Elastostatic model for platform supports Rev A

—

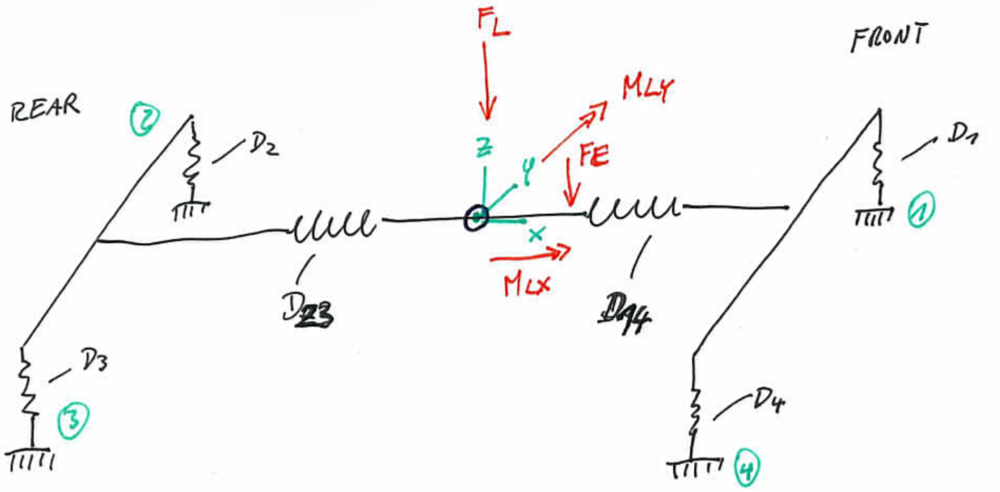
Author: LTro

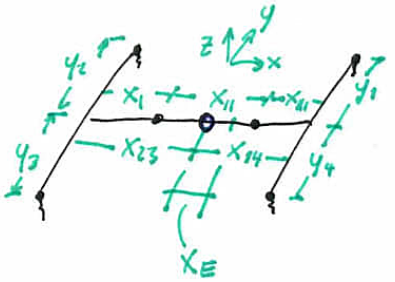
Modification Date: Tuesday 13th December 2022 11:29:23

—

Problem statement

A system of three rigid bodies is connected by torsion springs (spring rates D_{23} and D_{14}) and compression springs (spring rates $D_1 ... D_4$). The torsion springs only allow the bodies to rotate relative to each other, the compression springs only allow translation. Between the torsion springs, moments are introduced about the x- and y-axis as well as a compressive force in the z-direction.
The spring forces $F_1 ... F_4$ and the torsional moments in the torsion springs T_{14} and T_{23} need to be determined.





LEGEND

Rigid Body

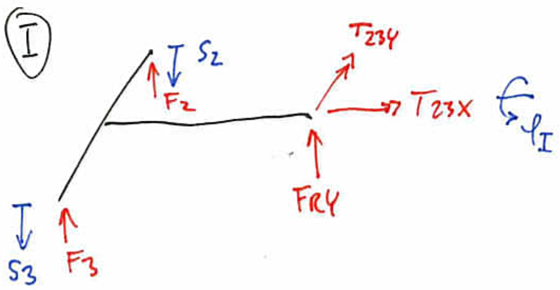
Torsion Spring (rotation only)

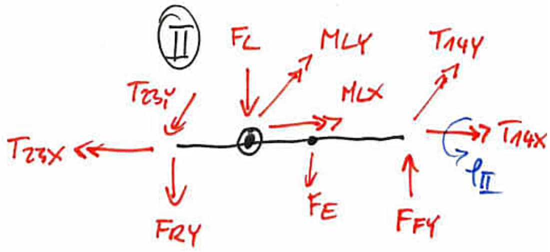
Compression Spring (translation only)

Loads, Reactions

Geometry

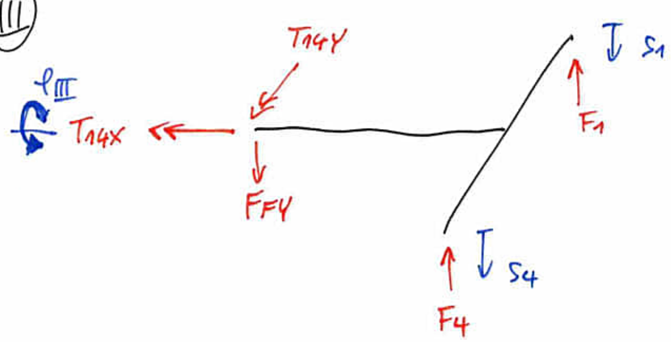
DOF





Σ: 10 Reactions

Σ: 7 DOF



Set up system of equations

Equilibrium equations for the rear T body (I)

$$\Sigma F_z = F_3 + F_2 + F_{Ry} = 0 \tag{1}$$
$$\Sigma M_x = F_2 \cdot y_2 - F_3 \cdot y_3 + T_{23x} = 0 \tag{2}$$
$$\Sigma M_y = (F_2 + F_3) \cdot x_i + T_{23y} = 0 \tag{3}$$

Equilibrium equations for the center body (II)

$$\Sigma F_z = -F_{Ry} + F_{Fy} - F_L - F_E = 0 \tag{4}$$
$$\Sigma M_x = T_{14x} + M_{Lx} - T_{23x} = 0 \tag{5}$$
$$\Sigma M_y = M_{Ly} + T_{14y} - T_{23y} - F_{Ry} \cdot (x_{23} - x_i) - F_{Fy} \cdot (x_{14} - x_{iii}) + F_E \cdot x_E = 0 \tag{6}$$

Equilibrium equations for the front T body (III)

$$\Sigma F_z = F_1 + F_4 - F_{Fy} = 0 \tag{7}$$
$$\Sigma M_x = F_1 \cdot y_1 - F_4 \cdot y_4 - T_{14x} = 0 \tag{8}$$
$$\Sigma M_y = -T_{14y} - (F_1 + F_4) \cdot x_{iii} = 0 \tag{9}$$

Spring equations (compression springs)

$$D_1 \cdot s_1 - F_1 = 0 \tag{10}$$
$$D_2 \cdot s_2 - F_2 = 0 \tag{11}$$
$$D_3 \cdot s_3 - F_3 = 0 \tag{12}$$
$$D_4 \cdot s_4 - F_4 = 0 \tag{13}$$

Spring equations (torsion springs)

$$(\phi_i - \phi_{ii}) \cdot D_{23} - T_{23x} = 0 \tag{14}$$
$$(\phi_{ii} - \phi_{iii}) \cdot D_{14} - T_{14x} = 0 \tag{15}$$

Kinematic constraints

$$\tan \phi_i = \phi_i = \frac{s_2 - s_3}{y_2 + y_3} \quad \text{(using small-angle approx.)} \tag{16}$$
$$\phi_{iii} = \frac{s_1 - s_4}{y_1 + y_4} \tag{17}$$

Solving (using python with sympy)

Please note: For the sake of clarity, some simplifications are made before outputting the results.
Support distances: $y_1 = y_2 = y_3 = y_4 = h$,
compression spring constants: $D_1 = D_2 = D_3 = D_4 = D_c$,
torsion spring constants: $D_{14} = D_{23} = D_t$

```
#!/usr/bin/env python
# coding: utf-8

from sympy import Eq, solve, symbols

def solve_model():
    # Known
    fl, mlx, mly, y1, y2, y3, y4, x1, x2, x3, x14, x23, d14, d23, d1, d2, d3, d4
= symbols('fl, mlx, mly, y1, y2, y3, y4, x1, x2, x3, x14, x23, d14, d23, d1, d2,
d3, d4')

    # Unknown
    f1, f2, f3, f4, t14x, t23x, t14y, t23y, s1, s2, s3, s4, phi1, phi2, phi3,
fry, ffy = symbols('f1, f2, f3, f4, t14x, t23x, t14y, t23y, s1, s2, s3, s4,
phi1, phi2, phi3, fry, ffy')

    # Rigid Body I
    eq1 = Eq(f3+f2+fry, 0)
    eq2 = Eq(f2*y2-f3*y3+t23x, 0)
    eq3 = Eq((f2+f3)*x1+t23y, 0)

    # Rigid Body II
    eq4 = Eq(-fry+ffy-fl, 0)
    eq5 = Eq(t14x+mlx-t23x, 0)
    eq6 = Eq(mly+t14y-t23y-fry*(x23-x1)-ffy*(x14-x3), 0)

    # Rigid Body III
    eq7 = Eq(f1+f4-ffy, 0)
    eq8 = Eq(f1*y1-f4*y4-t14x, 0)
    eq9 = Eq(-t14y-(f1+f4)*x3, 0)

    # Compression Springs
    eq10 = Eq(d1*s1-f1, 0)
    eq11 = Eq(d4*s4-f4, 0)
    eq12 = Eq(d2*s2-f2, 0)
    eq13 = Eq(d3*s3-f3, 0)

    # Torsion Springs
```

```
    eq14 = Eq((phi1-phi2)*d23-t23x, 0)
    eq15 = Eq((phi2-phi3)*d14-t14x, 0)

    # Kinematic constraints
    eq16 = Eq((s2-s3)/(y2+y3)-phi1, 0)
    eq17 = Eq((s1-s4)/(y1+y4)-phi3, 0)


    eqs = [eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8, eq9, eq10, eq11, eq12, eq13,
eq14, eq15, eq16, eq17]
    unknowns = [f1, f2, f3, f4, t14x, t23x, t14y, t23y, s1, s2, s3, s4, phi1,
phi2, phi3, fry, ffy]

    print(f'No of equations: {len(eqs)}\nNo of unknowns: {len(unknowns)}')

    ans = solve(eqs, unknowns)

    return ans

if __name__ == "__main__":
    ans = solve_model()

    y1, y2, y3, y4, d1, d2, d3, d4, dc, d23, d14, dt, h = symbols('y1 y2 y3 y4\
    d1 d2 d3 d4 dc d23 d14 dt h')

    ans_simpl = {}
    for key in ans:
        expr = ans[key]
        expr = expr.subs([(y1, h),
                          (y2, h),
                          (y3, h),
                          (y4, h),
                          (d1, dc),
                          (d2, dc),
                          (d3, dc),
                          (d4, dc),
                          (d23, dt),
                          (d14, dt)])

        ans_simpl[key] = simplify(expr)
        print(f'{key}: {ans_simpl[key]}\n---')
```

Results (simplified)

```
f1: (2*fl*h*x23 + 2*h*mly - mlx*x14 - mlx*x23)/(4*h*(x14 + x23))
---
f2: (2*fl*h*x14 - 2*h*mly - mlx*x14 - mlx*x23)/(4*h*(x14 + x23))
```

```
---
f3: (2*fl*h*x14 - 2*h*mly + mlx*x14 + mlx*x23)/(4*h*(x14 + x23))
---
f4: (2*fl*h*x23 + 2*h*mly + mlx*x14 + mlx*x23)/(4*h*(x14 + x23))
---
t14x: -mlx/2
---
t23x: mlx/2
---
t14y: x3*(-fl*x23 - mly)/(x14 + x23)
---
t23y: x1*(-fl*x14 + mly)/(x14 + x23)
---
s1: (2*fl*h*x23 + 2*h*mly - mlx*x14 - mlx*x23)/(4*dc*h*(x14 + x23))
---
s2: (2*fl*h*x14 - 2*h*mly - mlx*x14 - mlx*x23)/(4*dc*h*(x14 + x23))
---
s3: (2*fl*h*x14 - 2*h*mly + mlx*x14 + mlx*x23)/(4*dc*h*(x14 + x23))
---
s4: (2*fl*h*x23 + 2*h*mly + mlx*x14 + mlx*x23)/(4*dc*h*(x14 + x23))
---
phi1: -mlx/(4*dc*h**2)
---
phi2: -mlx/(2*dt) - mlx/(4*dc*h**2)
---
phi3: -mlx/(4*dc*h**2)
---
fry: (-fl*x14 + mly)/(x14 + x23)
---
ffy: (fl*x23 + mly)/(x14 + x23)
```