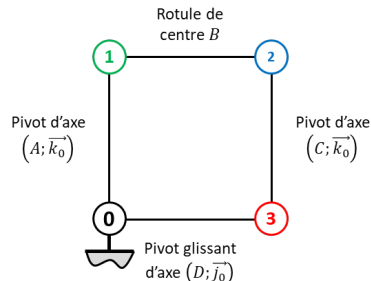


## Système bielle manivelle ★★

C2-06

**Question 1** Tracer le graphe des liaisons.



**Question 2** Exprimer  $\lambda(t)$  en fonction de  $\theta(t)$ .

En réalisant une fermeture géométrique, on obtient  $\vec{AB} + \vec{BC} + \vec{CA} = \vec{0} \Leftrightarrow R\vec{i}_1 - L\vec{i}_2 - \lambda(t)\vec{j}_0 = \vec{0}$ . On projette alors cette expression dans  $\mathcal{R}_0$  :

$$\begin{cases} R \cos \theta(t) - L \cos \varphi(t) = 0 \\ R \sin \theta(t) - L \sin \varphi(t) - \lambda(t) = 0 \end{cases} .$$

On cherche à éliminer  $\varphi(t)$  :

$$\begin{cases} R \cos \theta(t) = L \cos \varphi(t) \\ R \sin \theta(t) - \lambda(t) = L \sin \varphi(t) \end{cases} .$$

En élevant au carré, on a donc

$$\begin{cases} R^2 \cos^2 \theta(t) = L^2 \cos^2 \varphi(t) \\ (R \sin \theta(t) - \lambda(t))^2 = L^2 \sin^2 \varphi(t) \end{cases} .$$

En conséquence,  $R^2 \cos^2 \theta(t) + (R \sin \theta(t) - \lambda(t))^2 = L^2$  et

$$(R \sin \theta(t) - \lambda(t))^2 = L^2 - R^2 \cos^2 \theta(t) \Rightarrow \lambda(t) = \pm \sqrt{L^2 - R^2 \cos^2 \theta(t)} + R \sin \theta(t) .$$

**Question 3** Exprimer  $\dot{\lambda}(t)$  en fonction de  $\dot{\theta}(t)$ .

$$\dot{\lambda}(t) = \pm \left( \frac{R^2 \dot{\theta}(t) \cos \theta(t) \sin \theta(t)}{\sqrt{L^2 - R^2 \cos^2 \theta(t)}} \right) + \dot{\theta}(t) R \cos \theta(t)$$

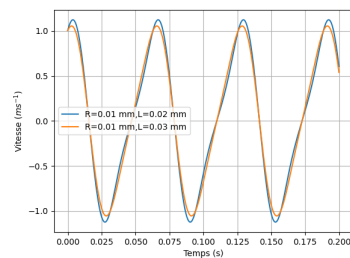
**Question 4** En utilisant Python, tracer la vitesse du piston en fonction du temps. La fréquence de rotation est  $\dot{\theta}(t) = 100 \text{ rad s}^{-1}$ , on prendra  $R = 10 \text{ mm}$  et  $L = 20 \text{ mm}$  puis  $L = 30 \text{ mm}$ .

```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3
4 """12_BielleManivelle.py"""
5 __author__ = "Xavier Pessoles"
6 __email__ = "xpessoles.ptsi@free.fr"
7
8 import numpy as np
9 import matplotlib.pyplot as plt
10 import math as m
11 from scipy.optimize import newton
12 from scipy.optimize import fsolve
```

```

13
14 R = 0.01 # m
15 L = 0.03 # m
16 w = 100
17 def calc_lambda(theta):
18     #res = R*np.sin(theta)
19     #print(L*L-R*R*np.cos(theta)*np.cos(theta))
20     #res = res + np.sqrt(L*L-R*R*np.cos(theta)*np.cos(theta))
21     res = np.sqrt(L*L-R*R*np.cos(theta)*np.cos(theta))+R*np.sin(theta)
22     return res
23
24 def plot_lambda():
25     les_theta=np.linspace(-2*np.pi,2*np.pi,1000)
26     les_l = [calc_lambda(x) for x in les_theta]
27     plt.grid()
28     plt.xlabel("Temps (s)")
29     plt.ylabel("Vitesse ({m}s^{-1})")
30     plt.plot(les_theta,les_l,label=str("R=")+str(R)+" mm,"+str("L=")+str(L)+"
31             mm")
32     plt.legend()
33     plt.show()
34 plot_lambda()

```



**Question 5** En utilisant Python, tracer l'accélération du piston en fonction du temps en utilisant les mêmes valeurs que dans la question précédente. On utilisera une dérivation numérique.

