

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет систем управления и робототехники

ПРАКТИЧЕСКАЯ РАБОТА №2
по дисциплине
«Имитационное моделирование робототехнических систем»

Студент:
Группа № R4135с

Е.А. Щерблюк

Предподаватель:

Е.А. Ракшин

Санкт-Петербург
2025

Цель работы: необходимо составить и решить дифференциальное уравнение (ОДУ), описывающего динамику системы «масса–пружина–демпфер» с заданными параметрами.

Ход работы

Таблица 1 – Задание варианта №73

Вариант	m, kg	k, N/m, Nm/rad	b, N*s/m, Nm*s/rad	I, m	θ_0 , rad	x_0 , m
1	0.5	14.2	0.025	0.76	0.3141690229	0.87

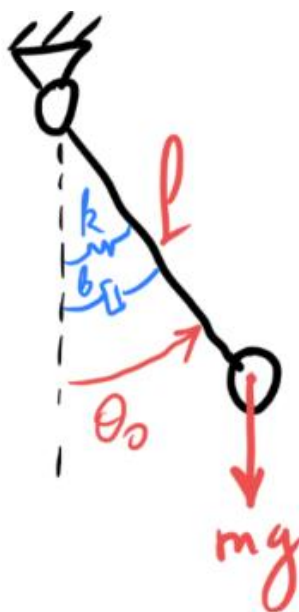


Рисунок 1 – Схема для 1 варианта

1. Составление ОДУ

Для вывода уравнения движения системы «масса–пружина–демпфер» необходимо воспользоваться методом Лагранжа. Система состоит из тела массой m , соединённого с неподвижной опорой пружиной жёсткостью k и демпфером с коэффициентом вязкого трения b . Движение происходит вдоль одной оси, поэтому система описывается одной обобщённой координатой $x(t)$ – смещением массы от нулевого положения пружины.

Кинетическая энергия системы:

$$K = \frac{1}{2} m \dot{x}^2$$

Потенциальная энергия (включая гравитационную и упругую):

$$P = mgx + \frac{1}{2} kx^2$$

Лагранжиан:

$$L = K - P = \frac{1}{2}m\dot{x}^2 - (mgx + \frac{1}{2}kx^2)$$

Внешняя неконсервативная сила (сила демпфирования):

$$Q = -b\dot{x}$$

Уравнение Лагранжа второго рода:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = Q$$

Вычислим производные:

$$\frac{\partial L}{\partial \dot{x}} = m\dot{x}, \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = m\ddot{x}$$

$$\frac{\partial L}{\partial x} = -mg - kx$$

Подставляем в уравнение Лагранжа:

$$m\ddot{x} - (-mg - kx) = -b\dot{x}$$

После преобразований получаем дифференциальное уравнение движения:

$$m\ddot{x} + b\dot{x} + kx + mg = 0$$

или, эквивалентно (по второму закону Ньютона $F = m\ddot{x}$):

$$m\ddot{x} = -b\dot{x} - kx - mg$$

Это линейное неоднородное дифференциальное уравнение второго порядка с постоянными коэффициентами.

2. Аналитическое решение ОДУ

Уравнение $m\ddot{x} + b\dot{x} + kx = -mg$ является линейным неоднородным дифференциальным уравнением с постоянными коэффициентами. Его общее решение складывается из общего решения однородного уравнения и частного решения неоднородного.

Частное решение. Правая часть уравнения — константа ($-mg$), поэтому ищем частное решение в виде константы $x_p = C$. Подстановка в уравнение даёт:

$$0 + 0 + kC = -mg \Rightarrow C = -\frac{mg}{k}$$

Следовательно,

$$x_p(t) = -\frac{mg}{k}$$

Однородное уравнение.

$$m\ddot{x} + b\dot{x} + kx = 0$$

Характеристическое уравнение:

$$mr^2 + br + k = 0$$

Дискриминант:

$$D = b^2 - 4mk$$

Подставим численные значения из задания: ($m = 0,5$; $k = 14,2$; $b = 0,025$, $D = (0,025)^2 - 4 \cdot 0,5 \cdot 14,2 = 0,000625 - 28,4 = -28,39 < 0$)

Поскольку дискриминант отрицателен, корни характеристического уравнения комплексные, и система находится в режиме неэкстремального затухания.

Обозначим:

- Собственная частота: $\omega_n = \sqrt{\frac{k}{m}}$
- Коэффициент затухания: $\zeta = \frac{b}{2\sqrt{mk}}$
- Демпфированная частота: $\omega_d = \omega_n \sqrt{1 - \zeta^2}$

Тогда общее решение однородного уравнения:

$$x_h(t) = e^{-\zeta\omega_n t} (A \cos(\omega_d t) + B \sin(\omega_d t))$$

Общее решение неоднородного уравнения:

$$x(t) = x_h(t) + x_p(t) = e^{-\zeta\omega_n t} (A \cos(\omega_d t) + B \sin(\omega_d t)) - \frac{mg}{k}$$

Определение констант A и B из начальных условий.

Из условия задачи:

- $x(0) = x_0 = 0,87 \text{ м}$
- Начальная скорость не задана; примем $\dot{x}(0) = v_0 = 0 \text{ м/с}$

Подставляя $t = 0$:

$$x(0) = A - \frac{mg}{k} = x_0 \Rightarrow A = x_0 + \frac{mg}{k}$$

Найдём производную:

$$\dot{x}(t) = -\zeta\omega_n e^{-\zeta\omega_n t} (A \cos(\omega_d t) + B \sin(\omega_d t)) + e^{-\zeta\omega_n t} (-A\omega_d \sin(\omega_d t) + B\omega_d \cos(\omega_d t))$$

При $t = 0$:

$$\dot{x}(0) = -\zeta\omega_n A + B\omega_d = v_0 = 0 \Rightarrow B = \frac{\zeta\omega_n A}{\omega_d}$$

Подставляя численные значения:

- $g = 9,81 \text{ м/с}^2$
- $\frac{mg}{k} = \frac{0,5 \cdot 9,81}{14,2} \approx 0,3454 \text{ м}$
- $A = 0,87 + 0,3454 = 1,2154$
- $\omega_n = \sqrt{14,2/0,5} \approx 5,3291 \text{ рад/с}$
- $\zeta = \frac{0,025}{2\sqrt{0,5 \cdot 14,2}} \approx 0,0047$

- $\omega_d = \omega_n \sqrt{1 - \zeta^2} \approx 5,3291 \text{ рад/с}$ (почти не отличается от ω_n из-за малого затухания)
- $B \approx \frac{0,0047 \cdot 5,3291 \cdot 1,2154}{5,3291} \approx 0,00571$

Таким образом, аналитическое решение существует и имеет вид:

$$x(t) = e^{-0,025t}(1,2154 \cos(5,3291t) + 0,00571 \sin(5,3291t)) - 0,3454$$

3. Численные методы

Для решения дифференциального уравнения движения системы «масса–пружина–демпфер» применяются численные методы интегрирования. Сравнение этих методов с аналитическим решением позволяет оценить их точность, устойчивость и применимость для моделирования колебательных систем.

В данной работе реализованы три классических численных метода различного порядка точности:

- явный метод Эйлера (Explicit Euler);
- неявный метод Эйлера (Implicit Euler);
- метод Рунге–Кутты четвёртого порядка (RK4).

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # -----
5 # ПАРАМЕТРЫ СИСТЕМЫ
6 # -----
7 m = 0.5      # масса, кг
8 k = 14.2     # жёсткость пружины, Н/м
9 b = 0.025    # коэффициент демпфирования, Н·с/м
10 g = 9.81     # ускорение свободного падения, м/с²
11
12 # Начальные условия
13 x0 = 0.87    # начальное смещение, м
14 v0 = 0.0     # начальная скорость, м/с (не задана – принимаем 0)
15
16 # Параметры интегрирования
17 T_final = 10.0 # конечное время, с
18 h = 0.01     # шаг интегрирования, с
19 t_vals = np.arange(0, T_final + h, h)
20
21 # -----
22 # ДИНАМИКА СИСТЕМЫ
23 # -----
24 def mass_spring_damper_dynamics(state):
25     x, v = state
26     a = -(b / m) * v - (k / m) * x - g
27     return np.array([v, a])
28
29 # -----
30 # АНАЛИТИЧЕСКОЕ РЕШЕНИЕ
31 # -----
32 x_eq = - (m * g) / k # равновесное положение

```

Рисунок 2 – Код часть 1

```

33
34 omega_n = np.sqrt(k / m)          # собственная частота
35 zeta = b / (2 * np.sqrt(m * k))    # коэффициент затухания
36 omega_d = omega_n * np.sqrt(1 - zeta**2) # демпфированная частота
37
38 # Константы из начальных условий
39 A = x0 - x_eq
40 B = (v0 + zeta * omega_n * A) / omega_d
41
42 def analytical_solution(t):
43     exp_decay = np.exp(-zeta * omega_n * t)
44     return exp_decay * (A * np.cos(omega_d * t) + B * np.sin(omega_d * t)) + x_eq
45
46 # Вычисляем аналитическое решение
47 x_anal = analytical_solution(t_vals)
48 v_anal = np.gradient(x_anal, h)    # приближение скорости
49
50 # -----
51 # ЧИСЛЕННЫЕ МЕТОДЫ
52 # -----
53 def explicit_euler(f, y0, t):
54     y = np.zeros((len(t), len(y0)))
55     y[0] = y0
56     for i in range(1, len(t)):
57         y[i] = y[i-1] + (t[i] - t[i-1]) * f(y[i-1])
58     return y.T
59
60 def implicit_euler(f, y0, t, tol=1e-10, max_iter=50):
61     y = np.zeros((len(t), len(y0)))
62     y[0] = y0
63     h = t[1] - t[0]

```

Рисунок 3 – Код часть 2

```

63     h = t[1] - t[0]
64     for i in range(1, len(t)):
65         y_guess = y[i-1] # начальное приближение
66         for _ in range(max_iter):
67             y_new = y[i-1] + h * f(y_guess)
68             if np.linalg.norm(y_new - y_guess) < tol:
69                 break
70             y_guess = y_new
71         y[i] = y_new
72     return y.T
73
74 def runge_kutta_4(f, y0, t):
75     y = np.zeros((len(t), len(y0)))
76     y[0] = y0
77     h = t[1] - t[0]
78     for i in range(1, len(t)):
79         k1 = f(y[i-1])
80         k2 = f(y[i-1] + 0.5 * h * k1)
81         k3 = f(y[i-1] + 0.5 * h * k2)
82         k4 = f(y[i-1] + h * k3)
83         y[i] = y[i-1] + (h / 6.0) * (k1 + 2*k2 + 2*k3 + k4)
84     return y.T
85
86 # Начальное состояние
87 y0 = np.array([x0, v0])
88
89 # Запуск численных методов
90 x_exp_euler = explicit_euler(mass_spring_damper_dynamics, y0, t_vals)
91 x_imp_euler = implicit_euler(mass_spring_damper_dynamics, y0, t_vals)
92 x_rk4 = runge_kutta_4(mass_spring_damper_dynamics, y0, t_vals)

```

Рисунок 4 – Код часть 3

```

94 # -----
95 # ВИЗУАЛИЗАЦИЯ
96 # -----
97 plt.figure(figsize=(15, 4))
98
99 # График 1: Положение
100 plt.subplot(1, 3, 1)
101 plt.plot(t_vals, x_anal, 'k-', linewidth=2, label='Analytical')
102 plt.plot(t_vals, x_exp_euler[0], '--', label='Explicit Euler')
103 plt.plot(t_vals, x_imp_euler[0], '--', label='Implicit Euler')
104 plt.plot(t_vals, x_rk4[0], '--', label='RK4')
105 plt.axhline(y=x_eq, color='gray', linestyle=':', label=r'$x_{\mathrm{eq}} = -mg/k$')
106 plt.xlabel('Time (s)')
107 plt.ylabel('Position $x(t)$ (m)')
108 plt.title('Position vs Time')
109 plt.legend()
110 plt.grid(True)
111
112 # График 2: Скорость
113 plt.subplot(1, 3, 2)
114 plt.plot(t_vals, v_anal, 'k-', linewidth=2, label='Analytical')
115 plt.plot(t_vals, x_exp_euler[1], '--', label='Explicit Euler')
116 plt.plot(t_vals, x_imp_euler[1], '--', label='Implicit Euler')
117 plt.plot(t_vals, x_rk4[1], '--', label='RK4')
118 plt.xlabel('Time (s)')
119 plt.ylabel('Velocity $\dot{x}(t)$ (m/s)')
120 plt.title('Velocity vs Time')
121 plt.legend()
122 plt.grid(True)

```

Рисунок 5 – Код часть 4

```

124 # График 3: Фазовый портрет
125 plt.subplot(1, 3, 3)
126 plt.plot(x_anal, v_anal, 'k-', linewidth=2, label='Analytical')
127 plt.plot(x_exp_euler[0], x_exp_euler[1], '--', label='Explicit Euler')
128 plt.plot(x_imp_euler[0], x_imp_euler[1], '--', label='Implicit Euler')
129 plt.plot(x_rk4[0], x_rk4[1], '--', label='RK4')
130 plt.xlabel('Position $x(t)$ (m)')
131 plt.ylabel('Velocity $\dot{x}(t)$ (m/s)')
132 plt.title('Phase Portrait')
133 plt.legend()
134 plt.grid(True)
135
136 plt.tight_layout()
137 plt.show()
138

```

Рисунок 6 – Код часть 5

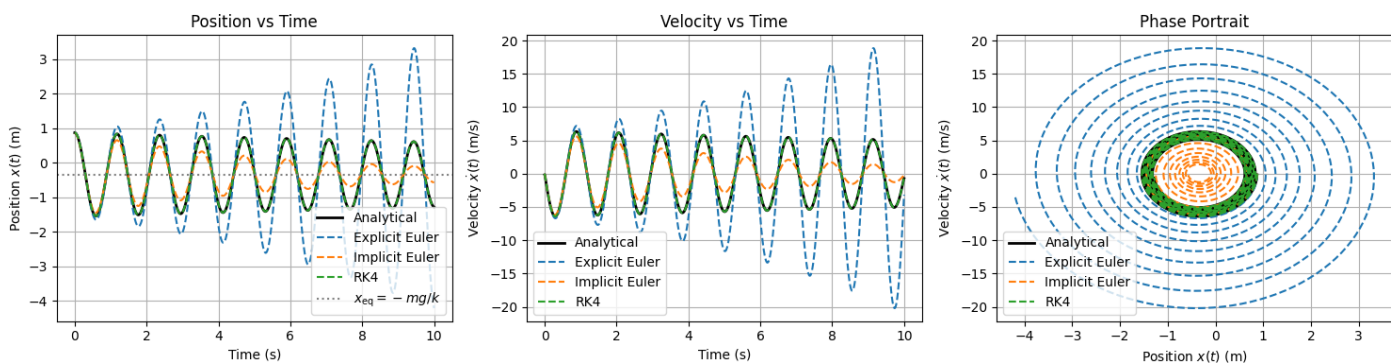


Рисунок 7 – Полученные графики

Вывод: в ходе работы было составлено уравнение движения системы «масса–пружина–демпфер» методом Лагранжа, найдено его аналитическое решение и проведено численное моделирование методами Explicit Euler, Implicit Euler и RK4. Сравнение с аналитическим решением показало, что RK4 обеспечивает наивысшую точность, Implicit Euler – устойчивость при умеренной погрешности, а Explicit Euler – наибольшую ошибку. Все графики подтверждают корректность модели: колебания затухают вокруг равновесного положения $x = -mg/k \approx -0,345\text{м}$, что полностью согласуется с теорией.