phi_s), -sp.sin(phi_s)],
                        [0, sp.sin(phi_s), sp.cos(phi_s)]])

M = rot_matrix * M_tr
print(f'Coordinates of point M is static coordinate system: {M[:]}')
```

```
Coordinates of point M is static coordinate system: [0, y*cos(pi*t),
y*sin(pi*t)]
```

4.1.2 Find Velocity of the point M:

$$V = V_{rel} + V_{tr} = \dot{\vec{x}} + \vec{\omega} \times \vec{OM}$$

[7]:
```python
# relative velocity of the point M in the coord system of the body A
v_rel_trans_coord_sys = sp.Matrix([[0], [dy_s], [0]])
print(f'Relative velocity in transport coord system: Vm rel=␣
 ↪{v_rel_trans_coord_sys[:]}')

# relative velocity of the point M in static coord system
v_rel_s = rot_matrix * v_rel_trans_coord_sys
# transport velocity
OM = M
omega_s = sp.Matrix([[sp.pi], [0], [0]])
v_tr_s = omega_s.cross(OM)

# full velocity
v_s = v_tr_s + v_rel_s

print(f'Full velocity: Vm = {v_s[:]}')
```

```
Relative velocity in transport coord system: Vm rel= [0, y', 0]
Full velocity: Vm = [0, -pi*y*sin(pi*t) + y'*cos(pi*t), pi*y*cos(pi*t) +
y'*sin(pi*t)]
```

4.1.3 Find transport and coriolis acceleration of the point M:

$$\vec{a}_{tr} = \vec{\omega} \times (\vec{\omega} \times \vec{OM})$$

$$\vec{a}_{cor} = 2\vec{\omega} \times \vec{V}_{rel}$$

[8]:
```python
a_tr_s = omega_s.cross(omega_s.cross(OM))
a_cor_s = 2 * omega_s.cross(v_rel_s)




print(f'Coriolis acceleration: a_cor={a_cor_s[:]}')
print(f'Transport acceleration: a_tr={a_tr_s[:]}')
```

```
Coriolis acceleration: a_cor=[0, -2*pi*y'*sin(pi*t), 2*pi*y'*cos(pi*t)]
Transport acceleration: a_tr=[0, -pi**2*y*cos(pi*t), -pi**2*y*sin(pi*t)]
```

P:

```
[9]: g_s = sp.Matrix([[0], [0], [9.8]])
     P = m * g_s
     P_A_cs = rot_matrix * P
     print(f'Weight of M in static coordinate system: P = {P}')
     print(f'Weight of M in coordinate system of body A: P = {P_A_cs}')
```

```
Weight of M in static coordinate system: P = Matrix([[0], [0],
[0.196000000000000]])
Weight of M in coordinate system of body A: P = Matrix([[0], [-0.196*sin(pi*t)],
[0.196*cos(pi*t)]])
```

4.1.4 Let's calculate the vectors and projections on axis Ox for all forces:

N:

```
[10]: P_magnitude = sp.sqrt(P[0] ** 2 + P[1] ** 2 + P[2] ** 2)

      # normal reaction force in static coord system
      normal_reaction_force = sp.cos(phi_s) * P_A_cs

      normal_reaction_force_x = sp.Matrix([[0],[0],[P_A_cs[2]]])
      print(f'Normal reaction force in static coordinate system: N =␣
       ↪{normal_reaction_force}')
      print(f'Normal reaction force in coordinate system of body A: N =␣
       ↪{normal_reaction_force_x}')
```

```
Normal reaction force in static coordinate system: N = Matrix([[0],
[-0.196*sin(pi*t)*cos(pi*t)], [0.196*cos(pi*t)**2]])
Normal reaction force in coordinate system of body A: N = Matrix([[0], [0],
[0.196*cos(pi*t)]])
```

$$y : \ddot{y} = -a_{try} - a_{cor} + g * sin(\phi)$$

$$y : \ddot{y} = y\pi^2 sin(\omega t)cos(\omega t) + g * sin(\omega t)$$

Laplace transform:

$$L(\ddot{y}) = s^2\overline{y} - sy(0) - \dot{y}(0)$$

$$L(y) = \overline{y}$$

$$L(\pi^2 sin(\omega t)cos(\omega t)) = \pi^2 * \frac{\omega}{s^2+\omega^2} * \frac{s}{s^2+\omega^2}$$

$$L(g * sin(\omega t)) = g * \frac{\omega}{s^2+\omega^2}$$

$$\begin{cases} s^2\overline{y} - \overline{y} * \pi^2 * \frac{\omega}{s^2+\omega^2} * \frac{s}{s^2+\omega^2} - sy(0) - \dot{y}(0) = g * \frac{\omega}{s^2+\omega^2} \\ \dot{y}(0) = 0.4 \\ y(0) = 0 \end{cases}$$

$$\overline{y}(s^2 - \pi^2 * \frac{\omega}{s^2+\omega^2} * \frac{s}{s^2+\omega^2}) - s * 0 - 0.4 = g * \frac{\omega}{s^2+\omega^2}$$

$$\overline{y} = \frac{g * \frac{\omega}{s^2+\omega^2} + 0.4}{(s^2 - \pi^2 * \frac{\omega}{s^2+\omega^2} * \frac{s}{s^2+\omega^2})}$$

$$\overline{y} = \frac{g * \omega}{s^2(s^2+\omega^2) - \pi^2 * \omega * s} + 0.4 * \frac{s^2+\omega^2}{s^2 * (s^2+\omega^2) - \pi^2 * \omega * s}$$

We can find the results only numerically, or(what is more possible) I screwed up

Sorry, just tired

[11]:
```
ddy = -a_tr_s[1] - a_cor_s[1] + P_A_cs[1]
```