Київський національний університет імені Т. Шевченка
Факультет комп'ютерних наук та кібернетики

Моделювання систем

Лабораторна робота 2

Виконав
студент групи ІС-31
А.С. Хома

Київ-2018

Умова

Для математичної моделі коливання трьох мас m_1, m_2, m_3, які поєднані між собою пружинами з відповідними жорсткостями c_1, c_2, c_3, c_4, і відомої функції спостереження координат моделі $\overline{y}(t)$, $t \in [t_0, t_k]$ потрібно оцінити частину невідомих параметрів моделі з використанням функції чутливості.

Математична модель коливання трьох мас описується наступною системою

$$\frac{dy}{dt} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{c_1+c_2}{m_1} & 0 & \frac{c_2}{m_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{c_2}{m_2} & 0 & -\frac{c_2+c_3}{m_2} & 0 & \frac{c_3}{m_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{c_3}{m_3} & 0 & -\frac{c_3+c_4}{m_3} & 0 \end{pmatrix} y = Ay \tag{1}$$

Показник якості ідентифікації параметрів невідомих параметрів β має вигляд

$$I(\beta) = \int_{t_0}^{t_k} (\overline{y}(t) - y(t))^T (\overline{y}(t) - y(t))\, dt \tag{2}$$

Якщо представити вектор невідомих параметрів $\beta = \beta_0 + \Delta\beta$, де β_0 – початкове наближення вектора параметрів,

$$\Delta\beta = \left(\int_{t_0}^{t_k} U^T(t) U(t)\, dt \right)^{-1} \int_{t_0}^{t_k} U^T(t) (\overline{y}(t) - y(t))\, dt \tag{3}$$

Матриці чутливості $U(t)$ визначається з наступної матричної системи диференціальних рівнянь

$$\begin{aligned} \frac{dU(t)}{dt} &= \frac{\partial(Ay)}{\partial y^T} U(t) + \frac{\partial(Ay)}{\partial \beta^T} \\ U(t_0) &= 0, \quad \beta = \beta_0 \end{aligned} \tag{4}$$

В нашому випадку $\frac{\partial(Ay)}{\partial y^T} = A$.

Хід роботи

1. Використовуючи метод Рунге-Кутта 4-го порядку, розв'язуємо рівняння 4.

   1.1. Визначимо матрицю A.

```
A[0, 1] = 1.
A[1, 0] = -(rigidity_param[0] + rigidity_param[1]) / weight[0]
A[1, 2] = rigidity_param[1] / weight[0]
A[2, 3] = 1.
A[3, 0] = rigidity_param[1] / weight[1]
A[3, 2] = -(rigidity_param[1] + rigidity_param[2]) / weight[1]
A[3, 4] = rigidity_param[2] / weight[1]
A[4, 5] = 1.
A[5, 2] = rigidity_param[2] / weight[2]
A[5, 4] = -(rigidity_param[3] + rigidity_param[2]) / weight[2]
```

   1.2. Порахуємо $\frac{\partial(Ay)}{\partial\beta^T}$, $\quad \beta^T = [c_3, m_1, m_3]$.

```
dA1 = np.zeros_like(A)
dA2 = np.zeros_like(A)
dA3 = np.zeros_like(A)

dA1[3, 2] = -1. / weight[1]
dA1[3, 4] = 1. / weight[1]
dA1[5, 2] = 1. / weight[2]
dA1[5, 4] = -1. / weight[2]

dA2[1, 0] = (rigidity_param[1] + rigidity_param[0]) / (weight[0] * weight[0])
dA2[1, 2] = -(rigidity_param[1]) / (weight[0] * weight[0])

dA3[5, 2] = -(rigidity_param[2]) / (weight[2] * weight[2])
dA3[5, 4] = (rigidity_param[3] + rigidity_param[2]) / (weight[2] * weight[2])
```

   1.3. Обчислимо матрицю U та y.

```
U = np.zeros((6, 3))
y_vec = np.copy(y_func[:, 0].reshape(-1, 1))

for i in range(1, n):
    # Update U
    delta_U = np.column_stack(((dA1 @ y_vec).reshape(-1),
                               (dA2 @ y_vec).reshape(-1), (dA3 @ y_vec).reshape(-1)))

    K1 = dt * ((A @ U) + delta_U)
    K2 = dt * ((A @ (U + .5 * K1)) + delta_U)
    K3 = dt * ((A @ (U + .5 * K2)) + delta_U)
    K4 = dt * ((A @ (U + K3)) + delta_U)
    U += (1. / 6. * (K1 + 2. * K2 + 2. * K3 + K4))

    # Calculate new y
    k1 = dt * (A @ y_vec)
    k2 = dt * (A @ (y_vec + .5 * k1))
    k3 = dt * (A @ (y_vec + .5 * k2))
    k4 = dt * (A @ (y_vec + k3))
    y_vec += 1. / 6. * (k1 + 2. * k2 + 2. * k3 + k4)
```

2. Маючи пораховані y та U, обчислимо перший та другий інтеграл з 3-го рівняння, а також інтеграл з 2-го рівняння.

```
left_int_part += U.T @ U
right_int_part += U.T @ (y_func[:, i].reshape(-1, 1) - y_vec)

temp_iden_beta += (y_func[:, i].reshape(-1, 1) - y_vec).T @ (y_func[:, i] - y_vec.reshape(-1))
```

3. Перерахуємо $\Delta\beta,\ \beta$ та $I\,(\beta)$.

```
dBeta = np.linalg.inv(left_int_part * dt) @ (right_int_part * dt)
beta += dBeta.reshape(-1)

prev_ident_beta = identification_beta
identification_beta = temp_iden_beta * dt
```

4. Запустимо ітераційний процес.

```
Iteration 1. Quality score of beta identification 28.550785190215727
Iteration 2. Quality score of beta identification 2.774030840628150
Iteration 3. Quality score of beta identification 0.083450176880015
Iteration 4. Quality score of beta identification 0.000073042203514
Iteration 5. Quality score of beta identification 0.000000011476591
Iteration 6. Quality score of beta identification 0.000000010692351
Iteration 7. Quality score of beta identification 0.000000010692340
```

```
print('Beta', beta)
```

```
Beta [ 0.19999989 11.99999485 17.9999918 ]
```

Маємо $\beta = [0.2, 12, 18]$, тобто $c_3 = 0.2,\ m_1 = 12,\ m_3 = 18$.