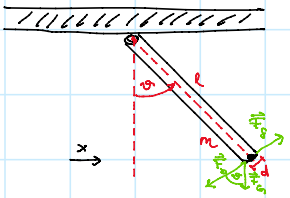
Rigid pendulum with friction - simulation

Sonntag, 28. November 2021 14:52

1) Set up



ϑ .. angle to rest position
l .. length of the pendulum
m .. mass of the pendulum
d .. diameter of the pendulum

- $\vec{F}_f$ .. Friction
- $\vec{F}_b$ .. Recoil force
- $\vec{F}_g$ .. Gravitation

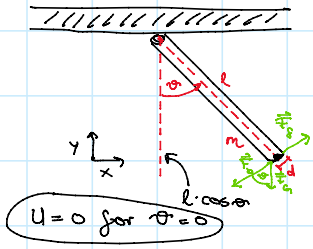
Assumptions: Friction is not negligible, pendulum is a perfect cylinder with diameter d, there are no other friction parameters than air friction from the cylinder

2) Theoretical description

→ generalized coordinate: ϑ

- $\dot{x}(t) = l \cdot \dot{\vartheta}(t)$ for the endpoint of the pendulum

$\Rightarrow T = \frac{1}{2} m \dot{x}^2 = \frac{1}{2} (l \cdot \dot{\vartheta})^2 = \frac{1}{2} l^2 \dot{\vartheta}^2$



- $y = l - l \cdot \cos\vartheta = l(1 - \cos\vartheta)$

$\Rightarrow U = m \cdot g \cdot y = m \cdot g \cdot l \cdot (1 - \cos\vartheta)$

- Generalized displacement function (friction):

$\vec{F}_{air} = \frac{1}{2} A c_w \rho_{air} \dot{x}^2$

with $A = \pi \cdot d \cdot l$, $c_w = 1{,}2$ (source: Wikipedia)

$\rho_{air} = 1{,}204 \frac{kg}{m^3}$ ($T = 20°$, source: Wikipedia)

$\dot{x} = \dot{\vartheta} l$

↳ $D = \frac{1}{n+1} \sum_j C_j v_j^{n+1}$ (C ← friction constant; n .. power of velocity dependence; $\sum_j$ ↑ sum over all particles)

$j = 1$, $n = 2$

$\Rightarrow D = \frac{1}{3} C \dot{x}^3 = \frac{1}{3} \left(\frac{\pi \cdot d \cdot l}{2} c_w \rho_{air} \right) \dot{x}^3$

$= \frac{\pi d l}{6} c_w \rho_{air} \dot{x}^3 = \underbrace{\frac{\pi d l}{6} c_w \rho_{air} l^3}_{\hat{c}} \dot{\vartheta}^3$

↳ Generalized force: $Q = \frac{\partial}{\partial \dot{\vartheta}} (\hat{c} \dot{\vartheta}^3)$

$\Rightarrow Q = 3 \hat{c} \dot{\vartheta}^2 = \underbrace{\frac{1}{2} \pi d l c_w \rho_{air}}_{\tilde{c}} \dot{\vartheta}^2$

↳ Lagrangian:

$L = T - V = \frac{m}{2} l^2 \dot{\vartheta}^2 - m \cdot g \cdot l \cos\vartheta$

⇒ Euler-Lagrange-equation:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot\vartheta} - \frac{\partial L}{\partial \vartheta} = Q$$

$$\Leftrightarrow \frac{d}{dt}\left(ml^2\dot\vartheta\right) - \left(\frac{m}{2}l^2\dot\vartheta^2 + mgl\sin\vartheta\right) = \tilde c\,\dot\vartheta^2$$

$$\Leftrightarrow \; ml\ddot\vartheta - \frac{m}{2}l^2\dot\vartheta^2 + mgl\sin\vartheta = \tilde c\,\dot\vartheta^2$$

$$\Leftrightarrow \underbrace{ml\ddot\vartheta}_{:=\frac{n}{l}\lambda_1} + \underbrace{\left[\frac{m}{2}l^2 - \tilde c\right]}_{=\frac{m}{2}l^2 - \frac{n}{2}\pi d l c_w \rho_{air}}\dot\vartheta^2 + \underbrace{mgl}_{:=\frac{n}{l}\lambda_3}\sin\vartheta = 0$$

$$= \frac{l}{2}\left(ml - \pi d c_w \rho_{air}\right) := \frac{n}{l}\lambda_2$$

$$\Leftrightarrow \boxed{\lambda_1\ddot\vartheta + \lambda_2\dot\vartheta^2 + \lambda_3\sin\vartheta = 0 \quad (*)}$$

$$\left[\lambda_1 = m;\ \lambda_2 = \frac{1}{2}\left(l - \pi d c_w \rho_{air}\right);\ \lambda_3 = m\cdot g\right]$$ ← DGL of motion

3) Solving (*)

```
from sympy import *
import sympy as sp
import numpy as np

"""constants"""
a=1
b=1
c=1

"""function"""
t=sp.Symbol("t")
f=sp.Function("f")(t)

"""equation"""
dgl=Eq(a*f.diff(t,t)+b*np.square(f.diff(t))+c*sp.sin(f),0)
display(dgl)

dsolve(dgl,f)
```

$$\sin\left(f(t)\right) + \left(\frac{d}{dt}f(t)\right)^2 + \frac{d^2}{dt^2}f(t) = 0$$

```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
<ipython-input-8-b78228b925ab> in <module>
     16 display(dgl)
     17
---> 18 dsolve(dgl,f)

~\Anaconda3\lib\site-packages\sympy\solvers\ode.py in dsolve(eq, func, hint, simplify, ics, xi, eta, x0, n, **kwargs)
    644         hints = _desolve(eq, func=func,
    645             hint=hint, simplify=True, xi=xi, eta=eta, type='ode', ics=ics,
--> 646             x0=x0, n=n, **kwargs)
    647         eq = hints.pop('eq', eq)
    648         all_ = hints.pop('all', False)

~\Anaconda3\lib\site-packages\sympy\solvers\deutils.py in _desolve(eq, func, hint, ics, simplify, **kwargs)
    242                 str(eq) + " is not a solvable differential equation in " + str(func))
    243         else:
--> 244             raise NotImplementedError(dummy + "solve" + ": Cannot solve " + str(eq))
    245     if hint == 'default':
    246         return _desolve(eq, func, ics=ics, hint=hints['default'], simplify=simplify,

NotImplementedError: solve: Cannot solve sin(f(t)) + Derivative(f(t), t)**2 + Derivative(f(t), (t, 2))
```