دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی مکانیک

ارتعاشات مکانیکی

ارتعاشات توربین بادی

نام نویسندگان:

محمدجواد محمدی

استاد راهنما:

دکتر بهرامی

بهمن ماه 1402

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی مکانیک

ارتعاشات مکانیکی

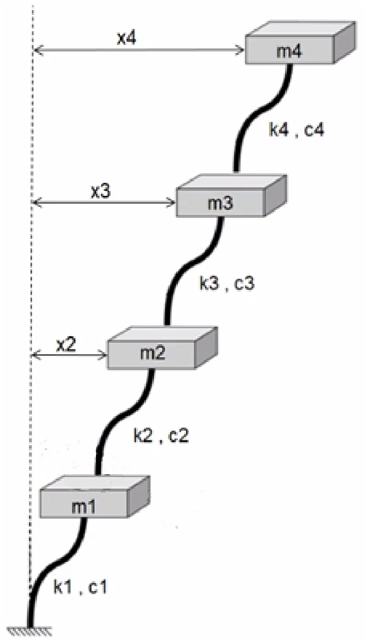
تشکر و قدردانی:

با نهایت احترام و سپاس، از زحمات و همکاری دستیار محترم درس، جناب آقای مهندس پیمان سبحانی، که با راهنمایی‌ها و پیشنهادات ارزشمندشان، مرا در انجام این پروژه یاری دادند. همچنین از جناب آقای مهندس ایمان سلجوقی نژاد، که با تهیه و رسم شکل‌های این پروژه، مرا در ارائه بهتر نتایج یاری رساندند.

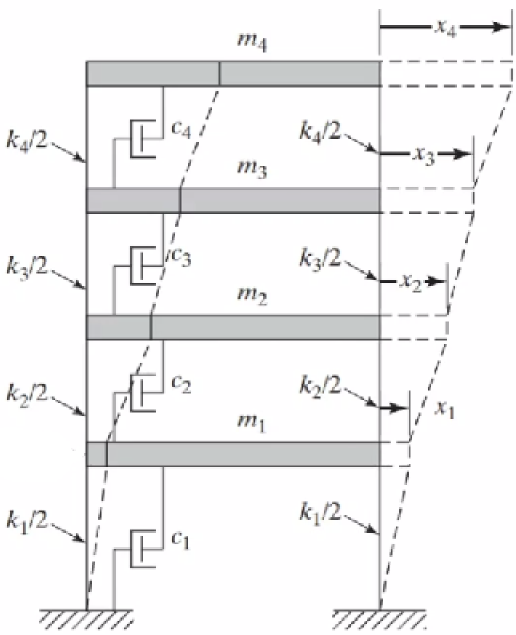
فهرست

مدل سازی مکانیکی

برای مدل سازی مکانیکی پایه توربین، از چهار جرم متمرکز تحت نیروهای خارجی استفاده شده است. بین این جرم ها میتوان فنر و دمپر فرض کرد (شکل ۱). ارتعاشات هر جرم متمرکز در راستای افقی است و فنر و دمپر نیز در همان راستا عمل میکند (شکل ۲).

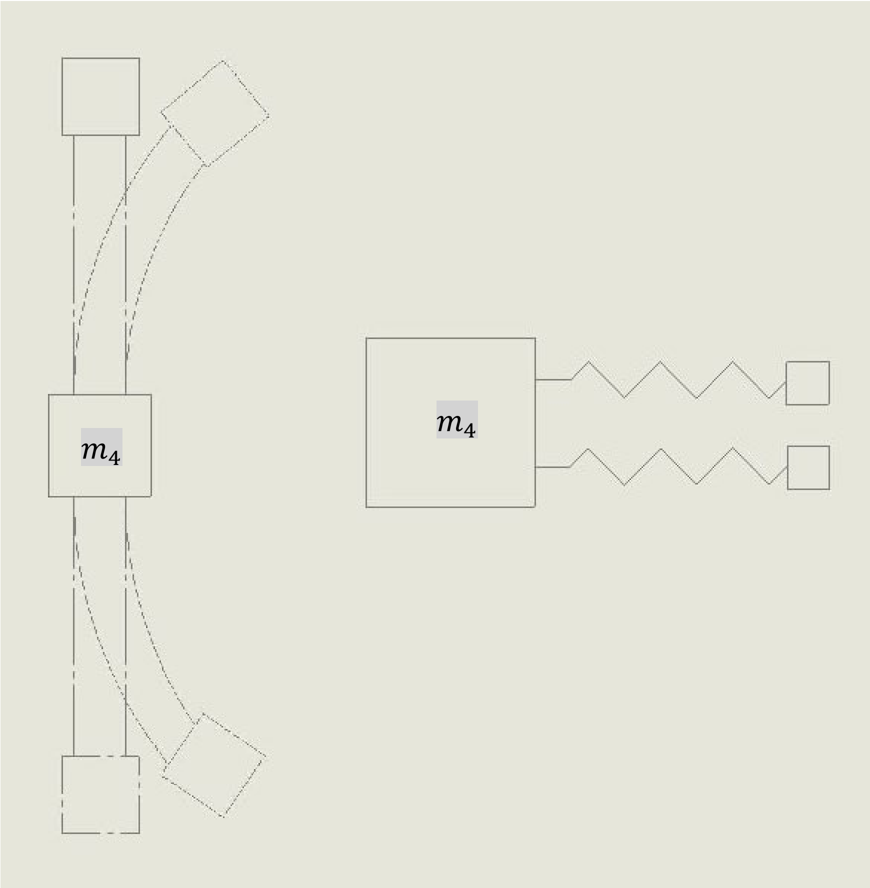


شکل ۲: راستای حرکت جرم های متمرکز



شکل ۱: شبیه سازی پایه توربین با چهار جرم متمرکز

(



شکل ۳: مدل سازی پره با تیر انعطاف پذیر و مدل سازی تیر با فنر

به عنوان یک فرض منطقی برای k_t همانطور که در طرح پروژه ذکر شده است، مقدار جابه‌جایی نباید بیشتر از هشت درجه باشد. با این فرض خواهیم داشت و حالت استاتیکی خواهیم داشت:

$$m_5 \times l_5 = k_t \times 8^\circ$$

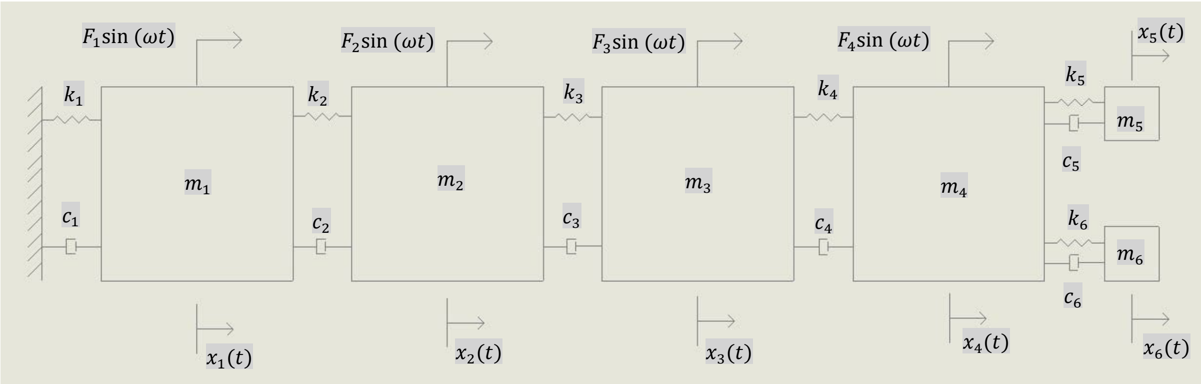
$$k_t = \frac{m_5 l_5}{8^\circ} = 251114.51 \frac{N.m}{rad}$$

برای k_6 و k_5 مقادیر داده شده در ضمیمه های پروژه برای اثر دادن فنر پیچشی و دمپر موجود در محل اتصال پره ها به جرم ۴، به شکل زیر تغییر میکنند (قانون فنرهای سری):

$$\frac{1}{k_{5,6}} = \frac{1}{k'_{5,6}} + \frac{3l_{5,6}}{k_t} \rightarrow k_5 = k_6 = 72.38\ N.m$$

همچنین برای c_t از فرض $c_{5,6} = 0.005\ k_{5,6}$ استفاده شده است.

در مجموع مدل سازی مکانیکی دارای ۷ درجه آزادی میباشد که با استفاده از قانون فنر های سری، فنر پیچشی و انعطاف تیر های متصل به جرم پره، به یک فنر تبدیل شده‌اند؛ در نتیجه توربین با ۶ درجه آزادی مدل سازی شده است. مدل سازی نهایی در شکل ۴ قابل مشاهده است.



شکل ۴: مدل سازی نهایی توربین

مدل سازی ریاضی

برای مدل سازی ریاضی و به دست آوردن معادلات حاکم بر سیستم، از روش لاگرانژ استفاده شده است.

با به دست آوردن انرژی جنبشی سیستم خواهیم داشت:

$$T = \frac{1}{2}[m_1\dot{x_1}^2 + m_2\dot{x_2}^2 + m_3\dot{x_3}^2 + m_4\dot{x_4}^2 + m_5\dot{x_5}^2 + m_6\dot{x_6}^2]$$

همچنین انرژی پتانسیل سیستم خواهد بود:

$$V = \frac{1}{2}[k_1x_1^2 + k_2(x_2 - x_1)^2 + k_3(x_3 - x_2)^2 + k_4(x_4 - x_3)^2 + k_5(-x_5 - x_4)^2 + k_6(x_6 - x_4)^2]$$

سپس R عامل دمپر ها در روش لاگرانژ به دست آمده است:

$$R=\frac{1}{2}\left[c_1\dot{x_1}^2+c_2(\dot{x_2}-\dot{x_1})^2+c_3(\dot{x_3}-\dot{x_2})^2+c_4(\dot{x_4}-\dot{x_3})^2+c_5(\dot{x_5}-\dot{x_4})^2 + c_6(\dot{x_6}-\dot{x_4})^2\right]$$

با توجه به روش لاگرانژ، معادلات حاکم بر سیستم به دست می‌آید:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x_i}}\right)-\frac{\partial T}{\partial x_i}+\frac{\partial R}{\partial \dot{x_i}}+\frac{\partial V}{\partial x_i}=F_i$$

که در اینجا i میتواند ۱ تا ۶ باشد.

معادلات سیستم (به ضمیمه ۱ مراجعه شود):

❶ $$m_1\ddot{x_1}+(c_1+c_2)\dot{x_1}-c_2\dot{x_2}+(k_1+k_2)x_1-k_2x_2=F_1\sin(\omega t)$$

❷ $$m_2\ddot{x_2}-c_2\dot{x_1}+(c_2+c_3)\dot{x_2}-c_3\dot{x_3}-k_2x_1+(k_2+k_3)x_2-k_3x_3=F_2\sin(\omega t)$$

❸ $$m_3\ddot{x_3}-c_3\dot{x_2}+(c_3+c_4)\dot{x_3}-c_4\dot{x_4}-k_3x_2+(k_3+k_4)x_3-k_4x_4=F_3\sin(\omega t)$$

❹ $$m_4\ddot{x_4}-c_4\dot{x_3}+(c_4+c_5+c_6)\dot{x_4}-c_5\dot{x_5}-c_6\dot{x_6}-k_4x_3+(k_4+k_5+k_6)x_4-k_5x_5 - k_6x_6=F_4\sin(\omega t)$$

❺ $$m_5\ddot{x_5}-c_5\dot{x_4}+c_5\dot{x_5}-k_5x_4+k_5x_5=0$$

❻ $$m_6\ddot{x_6}-c_6\dot{x_4}+c_6\dot{x_6}-k_6x_4+k_6x_6=0$$

از روی این معادلات میتوان ماتریس جرم، ماتریس سختی و ماتریس دمپر ها را به دست آورد:

$$[m]\ddot{\vec{x}}+[c]\dot{\vec{x}}+[k]\vec{x}=\vec{F}$$

ماتریس جرم:

$$\begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_6 \end{bmatrix}$$

ماتریس سختی:

$$\begin{bmatrix} k_1+k_2 & -k_2 & 0 & 0 & 0 & 0 \\ -k_2 & k_2+k_3 & -k_3 & 0 & 0 & 0 \\ 0 & -k_3 & k_3+k_4 & -k_4 & 0 & 0 \\ 0 & 0 & -k_4 & k_4+k_5+k_6 & -k_5 & -k_6 \\ 0 & 0 & 0 & -k_5 & k_5 & 0 \\ 0 & 0 & 0 & -k_6 & 0 & k_6 \end{bmatrix}$$

ماتریس دمپرها:

$$\begin{bmatrix} c_1+c_2 & -c_2 & 0 & 0 & 0 & 0 \\ -c_2 & c_2+c_3 & -c_3 & 0 & 0 & 0 \\ 0 & -c_3 & c_3+c_4 & -c_4 & 0 & 0 \\ 0 & 0 & -c_4 & c_4+c_5+c_6 & -c_5 & -c_6 \\ 0 & 0 & 0 & -c_5 & c_5 & 0 \\ 0 & 0 & 0 & -c_6 & 0 & c_6 \end{bmatrix}$$

با توجه به اینکه طبق فرضیات پروژه و برای حل شدن آن $C = 0.005\ K$ است، مقادیر c_1 تا c_6 مشخص میشود.

فرکانس طبیعی و شکل مد ها

فرکانس طبیعی

برای به دست آوردن فرکانس طبیعی و شکل مد ها، فرض میشود دمپر وجود ندارد و همچنین همه نیروهای خارجی صفر هستند.

$$[m]\ddot{\vec{x}} + [k]\vec{x} = 0$$

با فرض اینکه جواب x به صورت زیر است خواهیم داشت:

$$[x] = [u]\ e^{i\omega t}$$

با جایگذاری آن در معادله اصلی:

$$-\omega^2[m][u] + [k][u] = 0$$

$$\rightarrow ([k] - \omega^2[m])[u] = 0$$

برای به دست آوردن جواب غیر بدیهی باید دترمینان ماتریس پشت $[u]$ صفر باشد. و از آن مقدار ω ها به دست می‌آید (به ضمیمه ۲ مراجعه شود):

$$\omega_1 = 0.1921$$

$$\omega_2 = 0.3058$$

$$\omega_3 = 0.3701$$

$$\omega_4 = 0.721$$

$$\omega_5 = 1.1115$$

$$\omega_6 = 1.4038$$

شکل مد ها

برای به دست آوردن شکل مد ها از رابطه ای که در بالا به دست آوردیم استفاده میکنیم و مقادیر $[u]$ به ما شکل مد ها را خواهند داد (برای نرمالایز کردن نیز در همه شکل مد ها، u1 را برابر ۱ قرار میدهیم):

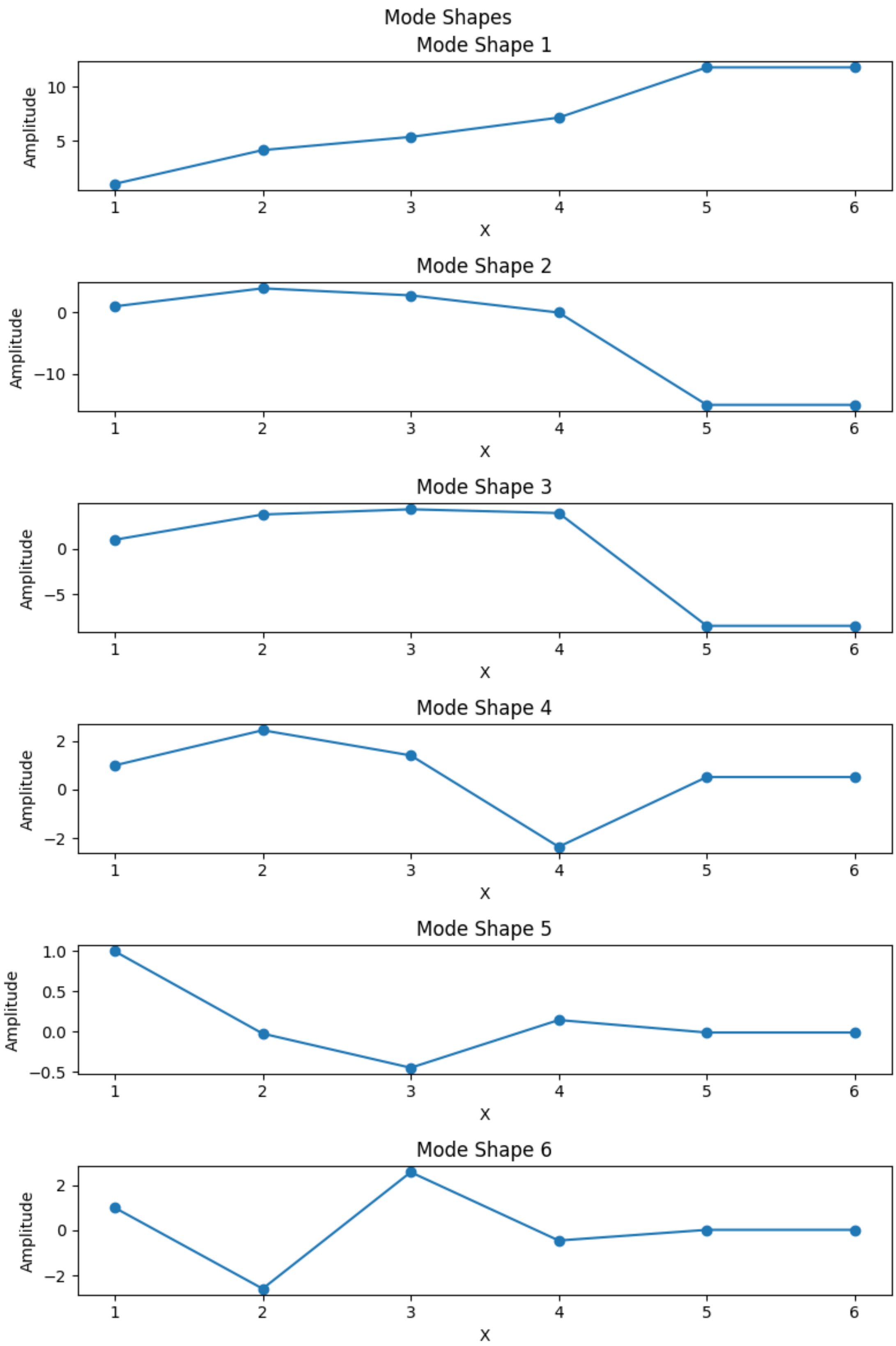
$$([k] - \omega^2[m])[u] = 0$$

خواهیم داشت (به ضمیمه ۳ مراجعه شود):

$$\begin{bmatrix} 1 \\ 4.1262 \\ 5.3462 \\ 7.1341 \\ 11.7817 \\ 11.7817 \end{bmatrix} \begin{bmatrix} 1 \\ 3.9301 \\ 2.7871 \\ 0 \\ -14.987 \\ -14.987 \end{bmatrix} \begin{bmatrix} 1 \\ 3.7795 \\ 4.3541 \\ 3.9392 \\ -8.4751 \\ -8.4751 \end{bmatrix} \begin{bmatrix} 1 \\ 2.4533 \\ 1.4134 \\ -2.3644 \\ 0.5186 \\ 0.5186 \end{bmatrix} \begin{bmatrix} 1 \\ -0.0252 \\ -0.4503 \\ 0.1437 \\ -0.0118 \\ -0.0118 \end{bmatrix} \begin{bmatrix} 1 \\ -2.5722 \\ 2.5641 \\ -0.4482 \\ 0.0223 \\ 0.0223 \end{bmatrix}$$

نکته قابل توجه این است که در تمام شکل مد ها، جابه‌جایی پروانه ها به یک اندازه و در یک جهت است (u5 u6, در تمام شکل مد ها).

شکل مد ها روی نمودار رسم شده اند:



شکل ۵: نمایش شکل مد ها روی نمودار

تحریک مد سوم

خواسته چهارم پروژه این است که با کدام مجموعه از جابه‌جایی اولیه میتوان فقط شکل مد سوم را تحریک کرد. تنها راهی که میتوان ارتعاشاتی بر مبنای شکل مد سوم داشت این است که جابه جایی های اولیه ای به نسبت شکل مد به سیستم داده شود و تمام نیروهای خارجی صفر شوند. در این صورت ارتعاش سیستم مبتنی بر شکل مد سوم خواهد بود. به عنوان یک مجموعه از جابه‌جایی اولیه که میتواند فقط و فقط شکل مد سوم را تحریک کند خواهیم داشت:

$$x_1(0) = 0.01m$$

$$x_2(0) = 0.0378m$$

$$x_3(0) = 0.0435m$$

$$x_4(0) = 0.0394m$$

$$x_5(0) = -0.0848m$$

$$x_6(0) = -0.0848m$$

حل عددی سیستم

در قسمت سوم پروژه خواسته شده است که سیستم با استفاده از Matlab ode45 حل شود. با مفروضات زیر سیستم حل میشود:

$$y_1 = x_1, \quad y_2 = \dot{x}_1, \quad y_3 = x_2, \quad y_4 = \dot{x}_2, \quad y_5 = x_3, \quad y_6 = \dot{x}_3, \quad y_7 = x_4,$$
$$y_8 = \dot{x}_4, \quad y_9 = x_5, \quad y_{10} = \dot{x}_5, \quad y_{11} = x_6, \quad y_{12} = \dot{x}_6$$

و مشتقات yi ها:

$$\dot{y}_1 = y_2, \quad \dot{y}_3 = y_4, \quad \dot{y}_5 = y_6, \quad \dot{y}_7 = y_8, \quad \dot{y}_9 = y_{10}, \quad \dot{y}_{11} = y_{12}$$

و بقیه مشتقات از معادلات ۱ تا ۶ که در بخش ۱.۲ (مدل سازی ریاضی) به دست آمد، به دست می‌آید.

$$\dot{y}_2 = \ddot{x}_1 = \frac{-(c_1+c_2)\dot{x}_1 + c_2\dot{x}_2 - (k_1+k_2)x_1 + k_2x_2 + F_1\sin(\omega t)}{m_1}$$

$$\dot{y}_4 = \ddot{x}_2 = \frac{c_2\dot{x}_1 - (c_2+c_3)\dot{x}_2 + c_3\dot{x}_3 + k_2x_1 - (k_2+k_3)x_2 + k_3x_3 + F_2\sin(\omega t)}{m_2}$$

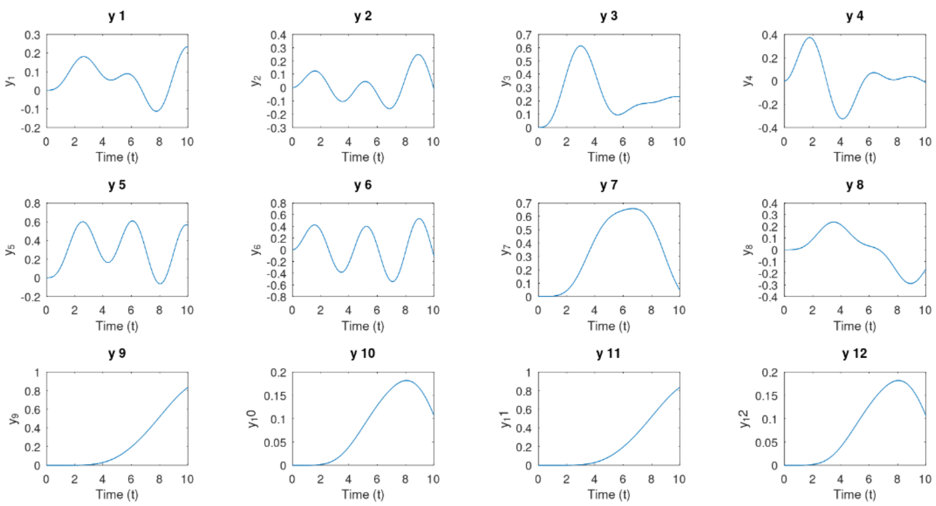
$$\dot{y}_6 = \ddot{x}_3 = \frac{c_3\dot{x}_2 - (c_3+c_4)\dot{x}_3 + c_4\dot{x}_4 + k_3x_2 - (k_3+k_4)x_3 + k_4x_4 + F_3\sin(\omega t)}{m_3}$$

$$\dot{y}_8 = \ddot{x}_4 = \frac{c_4\dot{x}_3 - (c_4+c_5+c_6)\dot{x}_4 + c_5\dot{x}_5 + c_6\dot{x}_6 + k_4x_3 - (k_4+k_5+k_6)x_4 + k_5x_5 + k_6x_6 + F_4\sin(\omega t)}{m_4}$$

$$\dot{y}_{10} = \ddot{x}_5 = \frac{c_5\dot{x}_4 - c_5\dot{x}_5 + k_5x_4 - k_5x_5}{m_5}$$

$$\dot{y}_{12} = \ddot{x}_6 = \frac{c_6\dot{x}_4 - c_6\dot{x}_6 + k_6x_4 - k_6x_6}{m_6}$$

و با استفاده از ode45 نتایج زیر حاصل میشود (به ضمیمه ۵ مراجعه شود):



شکل ۶: جواب حل عددی با استفاده از متلب

هر یک از y ها سرعت و جابه‌جایی هر یک از جرم ها هستد. y های فرد جابه‌جایی و y های زوج سرعت هستند. هر دو جفت برای یک جرم است.

ضمیمه ها

ضمیمه ۱: تعریف V و R و T و به دست آوردن معادلات از روش لاگرانژ

```python
T = (
        1
        / 2
        * (
            m1 * x1_**2
            + m2 * x2_**2
            + m3 * x3_**2
            + m4 * x4_**2
            + m5 * x5_**2
            + m6 * x6_**2
        )
)
V = (
        1
        / 2
        * (
            k1 * x1**2
            + k2 * (x2 - x1) ** 2
            + k3 * (x3 - x2) ** 2
            + k4 * (x4 - x3) ** 2
            + k5 * (x5 - x4) ** 2
            + k6 * (x6 - x4) ** 2
        )
)
R = (
        1
        / 2
        * (
            c1 * x1_**2
            + c2 * (x2_ - x1_) ** 2
            + c3 * (x3_ - x2_) ** 2
            + c4 * (x4_ - x3_) ** 2
            + c5 * (x5_ - x4_) ** 2
            + c6 * (x6_ - x4_) ** 2
        )
)
```

Python

```python
for xi, xi_, fi in zip(
    [x1, x2, x3, x4, x5, x6], [x1_, x2_, x3_, x4_, x5_, x6_], [f1, f2, f3, f4, f5, f6]
):
    print(
        latex(
            Eq(
                simplify(
                    diff(diff(T, xi_), t) - diff(T, xi) + diff(R, xi_) + diff(V, xi)
                ),
                fi,
            )
        )
        + "\\\\"
    )
```

Python

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}_i}\right)-\frac{\partial T}{\partial x_i}+\frac{\partial R}{\partial \dot{x}_i}+\frac{\partial V}{\partial x_i}=F_i$$

$$\begin{gathered}
m_1\ddot{x}_1(t)+(c_1+c_2)\,\dot{x}_1(t)-c_2\dot{x}_2(t)+(k_1+k_2)\,x_1(t)-k_2x_2(t)=F_1\sin(\omega t)\\
m_2\ddot{x}_2(t)-c_2\dot{x}_1(t)+(c_2+c_3)\,\dot{x}_2(t)-c_3\dot{x}_3(t)-k_2x_1(t)+(k_2+k_3)\,x_2(t)-k_3x_3(t)=F_2\sin(\omega t)\\
m_3\ddot{x}_3(t)-c_3\dot{x}_2(t)+(c_3+c_4)\,\dot{x}_3(t)-c_4\dot{x}_4(t)-k_3x_2(t)+(k_3+k_4)\,x_3(t)-k_4x_4(t)=F_3\sin(\omega t)\\
m_4\ddot{x}_4(t)-c_4\dot{x}_3(t)+(c_4+c_5+c_6)\,\dot{x}_4(t)-c_5\dot{x}_5(t)-c_6\dot{x}_6(t)-k_4x_3(t)+(k_4+k_5+k_6)\,x_4(t)-k_5x_5(t)-k_6x_6(t)=F_4\sin(\omega t)\\
m_5\ddot{x}_5(t)-c_5\dot{x}_4(t)+c_5\dot{x}_5(t)-k_5x_4(t)+k_5x_5(t)=0\\
m_6\ddot{x}_6(t)-c_6\dot{x}_4(t)+c_6\dot{x}_6(t)-k_6x_4(t)+k_6x_6(t)=0
\end{gathered}$$

ضمیمه ۲: محاسبه فرکانس طبیعی

```python
M = Matrix(
    [
        [m1, 0, 0, 0, 0, 0],
        [0, m2, 0, 0, 0, 0],
        [0, 0, m3, 0, 0, 0],
        [0, 0, 0, m4, 0, 0],
        [0, 0, 0, 0, m5, 0],
        [0, 0, 0, 0, 0, m6],
    ]
)
K = Matrix(
    [
        [k1 + k2, -k2, 0, 0, 0, 0],
        [-k2, k2 + k3, -k3, 0, 0, 0],
        [0, -k3, k3 + k4, -k4, 0, 0],
        [0, 0, -k4, k4 + k5 + k6, -k5, -k6],
        [0, 0, 0, -k5, k5, 0],
        [0, 0, 0, -k6, 0, k6],
    ]
)
C = Matrix(
    [
        [c1 + c2, -c2, 0, 0, 0, 0],
        [-c2, c2 + c3, -c3, 0, 0, 0],
        [0, -c3, c3 + c4, -c4, 0, 0],
        [0, 0, -c4, c4 + c5 + c6, -c5, -c6],
        [0, 0, 0, -c5, c5, 0],
        [0, 0, 0, -c6, 0, c6],
    ]
)
```
Python

به علت اینکه پایتون در محاسبه دترمینان ماتریس ضعیف عملکرد، قسمت محاسبه دترمینان با استفاده از متلب انجام شد و نتایج آن مجدد در پایتون استفاده شد:

```python
omega = symbols("omega", positive=True)
detA = (
    1415971023198159830 4 * omega**12
    - (621719671149961893345367825396371 * omega**10) / 10995116277760
    + (35050848776533742656295507205498664899695502531 * omega**8)
    / 483570327845851669882470400
    - (75311578242617871070629674718747481141348436261918044721138687 * omega**6)
    / 21778071482940061661655974875633165533184 00
    + (
        289964276793629251950850106624551977220525971357348930858240581725589383 83
        * omega**4
    )
    / 47890485652059026823698344598447161988085597568237568 00
    - (
        43115870826747025704178720590225194911843770453001531206491698746156282002069269 6937
        * omega**2
    )
    / 1053122916685571866979180276836704323188950954005491112543109775360
    + 1427029678053761462824127969060771598519075555227169025486289096253622491352945 5991
    / 1684996666696914987166688442938726917102321526408785780068975640576
)
slv = solve(detA)
slv
```

[0.192066845645279, 0.305803288657833, 0.370110092695305, 0.721020831766172, 1.11148705430564, 1.40382215327542]

ضمیمه ۳: محاسبه شکل مد ها و رسم آن

```python
u1, u2, u3, u4, u5, u6 = symbols("u_1 u_2 u_3 u_4 u_5 u_6")
U = Matrix([[1], [u2], [u3], [u4], [u5], [u6]])
for i, omega in enumerate(slv):
    eqns = list((K - omega**2 * M) * U)
    eqns.pop(1)
    slv_u = solve(eqns)
    print(
        f"mode shape {i + 1}:",
        [
            1,
            slv_u[u2],
            slv_u[u3],
            slv_u[u4],
            slv_u[u5],
            slv_u[u6],
        ],
    )
```

```
mode shape 1: [1, 4.12624527528305, 5.34616496240868, 7.13409361678757, 11.7816845394626, 11.7816845394626]
mode shape 2: [1, 3.93010319851127, 2.78705578674506, -7.26769201580176e-15, -14.9869519414685, -14.9869519414685]
mode shape 3: [1, 3.77954548071414, 4.35405678444675, 3.93919915879846, -8.47508906567973, -8.47508906567973]
mode shape 4: [1, 2.45328577193047, 1.41338097834737, -2.36436173865839, 0.518592784324139, 0.518592784324139]
mode shape 5: [1, -0.0251882745286197, -0.450313459499589, 0.143659977277757, -0.0117651279810719, -0.0117651279810719]
mode shape 6: [1, -2.57217918181050, 2.56406446716656, -0.448168994170771, 0.0223262270284678, 0.0223262270284678]
```

$u1$ در همه مد ها برای نرمالایز کردن، ۱ فرض شد.

رسم مد ها:

```python
mode_shapes = [···

fig, axes = plt.subplots(nrows=6, ncols=1, figsize=(8, 12))

for i, mode_shape in enumerate(mode_shapes):
    x, y = zip(*mode_shape)
    axes[i].plot(x, y, marker="o", label=f"Mode {i+1}")
    axes[i].set_title(f"Mode Shape {i+1}")
    axes[i].set_xlabel("X")
    axes[i].set_ylabel("Amplitude")

fig.suptitle("Mode Shapes")

plt.tight_layout()

plt.show()
```

Python

ضمیمه ۵: حل عددی ode45

```matlab
function y_ = odefunc(t, y)
    f = [306.25, 612.5, 918.75, 1225, 0, 0];
    m = [2.294e3, 1.941e3, 1.943e3, 2.732e3, 0.774e3, 0.774e3];
    k = [2155.059, 662.275, 1455, 778.436, 1 / (1/75.332 + 3 * deg2rad(8) / m(5)), 1 / (1/75.332 + 3 * deg2rad(8) / m(6))];
    c = 0.005 .* k;
    omega = 1.75;
    y_ = zeros(12, 1);
    y_(1) = y(2);
    y_(2) = (f(1) * sin(omega * t) - (c(1) + c(2)) * y(2) + c(2) * y(4) - (k(1) + k(2)) * y(1) + k(2) * y(3)) / m(1);
    y_(3) = y(4);
    y_(4) = (f(2) * sin(omega * t) + c(2) * y(2) - (c(2) + c(3)) * y(4) + c(3) * y(6) + k(2) * y(1) - (k(2) + k(3)) * y(3) + k(3) *
y(5)) / m(2);
    y_(5) = y(6);
    y_(6) = (f(3) * sin(omega * t) +c(3) * y(4) - (c(3) + c(4)) * y(6) + c(4) * y(8) + k(3) * y(3) - (k(3) + k(4)) * y(5) + k(4) * y(7))
/ m(3);
    y_(7) = y(8);
    y_(8) = (c(4) * y(6) - (c(4) + c(5) + c(6)) * y(8) + c(5) * y(10) + c(6) * y(12) + k(4) * y(5) - (k(4) + k(5) + k(6)) * y(7) + k(5)
* y(9) + k(6) * y(11)) / m(4);
    y_(9) = y(10);
    y_(10) = (c(5) * y(8) - c(5) * y(10) + k(5) * y(7) - k(5) * y(9)) / m(5);
    y_(11) = y(12);
    y_(12) = (c(6) * y(8) - c(6) * y(12) + k(6) * y(7) - k(6) * y(11)) / m(6);
end
```

تعریف odefunc

```
tspan = [0:0.01:10];
y0 = zeros(12, 1);
[t, y] = ode45('odefunc', tspan, y0);

% plot
for col = 1:12
    subplot(3, 4, col);
    plot(t, y(:, col));
    title(['y ', num2str(col)]);
    xlabel('Time (t)');
    ylabel(['y_', num2str(col)]);
end
```

استفاده از ode45 و کشیدن نمودار

منابع

- Mechanical Vibrations Rao: Section 1.7.4, Combinations of Springs
- Mechanical Vibrations Rao: Chapter 6: Multidegree-of-Freedom Systems