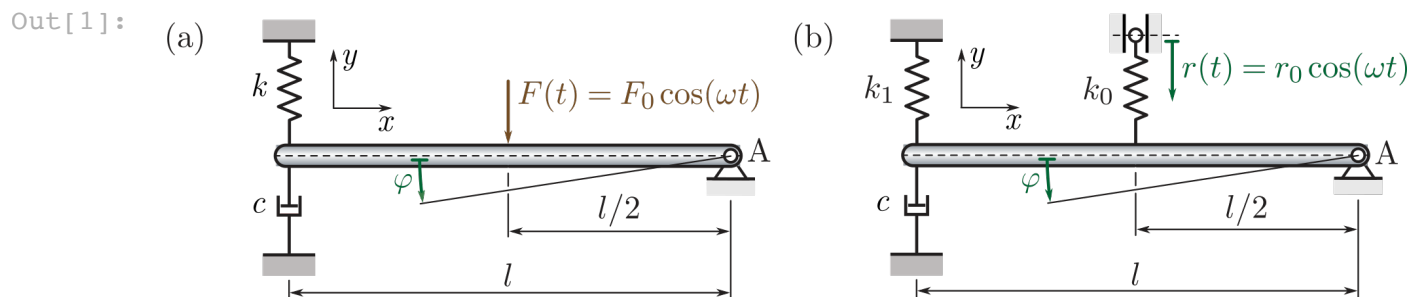


6. Gyakorlat - 1 DoF gerjesztett, csillapított lengőkar

2021.03.15

Feladat:

```
In [1]: from IPython.display import Image  
Image(filename='gyak6_1.png',width=900)
```



A mellékelt ábrán egy lengőkar látható, mely egy l hosszúságú és m tömegű rúddal van modellezve. Ez csak az A csukló körüli elfordulásra képes. Két eset van megkülönböztetve: az (a) ábra egy harmonikus erőgerjesztést mutat, míg a (b) ábrán egy harmoikus útgerjesztés történik a k_0 rugómerevségű rugón keresztül. A rúd mindkét esetben egy rugón (k , valamint a (b) ábrán k_1) és egy csillapításon (c) keresztül van a környezethez rögzítve. Ezen 1 szabadságfokú lengőkar mozgását a φ általánosított koordináta segítségével írjuk le, mely a vízszintes síktól van mérve. A szerkezetet a vízszintes síkban vizsgáljuk, valamint az egyensúlyi helyzete a $\varphi = 0$ szögelforduláshoz tartozik. Ekkor az (a) összeállítás k rugóállandójú rugója, valamint a (b) k_1 rugóállandójú rugója erőmentes (nyújtatlan).

Adatok:

$m = 3 \text{ kg}$	$l = 1 \text{ m}$	$\omega = 30 \text{ rad/s}$
$k = 400 \text{ N/m}$	$F_0 = 10 \text{ N}$	$c = 28 \text{ Ns/m}$
$k_0 = 1000 \text{ N/m}$	$r_0 = 0.01 \text{ m}$	$k_1 = 150 \text{ N/m}$

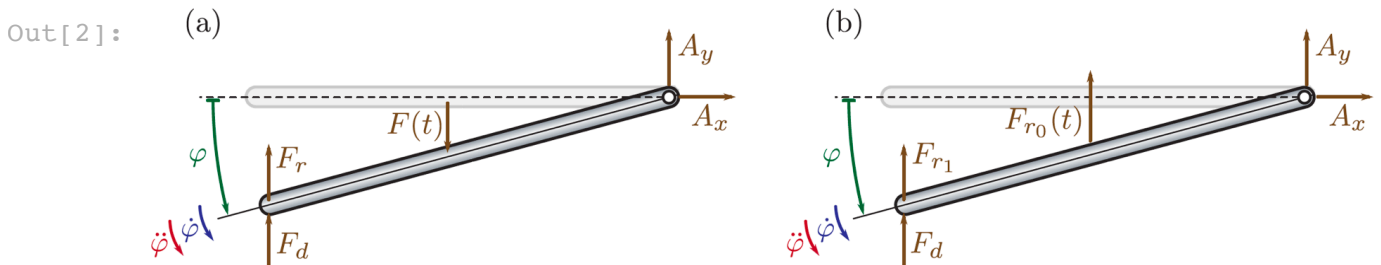
Részfeladatok:

1. Vezesse le a mozgásegyenletet mindkét modell esetén, és számítsa ki a csillapított, illetve a csillapítatlan rendszer sajátkörfrekvenciáját, csillapítási tényezőjét és statikus deformációját!
2. Ábrázolja a rezonanciagörbét és a fáziskésés diagrammot mindkét esetben!
3. Határozza meg a $\varphi(t)$ mozgástörvényt, amennyiben a kezdeti értékek $\varphi(t = 0) = \varphi_0 = 0,015 \text{ rad}$ és $\dot{\varphi}(t = 0) = 0 \text{ rad/s}$!

Megoldás:

1. Feladat

```
In [2]: from IPython.display import Image
Image(filename='gyak6_2.png',width=900)
```



(Mivel a szabadtest ábra felrajzolása és a dinamika alaptételének alkalmazása nem programozási feladat, így ez itt nincs részletezve.)

A következő mozgásegyenletek írhatóak fel (a szabadtest ábrák jelöléseit használva):

```
In [3]: import sympy as sp
from IPython.display import Math # hogy tudjunk LaTeX szöveget kiírni
sp.init_printing()

t, m, theta_A, l, Fr, Fr1, Fd = sp.symbols('t, m, theta_A, l, F_r, F_r1, F_d')
F0, k, c, k0, k1, omega, r0, zeta, f0 = sp.symbols('F_0, k, c, k0, k1, omega, r0, zeta, f0')

phi = sp.Function('phi')(t)
F = sp.Function('F')(t)
Fr0 = sp.Function('F_r0')(t)

# Készítsünk behelyettesítési listát az adatok alapján, SI-ben:
adatok = [(m, 3), (l, 1), (omega, 30), (k, 400), (F0, 10),
           (c, 28), (k0, 1000), (r0, 0.01), (k1, 150)]

mozgegy_a_eq = sp.Eq(theta_A*sp.diff(phi,t,2), F * l/2 - Fr*l - Fd*l) # az (a) a
mozgegy_b_eq = sp.Eq(theta_A*sp.diff(phi,t,2), -Fr0 * l/2 - Fr1*l - Fd*l) # a (b)
```

```
In [4]: display(mozgegy_a_eq, mozgegy_b_eq)
```

$$\theta_A \frac{d^2}{dt^2} \varphi(t) = -F_d l - F_r l + \frac{l F(t)}{2}$$

$$\theta_A \frac{d^2}{dt^2} \varphi(t) = -F_d l - F_{r1} l - \frac{l F_{r0}(t)}{2}$$

```
In [5]: # Linearizáljuk a rugóknban/csillapításokban ébredő erők kifejezését
```

```
Fr_kif = k*l*phi # közelítés; kis kitérések esetén
Fd_kif = c*l*sp.diff(phi,t) # közelítés; kis kitérések esetén
Fr1_kif = k1*l*phi # közelítés; kis kitérések esetén
```

```
In [6]: # Vizsgáljuk a gerjesztés alakját.
```

```
# harmonikus erőgerjesztés esetében:
F_kif = F0*sp.cos(omega*t)
```

```
# harmonikus útgerjesztés esetében:
Fr0_kif = k0 * (l/2*phi - r0*sp.cos(omega*t)) # linearizált alak -> kis kitérések
```

```
In [7]: # Steiner-tétel használatával határozzuk meg a rúd tehetetlenségi nyomatékát
# 'z' tengely körüli elfordulás esetén az 'A' pontra.
```

```
theta_A_kif = sp.Rational(1,12) * m * l**2 + m * (l/2)**2 # sp.Rational(számláló,nevező)
```

```
In [8]: # Készítsünk behelyettesítési listát a fenti kifejezésekre:
```

```
kif_lista = [(theta_A,theta_A_kif), (Fr0,Fr0_kif), (F,F_kif),
             (Fr1,Fr1_kif), (Fd,Fd_kif), (Fr,Fr_kif)]
```

```
mozgegy_a_eq = mozgegy_a_eq.subs(kif_lista)
mozgegy_b_eq = mozgegy_b_eq.subs(kif_lista)
display(mozgegy_a_eq,mozgegy_b_eq)
```

$$\frac{l^2 m \frac{d^2}{dt^2} \varphi(t)}{3} = \frac{F_0 l \cos(t\omega)}{2} - cl^2 \frac{d}{dt} \varphi(t) - kl^2 \varphi(t)$$

$$\frac{l^2 m \frac{d^2}{dt^2} \varphi(t)}{3} = -cl^2 \frac{d}{dt} \varphi(t) - \frac{k_0 l \left(\frac{l\varphi(t)}{2} - r_0 \cos(t\omega) \right)}{2} - k_1 l^2 \varphi(t)$$

```
In [9]: # A mozgásegyenletek most ún. sympy Equality objektumok,
display(type(mozgegy_a_eq))
display(mozgegy_a_eq.is_Equality)

# azaz relációk két kifejezés között (az egyenlet bal és jobb oldala).
# Ennek a használata eddig kényelmes volt, de az egyenletrendezés itt nem t
# nem lehet pl. a bal és a jobb oldalt is megszorozni egy számmal/kifejezé
# Térjünk át a másik reprezentációra, ahol csak a bal oldallal foglalkozunk
# oldalt 0-vá tesszük.
```

```
sympy.core.relational.Equality
True
```

```
In [10]: mozgegy_a = mozgegy_a_eq.rhs - mozgegy_a_eq.lhs
mozgegy_b = mozgegy_b_eq.rhs - mozgegy_b_eq.lhs

# osszunk le a főegyütthatóval
mozgegy_a = mozgegy_a / mozgegy_a.coeff(sp.diff(φ,t,2))
mozgegy_b = mozgegy_b / mozgegy_b.coeff(sp.diff(φ,t,2))

mozgegy_a = mozgegy_a.expand()
mozgegy_b = mozgegy_b.expand()

display(mozgegy_a, mozgegy_b) # egy oldalra rendezve!
```

$$-\frac{3F_0 \cos(t\omega)}{2lm} + \frac{3c \frac{d}{dt} \varphi(t)}{m} + \frac{3k\varphi(t)}{m} + \frac{d^2}{dt^2} \varphi(t)$$

$$\frac{3c \frac{d}{dt} \varphi(t)}{m} + \frac{3k_0 \varphi(t)}{4m} - \frac{3k_0 r_0 \cos(t\omega)}{2lm} + \frac{3k_1 \varphi(t)}{m} + \frac{d^2}{dt^2} \varphi(t)$$

```
In [11]: ω_nat_a = sp.sqrt(mozgegy_a.coeff(φ))
ω_nat_b = sp.sqrt(mozgegy_b.coeff(φ))

# φ első deriváltjának együtthatójából fejezhetjük ki a rel. csillapítási
φd_coeff_a = sp.Eq(mozgegy_a.coeff(sp.diff(φ)), 2*ζ*ω_nat_a)
φd_coeff_b = sp.Eq(mozgegy_b.coeff(sp.diff(φ)), 2*ζ*ω_nat_b)

ζ_a = sp.solve(φd_coeff_a, ζ)[0]
ζ_b = sp.solve(φd_coeff_b, ζ)[0]

ω_damp_a = ω_nat_a*sp.sqrt(1-ζ_a**2)
ω_damp_b = ω_nat_b*sp.sqrt(1-ζ_b**2)
```

```
In [12]: # Továbbá tudjuk, hogy a koszinuszos tag együtthatójából
# számolható a statikus deformáció. Ám az itt negatív együtthatóval
# szerepel, mert alapból az egyenlet jobb oldalán a helye, de most át lett l

cos_coeff_a = sp.Eq(-mozgegy_a.coeff(sp.cos(ω*t)), f0*ω_nat_a**2)
f0_a = sp.solve(cos_coeff_a, f0)[0]

cos_coeff_b = sp.Eq(-mozgegy_b.coeff(sp.cos(ω*t)), f0*ω_nat_b**2)
f0_b = sp.solve(cos_coeff_b, f0)[0]
```

```
In [13]: # Helyettesítsünk be:

symb_list = [ω_nat_a, ω_nat_b, ζ_a, ζ_b, ω_damp_a, ω_damp_b, f0_a, f0_b]

num_eredmenyek = [elem.subs(adatok) for elem in symb_list] # list comprehension
num_eredmenyek

# Látszik, hogy ugyanazok lettek az eredmények az (a) és (b) esetben.
```

```
Out[13]: [20, 20, 7/10, 7/10, 2√51, 2√51, 1/80, 0.0125]
```

```
In [14]: display(Math('\omega_n = {} \ \text{{rad/s}}'.format(num_eredmenyek[0])))
display(Math('\zeta = {} \ [1]'.format(num_eredmenyek[2].evalf(2))))
display(Math('\omega_d = {} \ \text{{rad/s}}'.format(num_eredmenyek[4].evalf(2))))
display(Math('f_0 = {} \ \text{{rad}}'.format(num_eredmenyek[7].evalf(3))))
```

$$\omega_n = 20 \text{ rad/s}$$

$$\zeta = 0.70 [1]$$

$$\omega_d = 14.28 \text{ rad/s}$$

$$f_0 = 0.0125 \text{ rad}$$

2. Feladat

```

In [15]: # A rezonanciagörbe és a fáziskésés diagram:

import matplotlib.pyplot as plt
from numpy import linspace

plt.subplots(figsize=(9, 9)) # beállítjuk a 'vászon' méretét

λ = sp.Symbol('λ')

# 2 koordináta rendszerre fogunk plottolni. Ezek elhelyezkedése egy rácson
# rácsnak 2 sora lesz, és 1 oszlopa. Azt, hogy éppen melyikre akarunk rajzolni
# a 3. paraméter adja meg: `subplot(sorok_száma, oszlopok_száma, kiválasztott_axis)`
ax1 = plt.subplot(2,1,1) # Tehát ő lesz az egyik krsz,
ax2 = plt.subplot(2,1,2) # és ő a másik.

ζ_list = [1E-5, 0.3, 0.5, 0.7] # ezen listaelemeken iterál végig a `for` ciklus

for ζ in ζ_list: # végigiterálunk a rel. csill. tényezőkön

    N = 1 / sp.sqrt( (1 - λ**2)**2 + 4 * ζ**2 * λ**2 )
    θ = sp.atan2((1-λ**2), 2*ζ*λ) + sp.pi/2

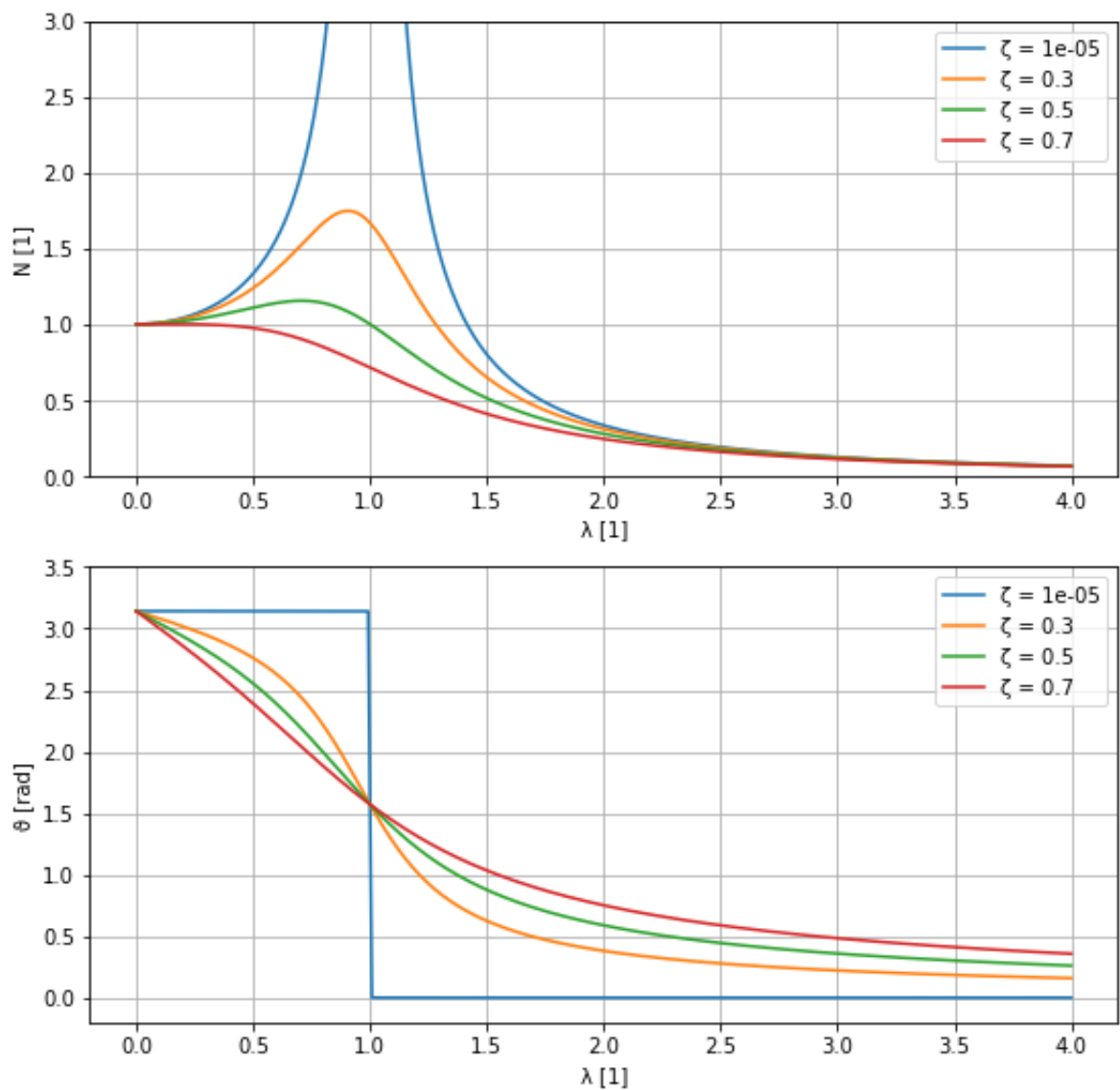
    λ_vals = linspace(0,4,501) # az x tengelyen felvesszünk 101 osztási pontot
    # mindegyikre kiszámoljuk a megfelelő y értéket
    N_vals = [N.subs(λ, λ_item) for λ_item in λ_vals] # ő lesz a felső krsz
    θ_vals = [θ.subs(λ, λ_item).evalf(4) for λ_item in λ_vals] # ő az alsó

    ax1.plot(λ_vals, N_vals) # itt a felső krsz-re rajzolunk
    ax1.set_xlabel("λ [1]")
    ax1.set_ylabel("N [1]")
    ax1.set_ylim([0,3])
    ax2.plot(λ_vals, θ_vals) # itt az alsóra
    ax2.set_xlabel("λ [1]")
    ax2.set_ylabel("θ [rad]")
    ax2.set_ylim([-0.2,3.5])
# itt véget ért a `for`, mert már nics indentálva

ax1.grid() # rácsvonalak
ax2.grid()
ax1.legend(['ζ = {}'.format(elem) for elem in ζ_list]) # jelmagyarázat
ax2.legend(['ζ = {}'.format(elem) for elem in ζ_list])

plt.show()

```



3. Feladat

```
In [16]: kezdeti_ert = {phi.subs(t,0): 0.015, phi.diff(t).subs(t,0): 0}

phi_mozgtorv = sp.dsolve(mozgegy_a.subs(adatok),phi,ics=kezdeti_ert)
display(phi_mozgtorv.evalf(4))

# Egyszerű megoldást nem találtam, amivel a gerjesztő harmonikus tagokat
# amplitúdó+fáziskésés alakra lehet hozni, csak a kézi módszert, ami a gyal

Phi = sp.sqrt(phi_mozgtorv.rhs.coeff(sp.sin(30*t))*2
              + phi_mozgtorv.rhs.coeff(sp.cos(30*t))*2).evalf(4)

theta = sp.atan2(phi_mozgtorv.rhs.coeff(sp.sin(30*t)),
                 phi_mozgtorv.rhs.coeff(sp.cos(30*t)))

display(Math('\Phi = {} \text{{rad}}'.format(Phi)))
display(Math('\theta = {} \text{{rad}}'.format(theta.evalf(4))))

# Mellesleg theta leolvasható a fáziskésés diagramról, valamint Phi kiszámítható
# a rezonanciagörbéről leolvasott 'N' nagyítás, és a statikus deformáció sz
```

$$\varphi(t) = (0.008036 \sin(2\sqrt{51}t) + 0.01762 \cos(2\sqrt{51}t)) e^{-14t} + 0.004395 \sin(30t) -$$

$$\Phi = 0.005115 \text{ rad}$$

$$\vartheta = 2.108 \text{ rad}$$

Készítette:

Csuzdi Domonkos (Alkalmazott Mechanika Szakosztály) és
Juhos-Kiss Álmos (Alkalmazott Mechanika Szakosztály)
Takács Dénes (BME MM) ábrái és Béri Bence (BME MM)
kidolgozása alapján.

Hibák, javaslatok:
amsz.bme@gmail.com
csuzdi02@gmail.com
almosjuhoskiss@gmail.com

2021.03.15