

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра механіки, автоматизації та інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

«Інформаційна технологія оцінки параметрів циліндричної гвинтової пружини»

(тема кваліфікаційної роботи українською мовою)

«Information technology for evaluating the parameters of a cylindrical helical spring»

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання спеціальності 126 Інформаційні системи та технології

(код, назва спеціальності)

Освітня програма «Інформаційні системи та технології»

(назва)

Яврумов Олександр Володимирович

(прізвище, ім'я, по-батькові здобувача)

Керівник к.ф.-м.н., доц. Рачинська А.Л.

(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент к.т.н. Косой М.Б.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри

№ ____ від ____ . ____ . 20 ____ р.

Завідувач(ка) кафедри

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК № ____
протокол № __ від ____ . ____ . 20 ____ р.

Оцінка ____ / ____ / ____
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

(підпис)

(прізвище, ім'я)

АНОТАЦІЯ

У дипломній роботі розробляється тема «Інформаційна технологія оцінки параметрів циліндричної гвинтової пружини».

Мета роботи – оцінка параметрів гвинтової циліндричної пружини для заданого переносного руху системи, а саме діаметр пружини та кількість витків пружини.

В результаті в роботі розроблено віртуальну лабораторію з наступним функціоналом:

- для заданих параметрів моделі (безрозмірні коефіцієнти пружності, опору середовища та сили тертя ковзання), переносного руху системи та початкових умов для переносного та відносного рухів системи тіл проводиться візуалізацію руху у тривимірному просторі;
- для заданих параметрів моделі (безрозмірні коефіцієнти опору середовища та сили тертя ковзання), переносного руху системи та початкових умов для переносного та відносного рухів системи тіл проводиться розрахунок коефіцієнту сили пружності для обмежень руху кульки у циліндричному каналі;
- для знайденого коефіцієнту сили пружності проводиться оцінка необхідних діаметру пружини та кількості витків пружини.

ABSTRACT

The topic "Information technology for estimating the parameters of a cylindrical helical spring" is developed in the thesis.

The purpose of the work is to estimate the parameters of the helical cylindrical spring for the given transfer movement of the system, namely the diameter of the spring and the number of turns of the spring.

As a result, the work developed a virtual laboratory with the following functionality:

- for the given parameters of the model (dimensionless coefficients of elasticity, resistance of the environment and the force of sliding friction), the transferable movement of the system and the initial conditions for the transferable and relative movements of the system of bodies, the movement is visualized in three-dimensional space;
- for the given parameters of the model (dimensionless coefficients of medium resistance and sliding friction force), transferable movement of the system and initial conditions for transferable and relative movements of the system of bodies, the coefficient of elastic force is calculated to limit the movement of the ball in the cylindrical channel;
- for the found coefficient of elastic force, the required diameter of the spring and the number of turns of the spring are estimated.

ЗМІСТ

	стор.
ВСТУП	5
1 ДОСЛІДЖЕННЯ СКЛАДНОГО РУХУ СИСТЕМИ ТІЛ	7
1.1 Огляд літератури по темі дипломної роботи	7
1.2 Методи дослідження відносного руху точки.....	16
1.3 Чисельні методи дослідження диференціальних рівнянь руху тіл системи у складному русі.....	18
2 ДОСЛІДЖЕННЯ ВІДНОСНОГО РУХУ КУЛЬКИ У СКЛАДНОМУ РУСІ СИСТЕМИ ТІЛ.....	22
2.1 Математична модель руху.....	22
2.2 Розрахунок жорсткості циліндричної гвинтової пружини та діапазону допустимих значень параметрів пружини.....	31
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ.....	33
3.1 Платформа для розробки системи дослідження	33
3.2 Декомпозиція системи	35
3.3 Реалізація класів	38
3.4 Інформаційний модуль	39
3.5 Модуль візуалізації руху системи тіл	40
3.6 Модуль параметрів	42
4 ФУНКЦІОНАЛ СИСТЕМИ ДОСЛІДЖЕННЯ	44
4.1 Дослідження відносного руху кульки.....	44
4.2 Оцінка параметра коефіцієнта пружності	46
4.3 Оцінка параметрів циліндричної гвинтової пружини	47
ВИСНОВОК.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТОК А. Код інформативного модуля	52
ДОДАТОК Б. Код модуля візуалізації.....	53
ДОДАТОК В. Код модуля параметрів.....	56

ВСТУП

Актуальність теми. Комп'ютерне моделювання є в даний час одним з основних методів дослідження динаміки складних механічних систем. Воно стало невід'ємним етапом проектування конструкцій, оптимізації їх параметрів і широко застосовується в різних галузях науки і техніки, таких як залізничний та автомобільний транспорт, авіабудування, робототехніка та ін.

Програми моделювання реалізують методи побудови та аналізу математичних моделей об'єктів, розроблені на основі узагальнених підходів до опису різноманітних конструктивних елементів, умов їх взаємодії та функціонування. Ефективність математичних моделей визначається припущеннями, які приймаються в рамках таких підходів.

Стрімкий розвиток ІТ-технологій дозволяє застосовувати нові алгоритми вирішення задач, розроблювати віртуальні комп'ютерні лабораторії для моделювання досліджуваних процесів.

Розробка віртуальних лабораторій для моделювання механічних процесів дозволяє проводити дослідження складного руху системи тіл під дією різних силових факторів, а так само їх сукупності. Для кожного силового фактора будується необхідна фізична і математична моделі, а також досліджується питання взаємодії силових факторів. Для узагальнення отриманих результатів необхідно будувати моделі в безрозмірному вигляді, вибираючи в якості масштабу характерних характерні параметри задачі.

В роботі моделюється відносний рух кульки по циліндричному каналу, який здійснює обертальний рух та проводиться оцінка параметрів гвинтової циліндричної пружини, до якої кріпиться кулька в каналі.

Ціль та задачі дослідження. Метою роботи є оцінка параметрів гвинтової циліндричної пружини для заданого переносного руху системи.

Для досягнення поставленою мети необхідно розв'язати наступні задачі:

1. Проаналізувати існуючі моделі складного руху системи тіл.
2. Створити математичну модель відносного руху кульки у циліндричному каналі.
3. Змодельовати віртуальну лабораторію, в якій реалізується тривимірне моделювання складного руху системи тіл.
4. Розробити алгоритм чисельного розрахунку зазначеної моделі для визначення діапазону значень безрозмірного коефіцієнту сили пружності.
5. Провести оцінку діаметру пружини та кількості витків пружини для знайденого коефіцієнту пружності.

Об'єкт дослідження дипломної роботи – відносний рух кульки у циліндричному каналі, який здійснює обертальний рух.

Предмет дослідження – параметри гвинтової циліндричної пружини: діаметр та кількість витків.

1 ДОСЛІДЖЕННЯ СКЛАДНОГО РУХУ СИСТЕМИ ТІЛ

1.1 Огляд літератури по темі дипломної роботи

Фундаментом динаміки систем тіл є класична механіка, основи якої було закладено у 17-18 століттях Ньютоном, Ейлером, Даламбером, Лагранжем. Рівняння руху вільної частки вперше було опубліковано Ньютоном у 1686 році. Поняття "абсолютно тверде тіло" було введено у 1775 р. року Ейлером. Моделюючи зв'язки в шарнірах силами реакцій, він отримав рівняння, відомі в механіці як рівняння Ньютона-Ейлера. У 1743 Даламбер розглянув систему пов'язаних твердих тіл. Маючи на увазі принцип віртуальної роботи, сили реакцій він назвав "втраченими" силами. Математичне формулювання принципу Даламбера представив Лагранж. У 1788 року він започаткував аналіз пов'язаних механічних систем. Застосувавши варіаційний принцип до загальної кінетичної та потенційної енергії системи з урахуванням її кінематичних зв'язків, Лагранж отримав рівняння руху, відомі як рівняння Лагранжа першого та другого роду.

Нехай точка рухається відносно деякої рухомої системи відліку $Oxuz$, яка в свою чергу переміщається по відношенню до основної (нерухомої) системи $\Omega\xi\eta\zeta$. Тоді рух, швидкість і прискорення точки, що розглядаються по відношенню до системи $Oxuz$, називаються відносними, а по відношенню до системи $\Omega\xi\eta\zeta$ - абсолютними. Зрозуміло, що термін «абсолютний» є лише спосіб вираження, що означає, що відповідні величини віднесені до системи $\Omega\xi\eta\zeta$, яка є основною (у цьому ж умовному сенсі основну систему називають нерухомою).

Рух рухомої системи $Oxuz$ по відношенню до нерухомий $\Omega\xi\eta\zeta$ є для рухомої точки переносним рухом, а швидкість і прискорення тієї незмінно пов'язаної з рухомою системою відліку точки простору, в якій зараз знаходиться точка, що рухається, називаються переносними. Інакше

переносну швидкість і переносне прискорення можна в кожний момент часу уявити як ту швидкість і те прискорення, які рухається точка мала б на даний момент, якщо вона починаючи з цього моменту виявилася б жорстко пов'язаною з рухомою системою (тобто не робила відносного руху).

Для вирішення ряду задач механіки необхідно встановити залежності між швидкостями та прискореннями точки по відношенню до рухомий та нерухомий системам відліку. Для швидкостей ця залежність була знайдена геометрично і має вигляд

$$\vec{V}_a = \vec{V}_r + \vec{V}_e. \quad (1.1)$$

Відповідну залежність для прискорень отримаємо, використовуючи поняття про відносної похідної вектору

Нехай рухома $Oxyz$ та нерухома $O\xi\eta\zeta$ системи відліку мають загальний початок O , і нехай ω - миттєва кутова швидкість рухомої системи $Oxyz$ по відношенню до нерухомої (рис. 1.1). Розглянемо точку M , що здійснює рух, який не залежить від руху триєдра $Oxyz$. Її радіус-вектор $\vec{r} = \overrightarrow{OM}$ буде, очевидно, з плином часу змінюватись у кожній із систем відліку за різними законами. Тоді за деякий проміжок часу Δt вектор \vec{r} отримає по відношенню до осей $O\xi\eta\zeta$ і $Oxyz$ різні прирости, які ми відповідно позначимо через $\Delta\vec{r}$ та $\tilde{\Delta}\vec{r}$.

Межі відносин $\Delta\vec{r}$ та $\tilde{\Delta}\vec{r}$ до Δt при $\Delta t \rightarrow 0$ дадуть відповідно похідні

$$\frac{d\vec{r}}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta\vec{r}}{\Delta t} \quad \text{та} \quad \frac{\tilde{d}\vec{r}}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\tilde{\Delta}\vec{r}}{\Delta t}. \quad (1.2)$$

Похідну $\frac{d\vec{r}}{dt}$ називатимемо «абсолютною» або «повною», а похідну $\frac{\tilde{d}\vec{r}}{dt}$

«відносної» або «локальної». Знайдемо залежність між цими похідними. З

цією метою звернемося до кінематичного змісту. З визначень відносної \vec{V}_r і абсолютної \vec{V}_a швидкостей випливає, що

$$\frac{d\vec{r}}{dt} = \vec{V}_r, \quad \frac{d\vec{r}}{dt} = \vec{V}_a. \quad (1.3)$$

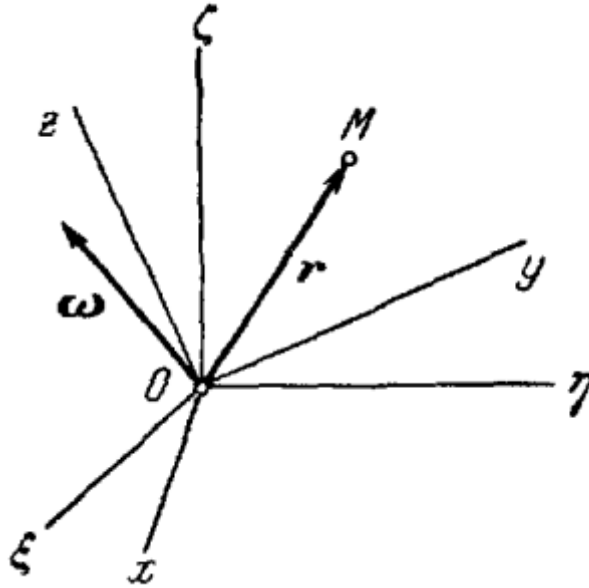


Рисунок 1.1 – Рухома та нерухома системи відліку

Але, згідно (1.1), $\vec{V}_a = \vec{V}_r + \vec{V}_e$, де переносна швидкість \vec{V}_e є швидкість що незмінно пов'язана з триєдром $Oxyz$ точки простору, в якій в даний момент знаходиться точка M . Тоді по формулі Ейлера $\vec{V}_e = \vec{\omega} \times \vec{r}$ і рівність (1.1) дає:

$$\frac{d\vec{r}}{dt} = \frac{d\vec{r}}{dt} + \vec{\omega} \times \vec{r}. \quad (1.4)$$

Так як будь-який вектор $\vec{a} = \vec{a}(t)$, заданий як безперервна і диференційована функція часу, можна розглядати як радіус-вектор деякої

точки (кінця цього вектора), то повна і локальні похідні будь-якого вектора $\vec{a}(t)$ будуть пов'язані тим же співвідношенням

$$\frac{d\vec{a}}{dt} = \frac{\tilde{d}\vec{a}}{dt} + \vec{\omega} \times \vec{r}. \quad (1.5)$$

Якщо скористатися проекціями, то локальна похідна будь-якого вектора \vec{a} щодо системи $Oxyz$ може бути визначена як вектор, проєкції якого на осі цієї системи рівні похідним від проєкції вектора \vec{a} на ті ж осі.

Надамо інший доказ справедливості формули (1.5). Нехай, як завжди, $\vec{i}, \vec{j}, \vec{k}$ суть одиничні координатні векторів рухомого триєдра $Oxyz$, тоді

$$\vec{a} = a_x \vec{i} + a_y \vec{j} + a_z \vec{k}.$$

Диференціюючи за часом, отримаємо:

$$\frac{d\vec{a}}{dt} = \left(\frac{da_x}{dt} \vec{i} + \frac{da_y}{dt} \vec{j} + \frac{da_z}{dt} \vec{k} \right) + \left(a_x \frac{d\vec{i}}{dt} + a_y \frac{d\vec{j}}{dt} + a_z \frac{d\vec{k}}{dt} \right) \quad (1.6)$$

Перші три члени праворуч дають локальну похідну.

$$\frac{da_x}{dt} \vec{i} + \frac{da_y}{dt} \vec{j} + \frac{da_z}{dt} \vec{k} = \frac{\tilde{d}\vec{a}}{dt}, \quad (1.7)$$

так як вони являють собою похідну вектора \vec{a} при умови, що $\vec{i}, \vec{j}, \vec{k}$ постійні. Похідні одиничних векторів суть швидкості їх кінців, т. б. швидкості точок незмінної системи, якою є триєдр $Oxyz$. Отже, за формулою Ейлера

$$\frac{d\vec{i}}{dt} = \vec{\omega} \times \vec{i}, \quad \frac{d\vec{j}}{dt} = \vec{\omega} \times \vec{j}, \quad \frac{d\vec{k}}{dt} = \vec{\omega} \times \vec{k}. \quad (1.8)$$

Рівності (1.8) називають формулами Пуассона. Тоді з виразів (1.8) знаходимо:

$$\frac{d\vec{a}}{dt} = \frac{\tilde{d}\vec{a}}{dt} + a_x \vec{\omega} \times \vec{i} + a_y \vec{\omega} \times \vec{j} + a_z \vec{\omega} \times \vec{k} = \frac{\tilde{d}\vec{a}}{dt} + \vec{\omega} \times (a_x \vec{i} + a_y \vec{j} + a_z \vec{k}),$$

або

$$\frac{d\vec{a}}{dt} = \frac{\tilde{d}\vec{a}}{dt} + \vec{\omega} \times \vec{a}. \quad (1.5')$$

Зауважимо, що формула (1.5) зберігає свій вигляд і в тому випадку, коли тригранник $Oxyz$, крім обертання навколо точки O , здійснює ще й поступальний рух, тобто переміщається як вільне тверде тіло. Насправді, від поступального переміщення триєдра $Oxyz$ одиничні вектори його осей $\vec{i}, \vec{j}, \vec{k}$ не змінюються, отже, формули Пуассона (1.8) зберігають свій вигляд і рівність (1.6) знову призводить до співвідношення (1.5).

Розглянемо окремі випадки.

1) Якщо система $Oxyz$ нерухома, то

$$\vec{\omega} = 0, \quad \frac{d\vec{a}}{dt} = \frac{\tilde{d}\vec{a}}{dt}. \quad (1.9)$$

2) Якщо вектор \vec{a} нерухомий. відношенню до основної системи $\Omega\xi\eta\zeta$, то

$$\frac{d\vec{a}}{dt} = 0, \quad \frac{\tilde{d}\vec{a}}{dt} = -\vec{\omega} \times \vec{a}. \quad (1.10)$$

3) Якщо вектор \vec{a} незмінно пов'язаний з триєдром $Oxyz$, то

$$\frac{d\vec{a}}{dt} = \vec{\omega} \times \vec{a}, \quad \frac{\tilde{d}\vec{a}}{dt} = 0. \quad (1.11)$$

тобто, як і має бути, швидкість кінця \vec{a} визначається в цьому випадку як швидкість точки твердого тіла, скріпленого з трієдром $Oxyz$.

До другого випадку відноситься обчислення відносних похідних від одиничних координатних векторів $\vec{\xi}^0, \vec{\eta}^0, \vec{\zeta}^0$ нерухомої системи $O\xi\eta\zeta$. За формулою (1.10) отримаємо:

$$\frac{\tilde{d}\vec{\xi}^0}{dt} = -\vec{\omega} \times \vec{\xi}^0, \quad \frac{\tilde{d}\vec{\eta}^0}{dt} = -\vec{\omega} \times \vec{\eta}^0, \quad \frac{\tilde{d}\vec{\zeta}^0}{dt} = -\vec{\omega} \times \vec{\zeta}^0. \quad (1.12)$$

Нехай рухома система $Oxyz$ рухається щодо нерухомої $O\xi\eta\zeta$ як вільне тверде тіло. Позначимо швидкість і прискорення початку (полюса) O по відношенню до осей $O\xi\eta\zeta$ через \vec{V}_0 та \vec{W}_0 , а миттєву кутову швидкість і кутове прискорення самого тригранника $Oxyz$ по відношенню до тих же осей $O\xi\eta\zeta$, через $\vec{\omega}$ і $\vec{\varepsilon}$ (Рис. 1.2). Розглянемо точку M , що здійснює рух, який взагалі не залежить від руху системи $Oxyz$. Позначимо через $\vec{\rho}$ і \vec{r} її абсолютний і відносний радіуси-вектори, а через $\vec{\rho}_0$ радіус-вектор точки O . Тоді в будь-який момент часу

$$\vec{\rho} = \vec{\rho}_0 + \vec{r}. \quad (1.15)$$

Візьмемо від обох частин цієї рівності повну похідну по часу. Враховуючи формули (1.5) та (1.3), матимемо:

$$\vec{V}_a = \frac{d\vec{\rho}}{dt} = \frac{d\vec{\rho}_0}{dt} + \frac{d\vec{r}}{dt} = \vec{V}_0 + \frac{\tilde{d}\vec{r}}{dt} + \vec{\omega} \times \vec{r} = \vec{V}_0 + \vec{\omega} \times \vec{r} + \vec{V}_r. \quad (1.16)$$

Але за формулою (1.1) $\vec{V}_0 + \vec{\omega} \times \vec{r}$ є швидкість незмінно пов'язаної з системою $Oxyz$ точки, в якій в даний момент знаходиться точка M , отже, за визначенням це — переносна швидкість, тобто

$$\vec{V}_0 + \vec{\omega} \times \vec{r} = \vec{V}_e. \quad (1.17)$$

В результаті з (1.16) знаходимо:

$$\vec{V}_a = \vec{V}_r + \vec{V}_e. \quad (1.18)$$

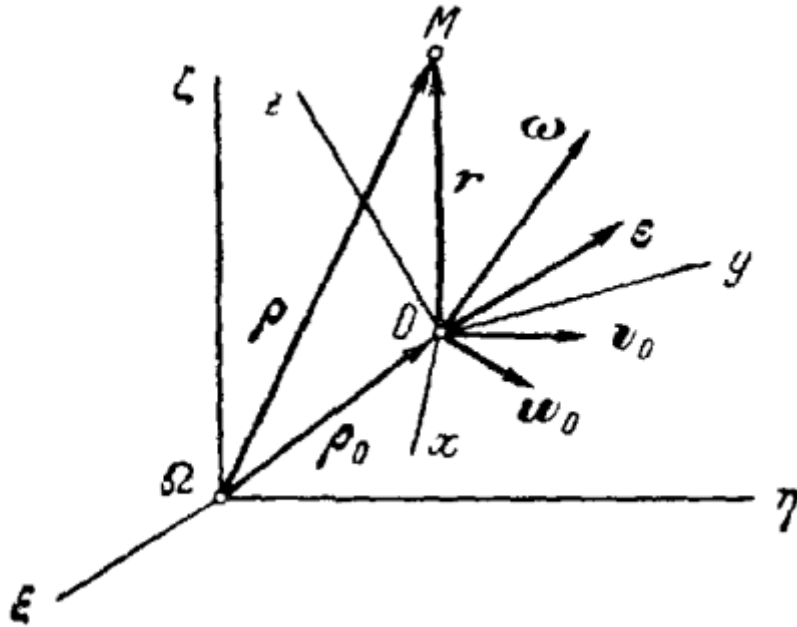


Рисунок 1.2 – Складання прискорень

Таким чином, ми іншим шляхом довели теорему про складання швидкостей.

Беручи тепер похідні від обох частин рівності (1.18), будемо мати:

$$\vec{W}_a = \frac{d\vec{V}_a}{dt} = \frac{d\vec{V}_r}{dt} + \frac{d\vec{V}_e}{dt},$$

або після заміни \vec{V}_e його значенням з (1.17),

$$\vec{W}_a = \frac{d\vec{V}_r}{dt} + \frac{d}{dt}(\vec{V}_0 + \vec{\omega} \times \vec{r}) = \frac{d\vec{V}_0}{dt} + \frac{d\vec{\omega}}{dt} \times \vec{r} + \vec{\omega} \times \frac{d\vec{r}}{dt} + \frac{d\vec{V}_r}{dt}.$$

Застосовуємо формулу (1.15) \vec{r} та \vec{V}_r , отримуємо

$$\vec{W}_a = \frac{d\vec{V}_0}{dt} + \frac{d\vec{\omega}}{dt} \times \vec{r} + \vec{\omega} \times \left(\frac{d\vec{r}}{dt} + \vec{\omega} \times \vec{r} \right) + \frac{d\vec{V}_r}{dt} + \vec{\omega} \times \vec{V}_r.$$

В цьому виразу

$$\frac{d\vec{V}_0}{dt} = \vec{W}_0, \quad \frac{d\vec{\omega}}{dt} = \vec{\varepsilon}, \quad \frac{d\vec{r}}{dt} = \vec{V}_r,$$

та його можна представити у вигляді

$$\vec{W}_a = \vec{W}_0 + \vec{\varepsilon} \times \vec{r} + \vec{\omega} \times \vec{V}_r + \vec{\omega} \times (\vec{\omega} \times \vec{r}) + \frac{d\vec{V}_r}{dt} + \vec{\omega} \times \vec{V}_r$$

або, після приведення,

$$\vec{W}_a = \frac{d\vec{V}_r}{dt} + \vec{W}_0 + \vec{\varepsilon} \times \vec{r} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) + 2(\vec{\omega} \times \vec{V}_r).$$

Розглянемо, що являють собою доданки, що входять до правої частини рівності (1.19). Величина

$$\vec{W}_r = \frac{d\vec{V}_r}{dt} = \frac{d^2\vec{r}}{dt^2}. \quad (1.20)$$

є за визначенням відносне прискорення (як локальна похідна від відносної швидкості за часом). Іншим шляхом у цьому можна переконатися, поклавши (1.19) $\omega = 0$, $\varepsilon = 0$, $W_0 = 0$, тобто вважаючи осі $Oxyz$ нерухомими. Тоді повне прискорення точки M має збігтися з відносним і ми прийдемо до рівності (1.20).

Величина

$$\vec{W}_a = \vec{W}_0 + \vec{\varepsilon} \times \vec{r} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) \quad (1.21)$$

є переносне прискорення, оскільки вона дорівнює прискоренню тієї незмінно пов'язаної із системою $Oxyz$ точки, в якій в даний момент знаходиться точка M .

Величина

$$\vec{W}_c = 2\vec{\omega} \times \vec{V}_r, \quad (1.21)$$

яка не входить ні до відносного, ні до переносного прискорення, називається поворотним чи коріоласовим прискоренням.

В результаті отримуємо наступну теорему про складання прискорення прискорень або теорему Коріоліса: абсолютне прискорення точки при складному русі дорівнює геометричній сумі відносного, переносного та коріолісового прискорень

$$\vec{W}_a = \vec{W}_r + \vec{W}_e + \vec{W}_c. \quad (1.23)$$

1.2 Методи дослідження відносного руху точки

Питання про відносний рух матеріальної точки тісно стикається з основними ідеями механіки. Будь-який рух точки (або тіла) ми повинні розглядати щодо деякої системи відліку. До цих пір ми вивчали рух по відношенню до так званої інерційної системи відліку, тобто система відліку, в якій справедливі основні закони динаміки та по відношенню до якої матеріальна точка, на яку ніякі сили не діють, рухається за інерцією (рівномірно та прямолінійно).

Інерційну систему відліку називають ще умовно нерухомою, а рух по відношенню до неї - абсолютним. Зараз ми перейдемо до розгляду руху точки щодо по відношенню до систем відліку, що завгодно переміщуються по відношенню до інерційної системи відліку. Такий рух точки називають відносним.

Розглянемо матеріальну точку M , на яку діє сила F , що є результатом взаємодії цієї точки з іншими матеріальними тілами. Складемо рівняння руху цієї точки по відношенню до системи відліку $Axuz$, що довільно переміщується щодо інерційної системи відліку $Bx_1y_1z_1$ (рис. 1.3). Відомо, що прискорення точки M щодо основної системи знаходиться за формулою (1.23).

Рівняння руху точки M щодо інерційної системи відліку B , згідно з основним законом динаміки, буде

$$m\vec{W}_a = \vec{F}. \quad (1.24)$$

Замінюючи \vec{W}_a його виразом з рівності (1.23), отримуємо

$$m(\vec{W}_r + \vec{W}_e + \vec{W}_c) = \vec{F}.$$

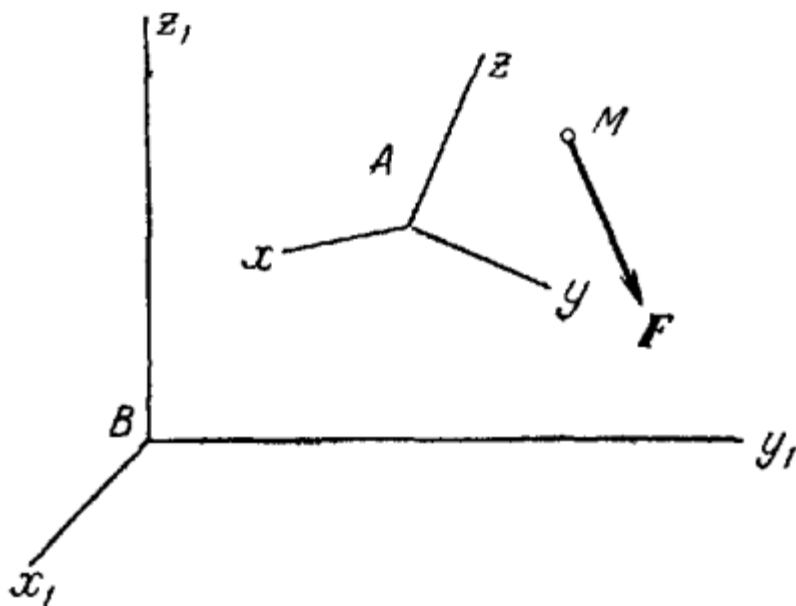


Рисунок 1.3 – Інерційні та неінерційні системи відліку

Звідки

$$m\vec{W}_r = \vec{F} + (-m\vec{W}_e) + (-m\vec{W}_c). \quad (1.25)$$

Введемо позначення

$$\vec{\Phi}_e = -m\vec{W}_e, \quad \vec{\Phi}_c = -m\vec{W}_c. \quad (1.26)$$

Величини $\vec{\Phi}_e$ і $\vec{\Phi}_c$, що мають розмірність сили, назвемо відповідно переносною та коріолісовою силами інерції. Тоді з рівності (1.25) отримаємо

$$m\vec{W}_r = \vec{F} + \vec{\Phi}_e + \vec{\Phi}_c. \quad (1.27)$$

Рівняння (1.27) являє собою у векторній формі рівняння відносного руху точки (по відношенню до рухомої системі відліку А). Порівнюючи між собою (1.27) та (1.24), укладаємо, що рівняння відносного руху точки можна

становити так само, як рівняння абсолютного руху, якщо до діючих на точку силам взаємодії з іншими матеріальними тілами додати переносну та коріолісову силу інерції.

Проектуючи обидві частини рівності (1.27) на осі $Axuz$, отримаємо диференціальні рівняння відносного руху точки в проекціях на прямокутні декартові осі координат

$$\begin{aligned} m \frac{d^2 x}{dt^2} &= F_x + \Phi_{ex} + \Phi_{cx}, \\ m \frac{d^2 y}{dt^2} &= F_y + \Phi_{ey} + \Phi_{cy}, \\ m \frac{d^2 z}{dt^2} &= F_z + \Phi_{ez} + \Phi_{cz}. \end{aligned} \quad (1.28)$$

Як видно з рівнянь (1.27) або (1.28), введення сил інерції $\vec{\Phi}_e$ та $\vec{\Phi}_c$ дозволяє при вивченні відносного руху складати рівні рівняння руху точки в неінерційній системі відліку в тій же формі, що має місце для інерційної системи. Іншими словами, за допомогою сил $\vec{\Phi}_e$ і $\vec{\Phi}_c$ враховується вплив руху рухомий системи відліку щодо відносного руху точки.

1.3 Чисельні методи дослідження диференціальних рівнянь руху тіл системи у складному русі

Для розв'язання диференціального рівняння (1.27) необхідно використовувати чисельні методи. Найчастіше використовуваним для інтегрування звичайних диференціальних рівнянь є метод Рунге-Кутта четвертого порядку при $h = const$. Його перевагою є висока точність (похибка h^6) та несильна схильність до виникнення нестійкості рішення. Цей метод найчастіше використовується як метод розв'язання системи диференціальних рівнянь першого порядку.

Нехай є задача Коші для системи виду

$$\begin{aligned} \frac{dy_1}{dx} &= F_1(x, y_1, \dots, y_n), \\ &\dots \\ \frac{dy_i}{dx} &= F_i(x, y_1, \dots, y_n), \\ &\dots \\ \frac{dy_n}{dx} &= F_n(x, y_1, \dots, y_n), \end{aligned} \tag{1.29}$$

де $i = 1, \dots, n$ - номер кожної залежної змінної y_i , x - незалежна змінна. Розв'язок системи (1.29) для заданих початкових умов $x = x_0$, $y_i(x_0) = y_{i0}$ ($i = 1, \dots, n$) зводиться до знаходження залежностей (інтегральних кривих) $y_i(x)$ ($i = 1, \dots, n$), що проходять через точки, які задані початковими умовами (x_0, y_{i0}) .

Узагальнена форма запису кожного із рівнянь системи (1.29) може бути представлена у загальному вигляді

$$\frac{dY_i}{dx} = F_i(x, Y_i), \tag{1.30}$$

де Y_i у правій частині рівняння – вектори змінних y_i , а F_i - права частина кожного із рівнянь (1.29). Зокрема, одне диференціальне рівняння ($y = Y_i = Y_1$ і $F_i = F = F_1$) і записується у вигляді

$$\frac{dy}{dx} = F_1(x, y). \tag{1.31}$$

Диференціальне рівняння вищого порядку

$$y^{(n)} = F(x, y, y', y'', \dots, y^{(n-1)}), \quad (1.32)$$

де (n) - порядок рівняння може бути зведено до системи виду (1.29) або (1.30) за допомогою таких перетворень:

$$\begin{aligned} \frac{dy}{dx} &= y_1, \\ \frac{dy_1}{dx} &= y_2, \\ &\dots \\ \frac{dy_{n-2}}{dx} &= y_{n-1}, \\ \frac{dy_{n-1}}{dx} &= F(x, y, y_1, y_2, \dots, y_{n-1}). \end{aligned} \quad (1.33)$$

Алгоритм методу Рунге-Кутта полягає у циклічних обчисленнях $Y_{i(j+1)}$ на кожному $j + 1$ кроку за такими формулами:

$$\begin{aligned} K_{1i} &= hF_i(x_j, Y_{ij}), \quad K_{2i} = hF_i\left(x_j + \frac{h}{2}, Y_{ij} + \frac{1}{2}K_{1i}\right), \\ K_{3i} &= hF_i\left(x_j + \frac{h}{2}, Y_{ij} + \frac{1}{2}K_{2i}\right), \quad K_{4i} = hF_i(x_j + h, Y_{ij} + K_{3i}), \\ Y_{i(j+1)} &= Y_{ij} + \frac{1}{6}(K_{1i} + 2K_{2i} + 2K_{3i} + K_{4i}). \end{aligned} \quad (1.34)$$

Зважаючи на особливе значення та широке застосування диференціальних рівнянь другого порядку корисно мати метод Рунге-Кутта розв'язання таких рівнянь.

Є рівняння виду

$$y'' = \frac{d^2 y}{dx^2} = F(x, y, y'). \quad (1.35)$$

Метод має формули:

$$\begin{aligned} K_1 &= hF\left(x_j, y_j, y'_j\right), \\ K_2 &= hF\left(x_j + \frac{h}{2}, y_j + \frac{h}{2}y'_j + \frac{h}{8}K_1, y'_j + \frac{K_1}{2}\right), \\ K_3 &= hF\left(x_j + \frac{h}{2}, y_j + \frac{h}{2}y'_j + \frac{h}{8}K_2, y'_j + \frac{K_2}{2}\right), \\ K_4 &= hF\left(x_j + h, y_j + hy'_j + \frac{h}{2}K_3, y'_j + K_3\right), \\ y_{j+1} &= y_j + h\left[y'_j + \frac{1}{6}(K_1 + K_2 + K_3)\right], \\ y'_{j+1} &= y'_j + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4). \end{aligned} \quad (1.36)$$

Слід пам'ятати, що з написанні функцій чисельного рішення обов'язковим етапом є тестування функцій.

2 ДОСЛІДЖЕННЯ ВІДНОСНОГО РУХУ КУЛЬКИ У СКЛАДНОМУ РУСІ СИСТЕМИ ТІЛ

2.1 Математична модель руху

Кулька M , що розглядається як матеріальна точка, переміщається по циліндричному каналу тіла A , що рухається необхідно визначити закон відносного руху кульки $x = f(t)$, прийнявши за початок відліку точку O . Тіло A рівномірно обертається навколо нерухомої осі z_1 .

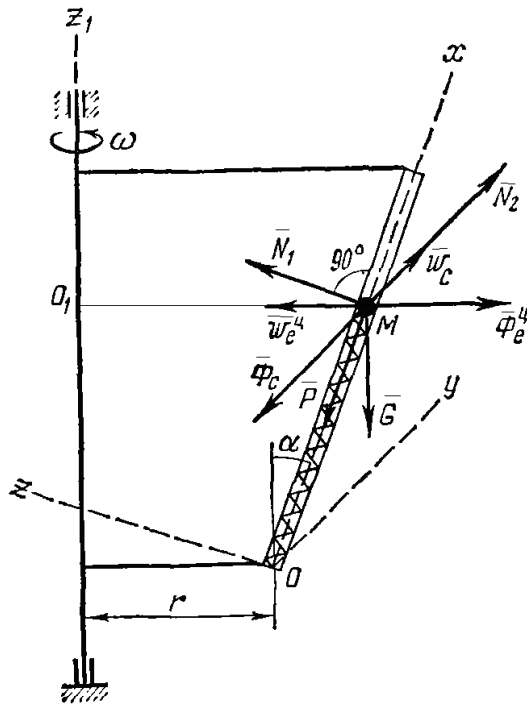


Рисунок 2.1 – Схематичне зображення складного руху точки

Зв'яжемо рухливу систему відліку $Oxyz$ з обертовим каналом (трубкою), сумістивши вісь x з траєкторією відносного руху кульки M .

Обертання цієї системи навколо осі z_1 є переносним рухом для кульки M . Відносним рухом кульки M є його рух уздовж трубки.

Для розробки необхідної математичної моделі використовуємо відомості з теоретичної механіки.

Відносний рух точки визначається рівнянням (1.27)

$$m\vec{W}_r = \sum \vec{P}_i + \vec{\Phi}_e + \vec{\Phi}_c,$$

де $\sum \vec{P}_i$ - сили, що діють на кульку M : вага \vec{G} , реакція пружини \vec{P} , та нормальна реакція стінки трубки; цю реакцію можна розкласти на дві взаємно перпендикулярні складові \vec{N}_1 і \vec{N}_2 у площині перерізу трубки, сила тертя ковзання \vec{F}_f , а також сила опору \vec{F}_r .

Напрямок сил тертя ковзання та опору має бути завжди протилежним напрямку швидкості кульки. Вважаємо, що в момент часу, що розглядається, кулька рухається вздовж осі x , тоді сили тертя ковзання \vec{F}_f та сила опору \vec{F}_r спрямовані протилежно до осі x .

Приєднуємо до сил, що діють на кульку M , переносну силу інерції $\vec{\Phi}_e$ і коріолісову силу інерції $\vec{\Phi}_c$, спрямовані протилежно прискорень \vec{W}_e і \vec{W}_c відповідно.

Основне рівняння відносного руху в даному випадку має вигляд:

$$m\vec{W}_r = \vec{G} + \vec{P} + \vec{N}_1 + \vec{N}_2 + \vec{\Phi}_e + \vec{\Phi}_c + \vec{F}_f + \vec{F}_r.$$

Переносна сила інерції $\vec{\Phi}_e$ визначається за формулою

$$\vec{\Phi}_e = -m\vec{W}_e.$$

Оскільки переносний рух є обертальним, то

$$\vec{W}_e = \vec{W}_e^{ob} + \vec{W}_e^{\partial\omega},$$

де $\vec{W}_e^{ob} = \vec{\varepsilon} \times \vec{O}_1 M$ - обертальна складова переносного прискорення,

$\vec{W}_e^{\partial\omega} = -\omega_e^2 \vec{O}_1 M$ - доцентрова складова переносного прискорення. У

поточній постановці завдання тіло A здійснює нерівномірне обертання

$$\omega = \omega(t), \text{ то } \varepsilon = \frac{d\omega}{dt}.$$

Маємо, що вектор $\vec{\Phi}_e = \vec{\Phi}_e^{\partial\omega} + \vec{\Phi}_e^{ob}$ і $\Phi_e^{\partial\omega} = m\omega^2(r + x \sin \alpha)$,
 $\Phi_e^{ob} = m\varepsilon(r + x \sin \alpha)$. Припустимо, що $\varepsilon > 0$, тоді вектор $\vec{\Phi}_e^{ob}$ напрямку
 вздовж осі y на нас.

Коріолісова сила інерції $\vec{\Phi}_c$ визначається за формулою

$$\vec{\Phi}_c = -m\vec{W}_c,$$

де коріолісове прискорення знаходиться за формулою $\vec{W}_c = 2\vec{\omega}_e \times \vec{V}_r$.

Напрямок вектора прискорення знайдемо за відомим правилом Жуковського, припустивши, що проекція відносної швидкості \vec{V}_r на вісь x позитивна. У цьому прикладі коріолісова сила інерції $\vec{\Phi}_c$ паралельна осі y і перпендикулярна до площини xOz .

Модуль сили інерції визначається за формулою

$$\Phi_c = 2m\omega_e V_r \sin \alpha,$$

де $\omega_e = \omega$ и $V_r = \dot{x}$.

Проектуючи цю векторну рівність на три осі рухомої системи координат маємо:

$$\begin{aligned}
 x: m\ddot{x} &= \Phi_e^{\partial y} \sin \alpha - G \cos \alpha - P - F_r - F_f, \\
 y: 0 &= N_2 - \Phi_c - \Phi_e^{\partial \phi}, \\
 z: 0 &= N_1 - \Phi_e^{\partial y} \cos \alpha - G \sin \alpha.
 \end{aligned} \tag{2.1}$$

Розглянемо дві моделі сили опору середовища:

2. Сила опору пропорційна лінійній швидкості руху кульки вздовж трубки

$$F_r = \mu V = \mu \dot{x}.$$

3. Сила опору пропорційна квадрату лінійної швидкості руху кульки вздовж трубки $F_r = \mu V^2 = \mu \dot{x}^2$.

Реакція пружини P дорівнює добутку коефіцієнта жорсткості c на величину деформації пружини $x - l_0$ (l_0 - довжина недеформованої пружини).

Величина сили тертя ковзання визначається як $F_f = fN$, где f - коефіцієнт сили тертя ковзання, який задаватиметься, як параметр моделі, а $N = \sqrt{N_1^2 + N_2^2}$.

З другого та третього рівнянь системи (2.1) маємо

$$N_2 = \Phi_c + \Phi_e^{\partial \phi}, \tag{2.2}$$

$$N_1 = \Phi_e^{\partial y} \cos \alpha + G \sin \alpha.$$

Тоді диференціальне рівняння руху має вигляд: для першої моделі з урахуванням сили опору пропорційної швидкості

$$\begin{aligned}
 \ddot{x} &= \omega^2 (r + x \sin \alpha) \sin \alpha - g \cos \alpha - \frac{c}{m} (x - l_0) - \frac{\mu \dot{x}}{m} - \\
 &- f \operatorname{sign}(\dot{x}) \sqrt{(2\omega \dot{x} \sin \alpha)^2 + \left(\omega^2 (r + x \sin \alpha) \cos \alpha + g \sin \alpha \right)^2},
 \end{aligned} \tag{2.3}$$

з урахуванням сили опору пропорційної квадрату швидкості

$$\ddot{x} = \omega^2 (r + x \sin \alpha) \sin \alpha - g \cos \alpha - c(x - l_0) - \frac{\mu \text{sign}(\dot{x}) \dot{x}^2}{m} - f \text{sign}(\dot{x}) \sqrt{(2\omega \dot{x} \sin \alpha)^2 + \left(\omega^2 (r + x \sin \alpha) \cos \alpha + g \sin \alpha \right)^2}. \quad (2.4)$$

Для безрозмірювання рівняння необхідно ввести характерні параметри задачі. Характерним параметром лінійного розміру буде довжина недеформованої пружини, тоді безрозмірна координата визначається рівністю

$$\tilde{x} = \frac{x}{l_0}.$$

Для безрозмірювання часу використовуватимемо комбінацію характерних параметрів задачі

$$\tau = \sqrt{\frac{l_0}{g}}.$$

Розділимо рівняння на довжину недеформованої пружини:

$$\frac{\ddot{x}}{l_0} = \omega^2 \left(\frac{r}{l_0} + \frac{x}{l_0} \sin \alpha \right) \sin \alpha - \frac{g}{l_0} \cos \alpha - \frac{c}{m} \left(\frac{x}{l_0} - 1 \right) - \frac{\mu \dot{x}}{m l_0} - f \sqrt{\left(\frac{2\omega \dot{x} \sin \alpha}{l_0} \right)^2 + \left(\frac{\omega^2 (r + x \sin \alpha) \cos \alpha}{l_0} + \frac{g \sin \alpha}{l_0} \right)^2}.$$

Введемо в рівняння безрозмірну координату та помножимо рівняння на величину l_0/g :

$$\frac{d^2 \tilde{x} l_0}{dt^2 g} = \omega^2 \frac{l_0}{g} \left(\frac{r}{l_0} + \tilde{x} \sin \alpha \right) \sin \alpha - \frac{g l_0}{l_0 g} \cos \alpha - \frac{c l_0}{m g} (\tilde{x} - 1) - \frac{\mu}{m} \frac{d\tilde{x} l_0}{dt g} -$$

$$- f \sqrt{\left(2\omega \sin \alpha \frac{d\tilde{x} l_0}{dt g} \right)^2 + \left(\omega^2 \left(\frac{l_0}{g} \right)^2 \left(\frac{r}{l_0} + \tilde{x} \sin \alpha \right) \cos \alpha + \frac{g \sin \alpha l_0}{l_0 g} \right)^2}.$$

Введемо до рівняння величину безрозмірного часу:

$$\frac{d^2 \tilde{x}}{d\tau^2} = \tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \sin \alpha - \cos \alpha - \frac{c l_0}{m g} (\tilde{x} - 1) - \frac{\mu}{m} \sqrt{\frac{l_0}{g}} \frac{d\tilde{x}}{d\tau} -$$

$$- f \sqrt{\left(2\tilde{\omega} \sin \alpha \frac{d\tilde{x}}{d\tau} \right)^2 + \left(\tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha \right)^2}.$$

Введено величини безрозмірної кутової швидкості

$$\tilde{\omega} = \omega \sqrt{\frac{l_0}{g}}$$

та безрозмірна геометрична характеристика тіла А

$$\tilde{r} = \frac{r}{l_0}.$$

Розглянемо комбінації параметрів задачі та визначимо їх розмірність

$$\left[\frac{c l_0}{m g} \right] = \frac{\text{кг} / \text{с}^2 \cdot \text{м}}{\text{кг} \cdot \text{м} / \text{с}^2} \quad \text{та} \quad \left[\frac{\mu}{m} \sqrt{\frac{l_0}{g}} \right] = \frac{\text{кг} / \text{с} \cdot \text{м}^{0.5}}{\text{кг} \cdot \text{м}^{0.5} / \text{с}}.$$

Введемо два безрозмірні коефіцієнти

$$\chi_1 = \frac{cl_0}{mg} \quad \text{та} \quad \chi_2 = \frac{\mu}{m} \sqrt{\frac{l_0}{g}}.$$

Тоді рівняння (2.3) у безрозмірному вигляді

$$\begin{aligned} \frac{d^2 \tilde{x}}{d\tau^2} = & \tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \sin \alpha - \cos \alpha - \chi_1 (\tilde{x} - 1) - \chi_2 \frac{d\tilde{x}}{d\tau} - \\ & - f \sqrt{\left(2\tilde{\omega} \sin \alpha \frac{d\tilde{x}}{d\tau} \right)^2 + \left(\tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha \right)^2}. \end{aligned} \quad (2.5)$$

Аналогічно проведемо обезрозмірювання рівняння (2.4)

$$\begin{aligned} \frac{d^2 \tilde{x}}{d\tau^2} = & \tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \sin \alpha - \cos \alpha - \chi_1 (\tilde{x} - 1) - \chi_3 \text{sign}(\dot{\tilde{x}}) \left(\frac{d\tilde{x}}{d\tau} \right)^2 - \\ & - f \sqrt{\left(2\tilde{\omega} \sin \alpha \frac{d\tilde{x}}{d\tau} \right)^2 + \left(\tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha \right)^2}, \end{aligned} \quad (2.6)$$

де

$$\chi_3 = \frac{\mu l_0}{m}.$$

Безрозмірні коефіцієнти характеризують вплив різних фізичних чинників на рух матеріальної точки. Коефіцієнт χ_1 характеризує вплив сили пружності, коефіцієнти χ_2 і χ_3 - вплив сили опору.

Проведемо обезрозмірювання рівнянь знаходження сили реакції зв'язку.

Нерівномірне обертання

$$N_1 = m\omega^2(r + x \sin \alpha) \cos \alpha + mg \sin \alpha, \quad (2.7)$$

$$N_2 = m(\sin \alpha(2\omega\dot{x} + \varepsilon x) + \varepsilon r). \quad (2.8)$$

Для характерного розміру сили оберемо силу тяжіння mg , тому розділимо кожен рівність (2.7) та (2.8) на mg .

$$\frac{N_1}{mg} = \frac{\omega^2(r + x \sin \alpha) \cos \alpha}{g} + \sin \alpha,$$

$$\frac{N_2}{mg} = \frac{(\sin \alpha(2\omega\dot{x} + \varepsilon x) + \varepsilon r)}{g}.$$

Помножимо кожен дріб правої частини на довжину недеформованої пружини

$$\frac{N_1}{mg} = \frac{\omega^2(r + x \sin \alpha) \cos \alpha l_0}{gl_0} + \sin \alpha,$$

$$\frac{N_2}{mg} = \frac{(\sin \alpha(2\omega\dot{x} + \varepsilon x) + \varepsilon r)l_0}{gl_0}.$$

Якщо обезрозмірені величини позначити тими самими символами, але з хвилею, маємо

$$\tilde{N}_1 = \tilde{\omega}^2(\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha,$$

$$\tilde{N}_2 = (\sin \alpha(2\tilde{\omega}\tilde{\dot{x}} + \tilde{\varepsilon}\tilde{x}) + \tilde{\varepsilon}\tilde{r}).$$

Таким чином, побудовано дві моделі руху матеріальної точки. Перша модель:

$$\begin{aligned}
\frac{d^2 \tilde{x}}{d\tau^2} &= \tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \sin \alpha - \cos \alpha - \chi_1 (\tilde{x} - 1) - \chi_2 \frac{d\tilde{x}}{d\tau} - \\
&- f \sqrt{\left(2\tilde{\omega} \sin \alpha \frac{d\tilde{x}}{d\tau} \right)^2 + \left(\tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha \right)^2}, \\
\chi_1 &= \frac{cl_0}{mg}, \quad \chi_2 = \frac{\mu}{m} \sqrt{\frac{l_0}{g}}, \\
\tilde{N}_1 &= \tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha, \\
\tilde{N}_2 &= \left(\sin \alpha (2\tilde{\omega} \dot{\tilde{x}} + \tilde{\varepsilon} \tilde{x}) + \tilde{\varepsilon} \tilde{r} \right).
\end{aligned} \tag{2.9}$$

Друга модель:

$$\begin{aligned}
\frac{d^2 \tilde{x}}{d\tau^2} &= \tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \sin \alpha - \cos \alpha - \chi_1 (\tilde{x} - 1) - \chi_3 \operatorname{sign}(\dot{\tilde{x}}) \left(\frac{d\tilde{x}}{d\tau} \right)^2 - \\
&- f \sqrt{\left(2\tilde{\omega} \sin \alpha \frac{d\tilde{x}}{d\tau} \right)^2 + \left(\tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha \right)^2}, \\
\chi_1 &= \frac{cl_0}{mg}, \quad \chi_3 = \frac{\mu l_0}{m}, \\
\tilde{N}_1 &= \tilde{\omega}^2 (\tilde{r} + \tilde{x} \sin \alpha) \cos \alpha + \sin \alpha, \\
\tilde{N}_2 &= \left(\sin \alpha (2\tilde{\omega} \dot{\tilde{x}} + \tilde{\varepsilon} \tilde{x}) + \tilde{\varepsilon} \tilde{r} \right).
\end{aligned} \tag{2.10}$$

Рівняння обертального руху твердого тіла відносно нерухомої вісі має вигляд:

$$I_z \frac{d\omega}{dt} = M_z. \tag{2.11}$$

Проведемо безрозмірювання рівняння (2.11)

$$\frac{d\tilde{\omega}}{d\tau} = \frac{M_z l_0}{I_z g}. \quad (2.12)$$

2.2 Розрахунок жорсткості циліндричної гвинтової пружини та діапазону допустимих значень параметрів пружини

Для розрахунку коефіцієнта жорсткості пружини фахівці орієнтуються на формулу

$$c = \frac{Gd^4}{8D^3n}, \quad (2.13)$$

де G – визначає модуль зсуву (властивість залежить, наприклад, від використовуваної сировини); d – діаметр шматка дроту (величина визначається в період виробництва шляхом прокату, а результат записується в технічній документації); D – діаметр витків, які виходять в результаті намотування на дріт (розрахунок здійснюється з урахуванням поставлених завдань і залежить від навантаження, що створюється для стиснення об'єкта); n – кількість витків в системі (показник варіюється в значному діапазоні, від чого залежать експлуатаційні характеристики предмета).

Для розрахунку моделей необхідно задавати початкові значення безрозмірних координат, кутової швидкості, геометричної характеристики тіла A , а також коефіцієнти χ_1 , χ_2 або χ_3 та f . Для комп'ютерного моделювання руху системи слід використовувати отримані результати. Однак не всі отримані математичні результати мають фізичний зміст.

Вісь x має початок відліку в т. O (рис. 1.2), а отже координата відносного руху точки не може приймати від'ємне значення. З іншого боку, в трубці є пружина, яка під час руху матеріальної точки у бік т. O стискатиметься. Природно, навіть при максимальному стисканні пружини вона не може займати в трубці нульову довжину, щоб тіло досягло хоча б координати 0 . У разі побудови аналітичного рішення завдання було отримано умову на початкові умови в припущенні, що пружина може стиснутися тільки на половину своєї величини, т. б. мінімальна довжина пружини дорівнює $l_0 / 2$, а значить мінімальна величина координати тіла у відносному русі має бути $l_0 / 2$. У наших моделях немає можливості побудови аналітичного рішення. Безрозмірна координата \tilde{x} відносного руху точки повинна мати значення більше $0,5$ і не перевищувати 10 .

$$0.5 < \tilde{x} < 10. \quad (2.14)$$

Для розрахунку діапазону коефіцієнту сили пружності необхідно зафіксувати значення всіх інших безрозмірних параметрів та підібрати необхідний діапазон коефіцієнту χ_1 .

Знаючи коефіцієнт χ_1 , матеріал з якого виконано пружину, діаметр проволочи підбираємо область допустимих значень для діаметру витка та кількості витків.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ

3.1 Платформа для розробки системи дослідження

В дипломній роботі для розробки системи дослідження складного руху тіла обрано .NET Framework. Розробка середовищі Visual Studio. Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms.

Windows Forms - інтерфейс програмування додатків (API), що відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за рахунок створення обгортки для існуючого Win32 API в керованому коді. Причому керований код - класи, що реалізують API для Windows Forms, що не залежать від мови розробки.

Застосування Windows Forms є подієво-орієнтованим додатком Microsoft .NET Framework. На відміну від пакетних програм, велика частина часу витрачається на очікування від користувача будь-яких дій, як, наприклад, введення тексту в текстове поле або кліка мишкою по кнопці.

Для реалізації інформаційних технологій дослідження використовується мова програмування C#. Це об'єктно-орієнтована мова програмування. C# відноситься до сім'ї мов з C-подібним синтаксисом, з них його синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, змінні, властивості, узагальнені типи і методи, ітератори, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі в форматі XML.

Переїнявши багато від своїх попередників - мов C ++, Delphi, Модула, Smalltalk і, особливо, Java - C#, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, C# на відміну від C++ не підтримує множинне успадкування класів (між тим допускається множинна реалізація інтерфейсів).

Складовими системи дослідження буде технологія тривимірної побудови моделі тіла та технологія тривимірної побудови годографу вектору кінетичного моменту. В роботі використано пакет DirectX.

Microsoft DirectX — це набір програмних інтерфейсів, використовуваних для керування поруч апаратних компонентів комп'ютера. До складу цих інтерфейсів входять: Directdraw — для швидкого доступу до відеопам'яті, Directsound — для висновку аудіоінформації на звукову карту, Directplay — для організації багатокористувацької роботи через модем, локальну мережу або Internet, Directinput — для обробки введення інформації із клавіатури, від миші або джойстика й Direct3D — ядро підтримки тривимірної графіки, використовуване разом з Directdraw.

Основна мета, яку переслідувала фірма Microsoft, створюючи інтерфейс DirectX, — перетворити комп'ютери, що працюють під керуванням операційної системи Windows, в ідеальну платформу для додатків, багатих мультимедійними елементами: повнобарвною графікою, відеофрагментами, тривимірною анімацією й стереозвуком.

Для розроблювачів Directx являє собою набір програмних інтерфейсів, використання яких дозволяє розв'язати дві основні задачі. По-перше, DirectX перетворює розроблені з його допомогою додатки в сумісні з будь-якою версією Windows і працюючі на будь-якому комп'ютері, де встановлена ця операційна система, незалежно від типу використовуваного апаратного забезпечення. При цьому подібні додатки максимально використовують можливості комп'ютера, забезпечуючи найкращу продуктивність. Це

досягається за рахунок сервісів, надаваних низкорівневими інтерфейсами DirectX Foundation.

По-друге, DirectX надає розроблювачам можливість абстрагуватися від того або іншого типу дисплейного адаптера, звукової карти або 3D-прискорювача й зосередитися на логіці роботи самої програми.

3.2 Декомпозиція системи

Інформаційна система дипломної роботи розробляється з метою максимально спростити процес дослідження відносного руху точки. За допомогою побудованої системи вирішуються дві основні задачі: візуалізація руху кульки у циліндричному каналі, що здійснює обертальний рух відносно та знаходження параметрів гвинтової пружини, до якої прикріплюється кулька.

Для реалізації розв'язку першої задачі повинні бути виділені наступні складові: інформаційна технологія чисельного розрахунку системи диференціальних рівнянь руху системи тіл, та інформаційна технологія моделювання руху системи тіл у тривимірному просторі.

Для реалізації розв'язку другої задачі повинні бути виділені наступні складові: інформаційна технологія чисельного розрахунку системи диференціальних рівнянь руху системи тіл, та інформаційна технологія знаходження параметрів гвинтової пружини.

На рис. 3.1 представлено схему декомпозиції інформаційної системи кваліфікаційної роботи.

Інформаційна технологія чисельного розрахунку візуалізації руху тіла відносно центру мас включає в себе застосування методу Рунге-Кутта (1.36) для чисельного розрахунку системи диференціальних рівнянь (2.9)-(2.10), (2.12). У якості вхідних даних задаються:

- параметри силових факторів відносного руху - коефіцієнти сил пружності, тертя, опору;

- параметри силових факторів переносного руху – коефіцієнти та закон моменту сил, що діє на пластину;
- величини геометрії системи – кут нахилу циліндричного каналу, розмір трубки;
- початкові дані руху – початкові координата та швидкість кульки в каналі, початкова швидкість обертання пластини з каналом.

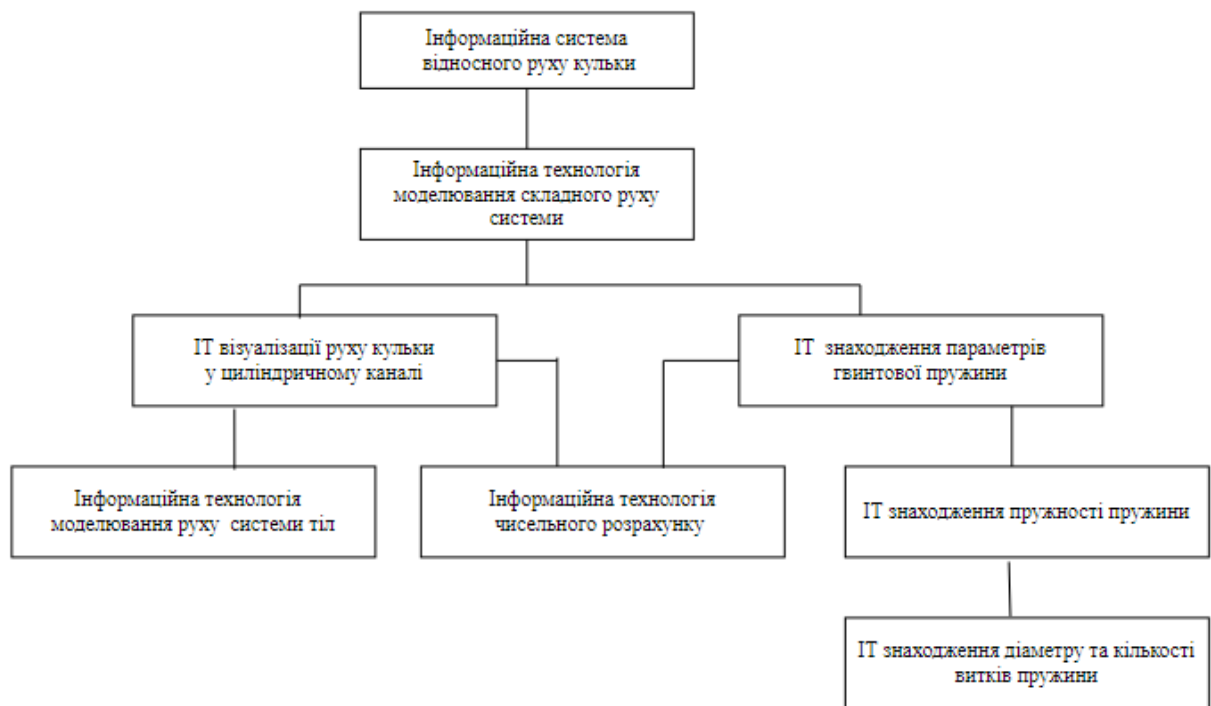


Рисунок 3.1- Декомпозиція ІС

Інформаційна технологія 3-D візуалізації руху системи тіл включає розробку тривимірної моделі тіла. Моделювання об'єктів для візуалізації виконується за допомогою пакету DirectX. У якості вхідних даних є значення кута повороту пластини та відносна координата кульки у циліндричному каналу у поточний момент часу. У якості вихідних даних є змодельована сцена тривимірної візуалізації.

Інформаційна технологія знаходження параметрів гвинтової пружини складається з двох етапів. Спочатку підбирається діапазон значень коефіцієнту пружності пружини згідно критерію (2.14). Для кожного

значення коефіцієнту пружності при заданих матеріалі пружини та діаметру проволочку пружини із знайденого інтервалу підбираються можлива кількість витків та діаметр пружини.

Для програмної реалізації системи розроблено три модулі проекту та два класи. На рис. 3.2 представлено схему структури системи. На рис. 3.3 представлено схему застосування інформаційної системи.

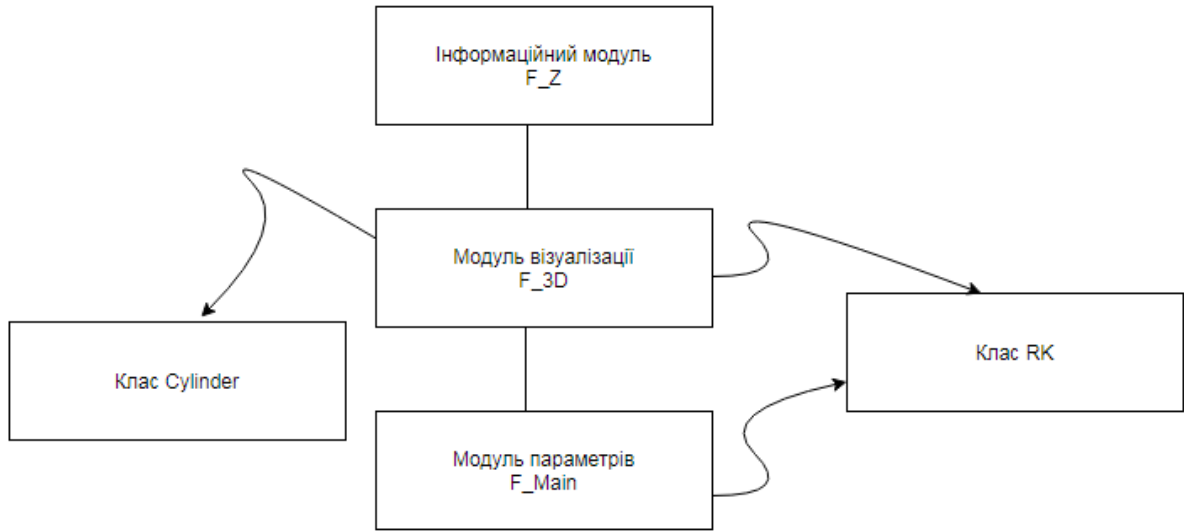


Рисунок 3.2 – Структура системи

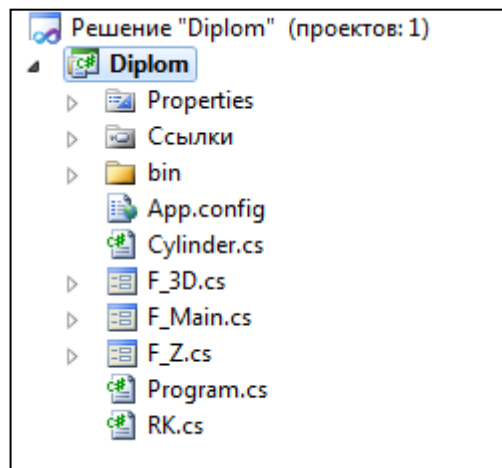


Рисунок 3.3 – Структура застосування

3.3 Реалізація класів

Розроблено клас RK, діаграму якого представлено на рис. 3.4.

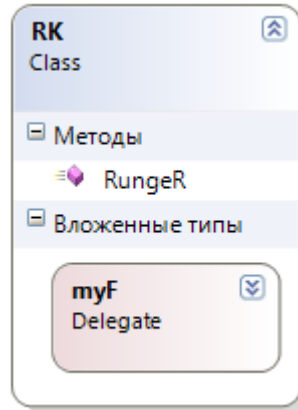


Рисунок 3.4 – Діаграма класу RK

Клас містить один метод та один делегат. Делегат забезпечує передачу вхідних параметрів до методу чисельного методу Рунге-Кутта RungeR.

Метод RungeR запрограмований для розрахунку ітераційної формули (1.36). Метод Runge приймає п'ять вхідних параметрів: початковий аргумент функції, початкове значення вектор – функції, масив делегатів правих частин системи диференціальних рівнянь, кількість диференціальних рівнянь системи та шаг обчислення. У якості вихідних даних є нові значення масиву значення вектор – функції системи диференціальних рівнянь.

Розроблено клас Cylinder, діаграму якого представлено на рис. 3.5. Клас містить дві властивості для геометрії об'єкту.

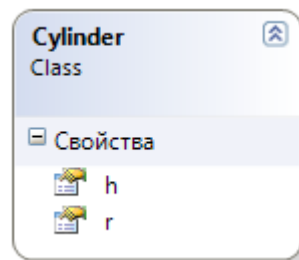


Рисунок 3.5 – Діаграма класу Cylinder

Клас використовується для моделювання 3-D об'єктів за допомогою пакету DirectX. **3D-Об'єкт** — це об'ємне тіло, що володіє довжиною, шириною й глибиною — властивостями, які протягом багатьох років у комп'ютерних зображень були відсутні. Насправді **3D-Об'єкти** існують тільки в пам'яті комп'ютера. Монітор лише відображає їх на плоскій поверхні екрана. Тривимірний об'єкт характеризується властивою йому формою й текстурою поверхні. Форма — це геометрія об'єкта, яка звичайно описується серією взаємозалежних точок (вершин) у тривимірному просторі й багатокутників (граней — замкнених двомірних фігур із трьома або більш сторонами). Наприклад, куб має вісім вершин і шість граней. Причому для простоти в більшості випадків як базових багатокутників вибираються трикутники й з'єднуються один з одним по кілька сотень або навіть тисяч, утворюючи складні тривимірні сітки. Текстура поверхні — це простий опис властивостей поверхні подібних багатокутників: їх кольору, прозорості, що відбиває здатності і так далі.

Усі фігури, що рисуються на сцені за допомогою Direct3D, являють собою сукупність точок – вершин. Інформація про кожну вершину містить такі параметри, як положення в просторі, колір вершини, інформацію про використовувану текстуру і т.д. Таким чином, вершина в Direct3D являє собою єдину структуру даних, названу FVF (Flexible Vertex Format – гнучкий формат вершини). У керованому DirectX є об'єкт, який може інкапсулювати вершини, — об'єкт Mesh [49]. Об'єкти Mesh можуть використовуватися для збереження будь-якого типу графічних даних, але головним чином призначені для формування складних моделей.

3.4 Інформаційний модуль

Інформаційний модуль додатка застосування (клас F_Z) надає користувачеві інформацію про призначення застосування. На етапі

візуального програмування головна форма має вигляд, що представлено на рис.3.6.

На формі розташовано не візуальний компонент таймер, для якого робиться обробка події Tick, щоб зробити функцію через деякий час невидимою та відкрити форму другого модуль (див. Додаток А).



Рисунок 3.6 – Зовнішній вигляд форми інформаційного вигляду

3.5 Модуль візуалізації руху системи тіл

Модуль візуалізації - клас F_3D, призначений для тривимірного моделювання руху системи. Для моделювання відносного руху кульки в циліндричному каналі задаються значення всіх параметрів моделі за замовченням.

Вісь обертання, та пластина моделюються за допомогою Mesh об'єктів – циліндрів, які пов'язанні з екземплярами класу Cylinder. Всі циліндри, крім

циліндричного каналу відносного руху мають суцільний вигляд. Циліндр каналу має каркасний вигляд, що дозволяє моделювати рух кульки відносно каналу. Для моделювання кульки використовується Mesh - об'єкт сфера, яка кріпиться до пружина. Сама пружина моделюється за допомогою невеликих сфер, координат яких на сцені знаходяться згідно розрахунку гвинтової лінії (рис.3.7).

Для розрахунку положення кульки у відносному русі використовується об'єкт класу RK. Згідно отриманого чисельного результату виконується візуалізація положення кульки у циліндричному каналі в функції класу F_3D Picture() (див. Додаток Б).

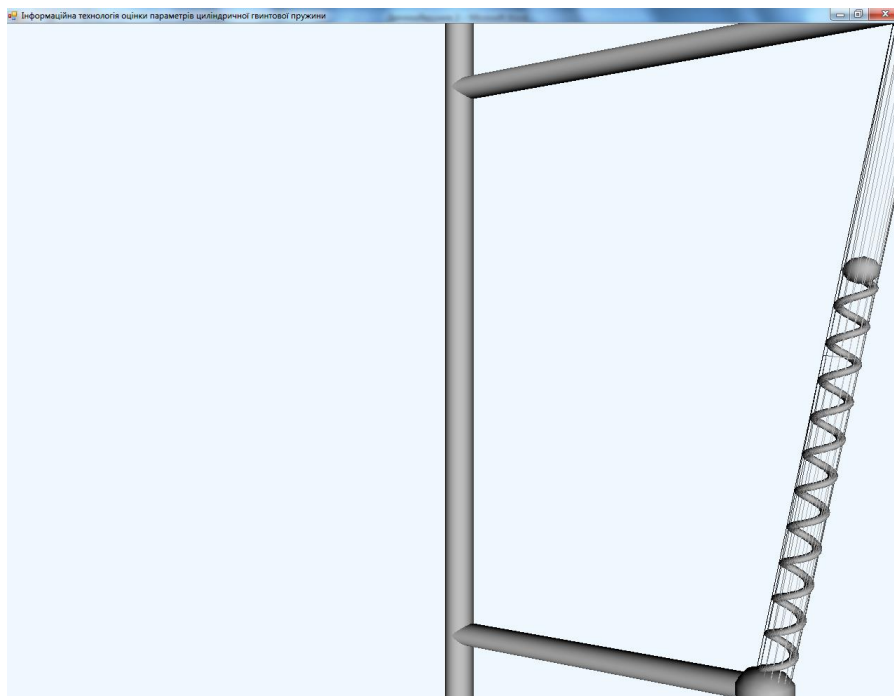


Рисунок 3.7 – Анімація руху системи тіл

Форма модуля має контекстне меню вигляду, що представлено на рис. 3.8. В меню користувачеві є доступним наступний функціонал:

- почати анімацію рух;
- призупинити анімацію руху;
- зупинити рух системи тіл;
- перейти до форми модуля параметрів моделі.

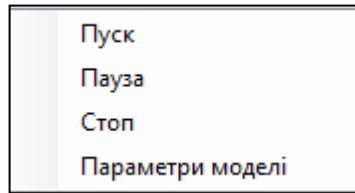


Рисунок 3.8 – Контекстне меню модуля візуалізації

Для кожного пункту меню викликається відповідна функція події (див. Додаток Б).

3.6 Модуль параметрів

Модуль застосування, клас F_Main, призначений для проведення аналізу рух кульки відносно циліндричного каналу та оцінки параметрів гвинтової пружини.

На етапі візуального програмування форма модуля має вигляд, що представлено на рис. 3.9.

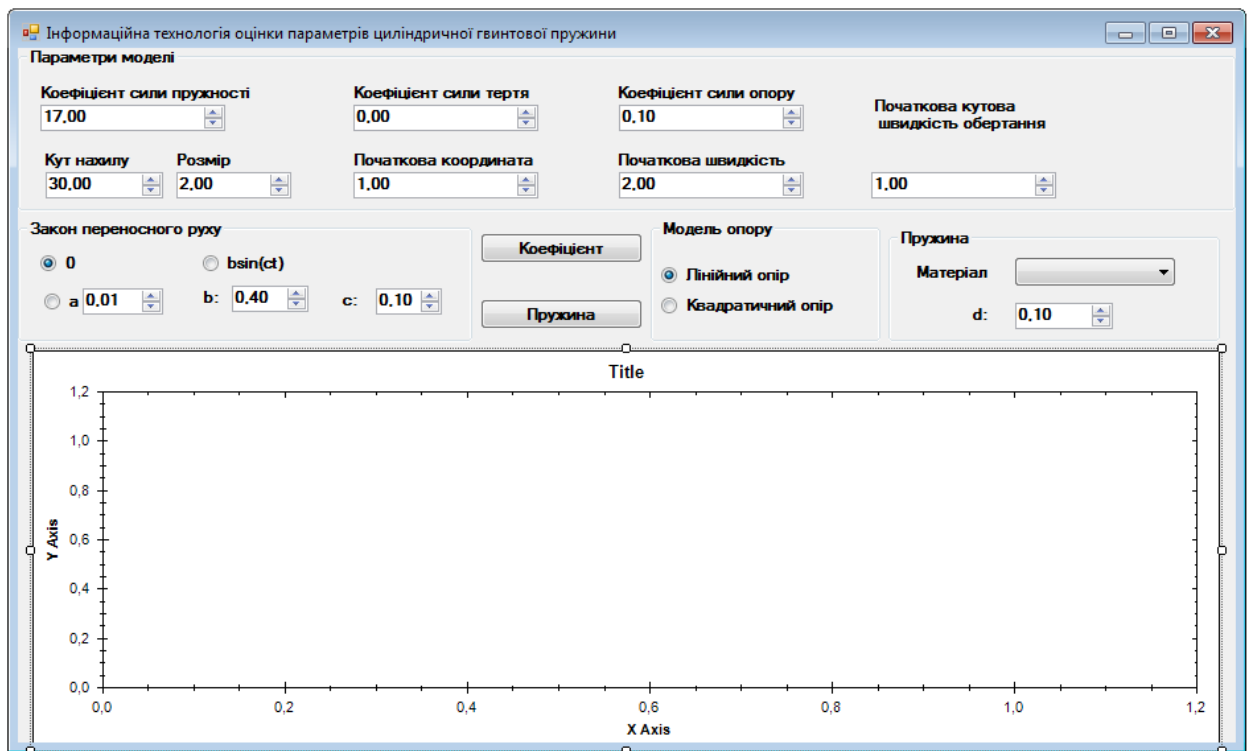


Рисунок 3.9 – Форма F_Main на етапі візуального програмування

На формі розташовано чотири контейнери для параметрів моделі:

1. Контейнер параметрів моделі.
2. Контейнер для розрахунку закону переносного руху системи.
3. Контейнер для вибору моделі опору середовища.
4. Контейнер для параметрів гвинтової пружини, що отримуються в процесі її виробництва.

В нижній частині вікна розташовано компонент побудови необхідних графіків для проведення аналізу відносного руху кульки та оцінки параметрів циліндричної гвинтової пружини.

Крім того, є дві функціональні кнопки для вирішення основних задач цього модуля:

- кнопка «Коефіцієнт» забезпечення знаходження діапазону параметру пружності пружини для виконання обмеження (2.14);
- кнопка «Пружина» відповідає функціональній реалізації оцінки параметрів пружини.

Реалізація функціоналу відбувається за рахунок виконання ряду чисельних експериментів для різних значень параметрів моделі (див. Додаток В).

4 ФУНКЦІОНАЛ СИСТЕМИ ДОСЛІДЖЕННЯ

4.1 Дослідження відносного руху кульки

Всі параметри моделі мають значення за замовченням, тому при відкриті форми модуля параметрів будується графік зміни відносної координати кульки у циліндричному каналі, як показано на рис. 4.1.

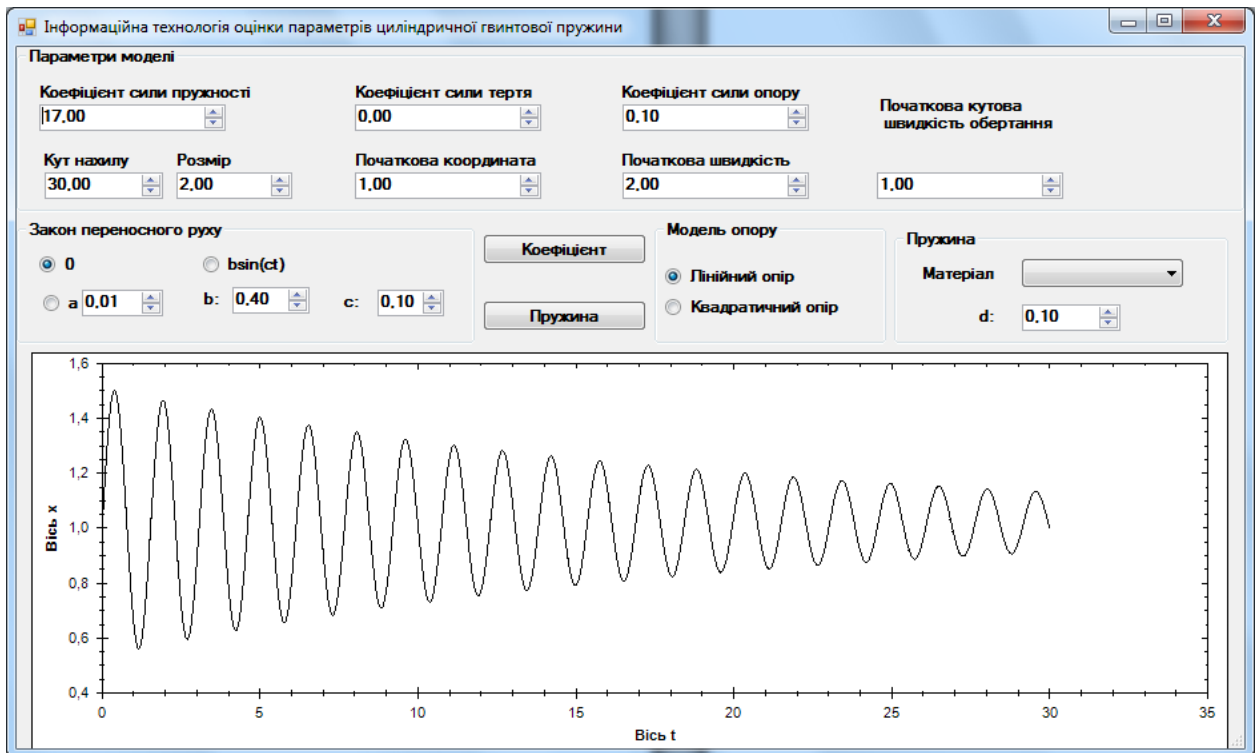


Рисунок 4.1 - Графік зміни відносної координати для значень параметрів за замовченням

Видно, що для заданих значень параметрів для руху кульки виконується обмеження (2.14). Розрахунок виконувався для лінійної моделі опору середовища, а пластина рухається з постійною кутовою швидкістю. Обираємо квадратичну модель опору середовища для тих самих значень параметрів моделі. Результат представлено на рис. 4.2. Видно, що істотної зміни характеру руху кульки не відбулося для заданих параметрів моделі.

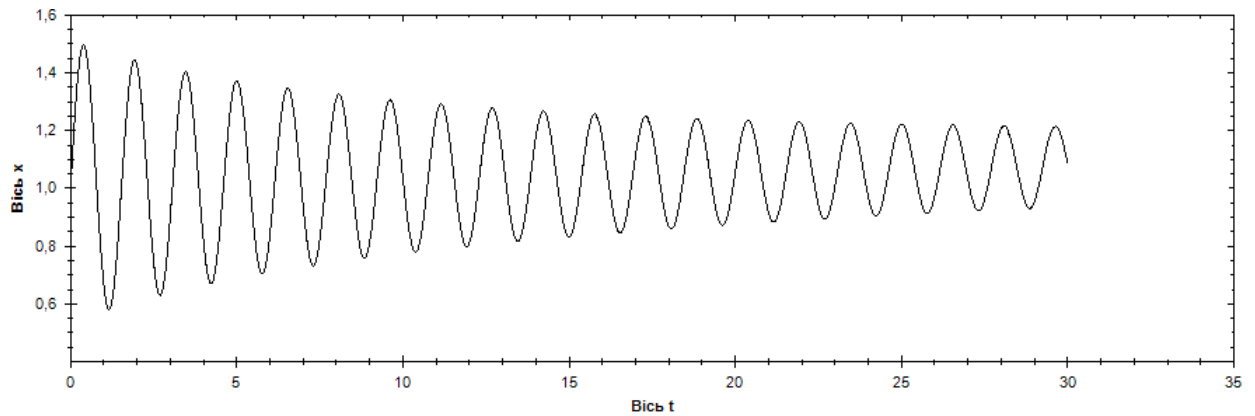


Рисунок 4.2 – Модель квадратичного опору середовища

Якщо повернутися до лінійної моделі опору та задати рух пластини під дією постійного моменту сил з коефіцієнтом $a=0.1$. Результат представлено на рис. 4.3. Характер коливання істотно змінюється, але залишається в обмеженнях координат кульки (2.14).

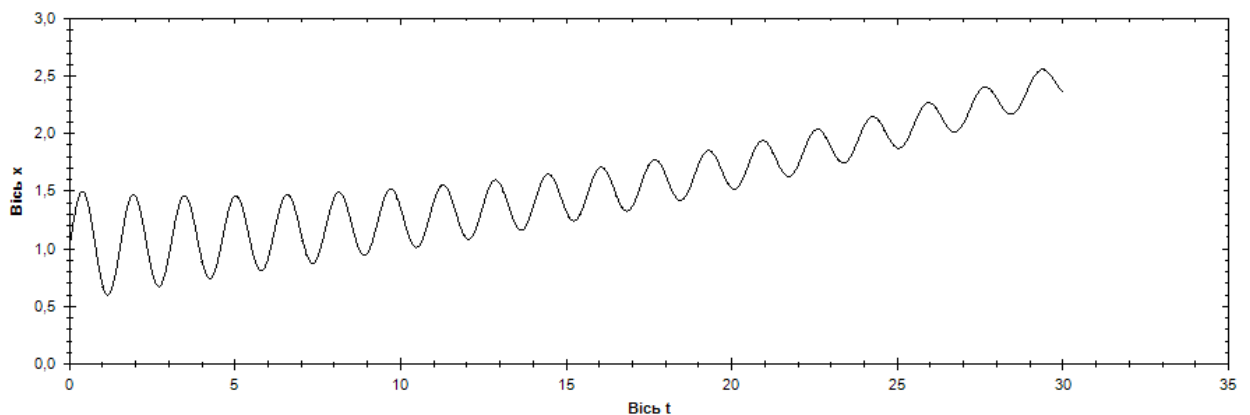


Рисунок 4.3 – Рух кульки у циліндричному каналі з постійним прискоренням

Цікавим є результат чисельного дослідження відносної координати кульки під дією моменту сил $M_z = b \sin(ct)$ для параметрів $b=0.32$ та $c=4$, що представлено на рис. 4.4. Видно, що коливання відбуваються з непостійною амплітудою, яка періодично змінюється.

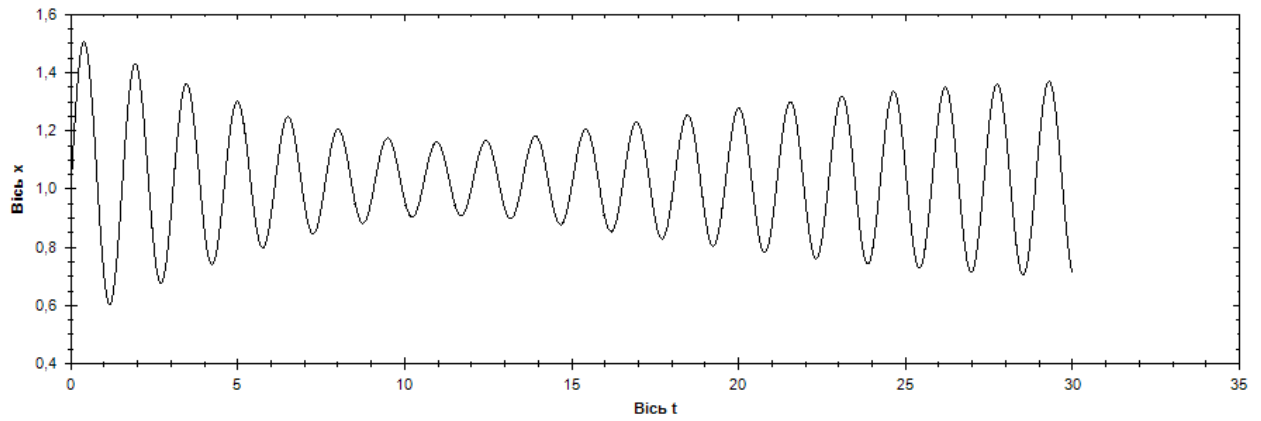


Рисунок 4.4 – Рух кульки у циліндричному каналі зі змінним прискоренням

Але не для всіх значень параметрів моделі виконується рух кульки в діапазоні обмеження (1.24). Наприклад, для пружності пружини $c=5$ маємо графік результатів, що представлено на рис. 4.5.

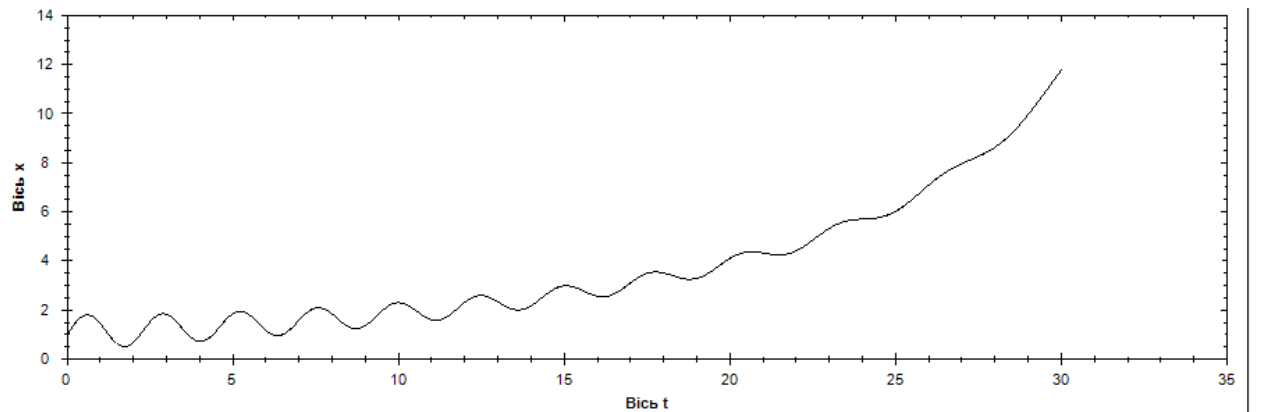


Рисунок 4.5 – Невиконання обмеження (1.24)

4.2 Оцінка параметра коефіцієнта пружності

Знайдення діапазону зміни коефіцієнту сили пружності виконується у функції події кнопки «Коефіцієнт» (див. Додаток В). Чисельний експеримент проводиться для значень від 0.1 до 30.

Діапазон параметра χ_1 у випадку, що відповідає розрахунку рис. 4.1 дає результат $\chi_1 \in [0.4; 30]$. Якщо провести оцінку коефіцієнта пружності для розрахунку рис. 2, то маємо той самий результат.

Для обертання пластини з постійним прискорення $a=0.1$ не має значень параметру χ_1 . Зменшення значення a до величини $a=0.01$ дає діапазон $\chi_1 \in [0.4; 30]$ від 1.2 до 1.3.

У випадку обертання пластини під дією моменті сил $M_z = b \sin(ct)$ для $b=0.32$ та $c=4$ маємо результат $\chi_1 \in [4.6; 23.4]$. Збільшимо b до 0.5 для того же значення c , тоді $\chi_1 \in [4.8; 12.7]$. Подальше збільшення b до величини 0.6 приводить до ще більшого зменшення діапазону параметру $\chi_1 \in [4.8; 6.2]$. Це дозволяє нам зробити висновок, що збільшення параметру b приводить до зменшення діапазону допустимих значень коефіцієнта пружності.

У випадку обертання пластини під дією моменті сил $M_z = b \sin(ct)$ для $b=0.32$ та $c=2$ не має значень параметру χ_1 . Для $c=3$ маємо результат $\chi_1 \in [2.3; 2.4]$. При збільшенні c до величини 5 приводить до збільшення діапазону параметру $\chi_1 \in [1.6; 30]$. Для $c=6$ діапазон зменшується $\chi_1 \in [2.6; 2.7]$.

Після знаходження діапазону параметра χ_1 для відповідного вікна мінімальне та максимальне значення становляться рівними границям діапазону.

4.3 Оцінка параметрів циліндричної гвинтової пружини

Для кожного значення параметра χ_1 із розрахованого діапазону може бути проведено оцінка параметрів гвинтової циліндричної пружини згідно (2.13).

Користувач може обирати матеріал пружини та діаметр проволочи за допомогою функціонал, як показано на рис. 4.6.

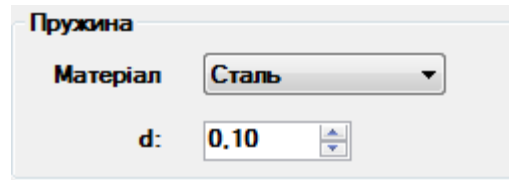


Рисунок 4.6 – Панель статичних параметрів пружини

Після натискання кнопки «Пружина» будується графік залежності кількості витків пружини від діаметру пружини, як показано на рис. 4.7.

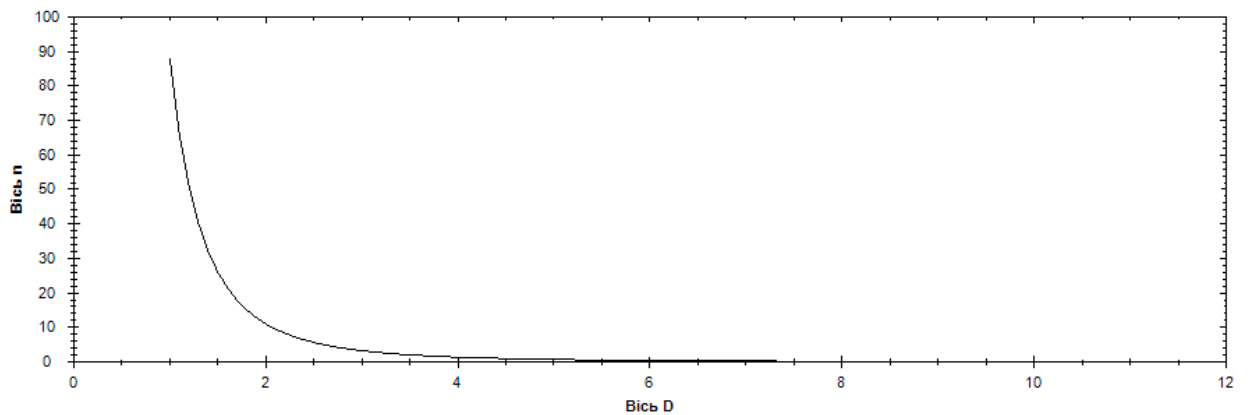


Рисунок 4.7 – Залежність $n=n(D)$ для $\chi_1 = 14.2$

Збільшимо значення параметру χ_1 до величини 17, та зробимо оцінку параметрів пружини. Маємо, що для того ж само діаметру необхідно зробити меншу кількість витків, як видно з рис. 4.8.

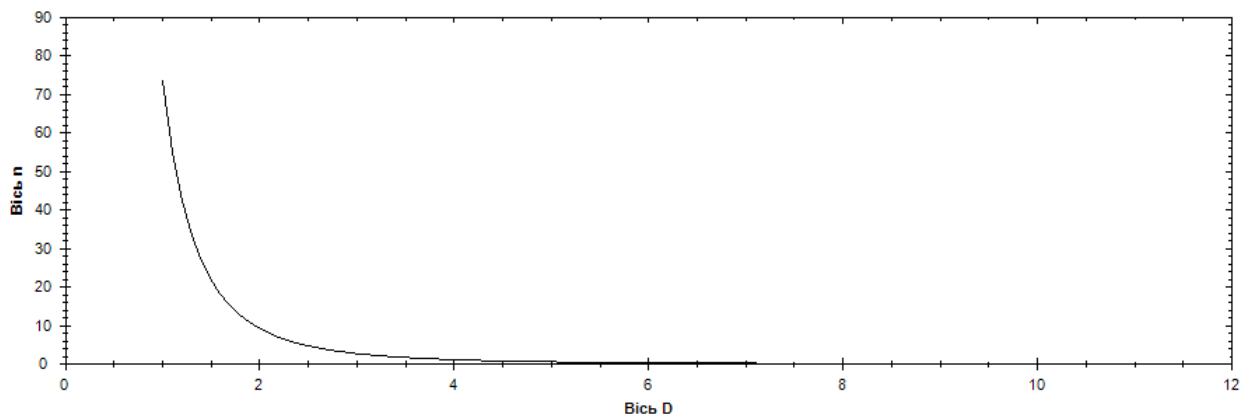


Рисунок 4.8 – Залежність $n=n(D)$ для $\chi_1 = 17$

Зменшимо значення параметру χ_1 до величини 10, та зробимо оцінку параметрів пружини. Маємо, що для того ж само діаметру необхідно зробити більшу кількість витків, як видно з рис. 4.9.

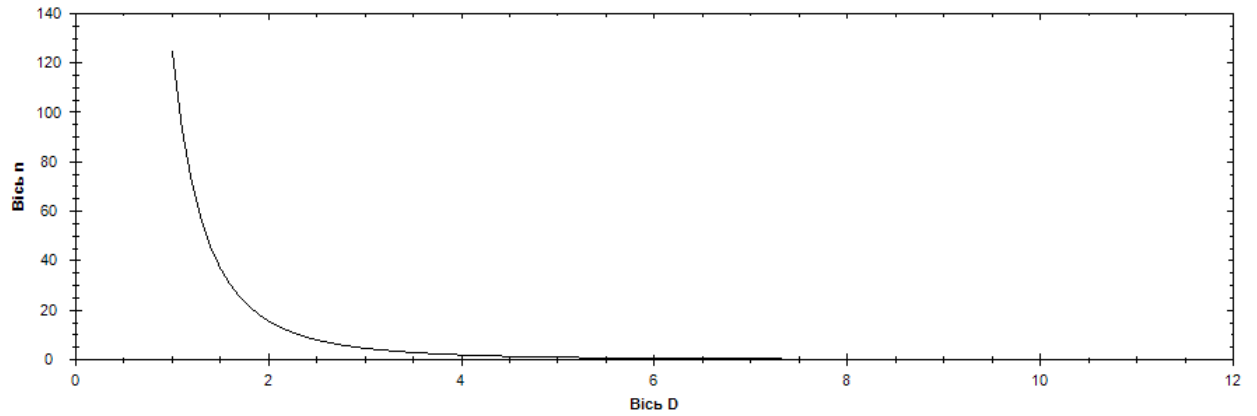


Рисунок 4.9 – Залежність $n=n(D)$ для $\chi_1 = 10$

ВИСНОВОК

В роботі розроблено інформаційну систему для моделювання відносного руху кульки в циліндричному каналі для заданого переносного руху системи.

Для розробки системи дослідження розв'язано наступні задачі:

1. Проаналізовано існуючі моделі складного руху системи тіл.
2. Створено математичну модель відносного руху кульки у циліндричному каналі.
3. Змодельовано віртуальну лабораторію, в якій реалізується тривимірне моделювання складного руху системи тіл.
4. Розроблено алгоритм чисельного розрахунку зазначеної моделі для визначення діапазону значень безрозмірного коефіцієнту сили пружності.
5. Проведено оцінку діаметру пружини та кількості витків пружини для знайденого коефіцієнту пружності.

Результатом виконання дипломної роботи є можливість оцінити параметри пружини для розробленої моделі складного руху системи тіл.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Павловський М.А., Теоретична Механіка // Київ: «ТЕХНІКА», 2002 – 512 с.
2. Апостолюк О. С., Теоретична механіка: Збірник задач / О. С. Апостолюк, В. М. Воробйов, Д. І. Ільчишина та ін.// Київ: «Техніка», 2007 – 400 с.
3. Березін Л.М., Теоретична механіка / Березін Л.М., Кошель С.О.// Київ: «Центр учбової літератури», 2020 – 218 с.
4. Кузьо І.В., Теоретична механіка // Київ: «Фоліо», 2017 – 780 с.
5. Булгаков В.М., Теоретична механіка / В.М. Булгаков, В.В. Яременко, О.М. Черниш, М.Г.Березовий// Київ: «Центр учбової літератури», 2021 – 640 с.
6. Яскілка М.Б., Збірник завдань для розрахунково-графічних робіт з теоретичної механіки/ Київ: Вища школа: «Веселка», 1999. – 351 с.
7. Бублик В.В. Об'єктно-орієнтоване програмування// Київ: Підручник: «ІТ-книга», 2015 – 624 с.
8. Кравець П.О., Об'єктно-орієнтоване програмування// Львів: Видавництво Львівської політехніки, 2012 – 624 с.
9. Дібрівний О.А., Вступ до об'єктно-орієнтованого програмування C#/ Дібрівний О.А., Гребенюк В.В.// Київ: Державний університет телекомунікацій, 2018. – 190 с.
10. Коноваленко І.В., Програмування мовою C# 6.0// Тернопіль: ТНТУ, 2016. – 227 с.

ДОДАТОК А

Код інформативного модуля

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Diplom
{
    public partial class F_Z : Form
    {
        public F_Z()
        {
            InitializeComponent();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            Hide();
            F_3D f = new F_3D();
            f.Show();
            timer1.Enabled = false;
        }
    }
}
```

ДОДАТОК Б

Код модуля візуалізації

```

public partial class F_3D : Form
{
    Device d3d;
    Mesh M_os, M_pr1, M_pr2, M_canal, M_soed;
    Cylinder C_os, C_pr1, C_pr2, C_canal;
    Material metal;
    float X=0.01f, Y=1f, Z = 3f;
    double t = 0;
    double eps = 0.01;
    RK raset;
    RK.myF[] masF;
    double[] y;
    //пружина
    Mesh sf;
    int n = 23;
    float hsp = 0.1f;
    float asp = 0.2f;
    bool nach = true;
    Mesh sharik;
    F_Main fm;
    public F_3D()
    {
        InitializeComponent();
        raset = new RK();
        masF = new RK.myF[3];
        masF[0] = f1;
        masF[1] = f2;
        masF[2] = f3;
        y = new double[3];
        y[0] = 1f;
        y[1] = 0.1f;
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        try
        {
            PresentParameters d3dpp = new PresentParameters();
            d3dpp.BackBufferCount = 1;
            d3dpp.SwapEffect = SwapEffect.Discard;
            d3dpp.Windowed = true;
            d3dpp.MultiSample = MultiSampleType.None;
            d3dpp.EnableAutoDepthStencil = true;
            d3dpp.AutoDepthStencilFormat = DepthFormat.D16;
            d3d = new Device(0, DeviceType.Hardware, this,
                CreateFlags.SoftwareVertexProcessing, d3dpp);
        }
        catch (Exception)
        {
            Close();
        }
        C_os=new Cylinder();
        C_os.r=0.04f;
        C_os.h=5f;
        C_pr1 = new Cylinder();
        C_pr1.r = 0.04f;
        C_pr1.h = 1.2f;
        C_pr2 = new Cylinder();
        C_pr2.r= 0.04f;
        C_pr2.h = 0.87f;
        C_canal = new Cylinder();
        C_canal.r = 0.04f;
        C_canal.h = 2.0f;
        M_os = Mesh.Cylinder(d3d, C_os.r, C_os.r, C_os.h, 25, 25);
        M_pr1 = Mesh.Cylinder(d3d, C_pr1.r, C_pr1.r, C_pr1.h, 25, 25);
        M_pr2 = Mesh.Cylinder(d3d, C_pr2.r, C_pr2.r, C_pr2.h, 25, 25);
        M_canal = Mesh.Cylinder(d3d, C_canal.r, C_canal.r, C_canal.h, 15, 2);
        M_soed = Mesh.Sphere(d3d, 0.07f, 10, 10);
        metal = new Material();
        metal.Diffuse = Color.Silver;
        metal.Specular = Color.White;
    }
}

```

```

//пружина
sf = Mesh.Sphere(d3d, 0.01f, 10, 10);
sharik = Mesh.Sphere(d3d, C_canal.r, 10, 10);
fm = new F_Main();
fm.MyScroll();
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Picture();
}
void Picture()
{
    d3d.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.AliceBlue, 1.0f, 0);
    SetupProekcii();
    d3d.BeginScene();
    d3d.Material = metal;
    d3d.RenderState.FillMode = FillMode.Solid;
    d3d.Transform.World = Matrix.RotationX((float)Math.PI/2) *
Matrix.RotationY((float)(y[2]*t))*Matrix.Translation(X, 0, Z);
    M_os.DrawSubset(0);
    d3d.Transform.World = Matrix.Translation(0, 0, C_pr1.h / 2) *
Matrix.RotationY((float)Math.PI / 2) * Matrix.RotationY((float)(y[2] * t)) *
Matrix.Translation(X, Y, Z);
    M_pr1.DrawSubset(0);
    d3d.Transform.World = Matrix.Translation(0, 0, C_pr2.h / 2) *
Matrix.RotationY((float)Math.PI / 2) * Matrix.RotationY((float)(y[2] * t)) *
Matrix.Translation(X, -Y, Z);
    M_pr2.DrawSubset(0);
    d3d.Transform.World = Matrix.Translation(C_pr1.h, Y, 0) *
Matrix.RotationY((float)(y[2] * t)) * Matrix.Translation(X, 0, Z);
    M_soed.DrawSubset(0);
    d3d.Transform.World = Matrix.Translation(C_pr2.h, -Y, 0) *
Matrix.RotationY((float)(y[2] * t)) * Matrix.Translation(X, 0, Z);
    M_soed.DrawSubset(0);
    d3d.RenderState.FillMode = FillMode.WireFrame;
    d3d.Transform.World = Matrix.Translation(0, 0, C_canal.h / 2) *
Matrix.RotationX((float)Math.PI / 2) * Matrix.RotationZ(-(float)Math.PI / 20) *
Matrix.Translation(C_pr1.h, Y, 0) * Matrix.RotationY((float)(y[2] * t)) * Matrix.Translation(X,
0, Z);
    M_canal.DrawSubset(0);
    //пружина
    float koeff = 1;
    float x_pr = 0, y_pr = 0, z_pr = 0;
    asp = C_canal.r*0.8f;
    hsp = Convert.ToSingle(y[0] / n/koeff);
    for (float fi = 0; fi <= n * Math.PI; fi += 0.1f)
    {
        x_pr = Convert.ToSingle(asp * Math.Cos(fi));
        z_pr =Convert.ToSingle(asp * Math.Sin(fi));
        y_pr = Convert.ToSingle(hsp * fi / Math.PI);
        d3d.Transform.World = Matrix.Translation(x_pr, y_pr, z_pr) * Matrix.RotationZ(-
(float)Math.PI / 20) * Matrix.Translation(C_pr2.h + asp / 2, -Y, 0) *
Matrix.RotationY((float)(y[2] * t)) * Matrix.Translation(X, 0, Z);
        sf.DrawSubset(0);
    }
    d3d.RenderState.FillMode = FillMode.Solid;
    d3d.Transform.World = Matrix.Translation(0, Convert.ToSingle(y[0]) / koeff, 0) *
Matrix.RotationZ(-(float)Math.PI / 20) * Matrix.Translation(C_pr2.h + asp / 2, -Y, 0) *
Matrix.RotationY((float)(y[2] * t)) * Matrix.Translation(X, 0, Z);
    sharik.DrawSubset(0);
    d3d.EndScene();
    d3d.Present();
}
private void SetupProekcii()
{
    d3d.Lights[0].Enabled = true;
    d3d.Lights[0].Diffuse = Color.White;
    d3d.Lights[0].Position = new Vector3(0, 0, 0);
    d3d.Transform.Projection = Matrix.PerspectiveFovLH((float)Math.PI / 4, this.Width /
this.Height, 1.0f, 50.0f);
}
private void pictureBox1_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
}
double f1(double x, double[] y, int n)
{
    return y[1];
}

```

```

}

double f2(double x, double[] y, int n)
{
    return fm.f2(x,y,n);
}

double f3(double x, double[] y, int n)
{
    return fm.ff(x, y, n);
}

private void timer1_Tick(object sender, EventArgs e)
{
    t = t + 0.01;
    y = rashet.RungeR(t, y, masF, 3, eps);
    // y[0] = 1 + Convert.ToSingle(0.5f*Math.Sin(t));
    Picture();
}

private void ныскToolStripMenuItem_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
    rashet = new RK();
    masF = new RK.myF[3];
    masF[0] = fm.f1;
    masF[1] = fm.f2;
    masF[2] = fm.ff;
    y = new double[3];
    y[0] = fm.x0;
    y[1] = fm.xs0;
    y[2] = fm.omega0;
}

private void cronToolStripMenuItem_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
    t = 0;
    y[0] = 0.5f;
    y[1] = 0.1f;
}

private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
}

private void F_3D_FormClosed(object sender, FormClosedEventArgs e)
{
    Application.Exit();
}

private void параметриМоделиToolStripMenuItem_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
    fm.ShowDialog();
}
}

```

ДОДАТОК В

Код модуля параметрів

```

namespace Diplom
{
    public partial class F_Main : Form
    {
        //delegate double delf(double x);
        GraphPane myPane = null;
        RK rashet;
        double eps = 0.001;
        RK.myF[] masF;
        double[] y;
        public double alfa, r, x0, xs0, xi1, xi2, xi3, ftr, t_k=30, omega0;
        public double xi1_min {get;set;}
        public double xi1_max { get; set; }
        double G = 1e8;
        double d = 0.1;
        double A, B, C;
        public RK.myF ff;
        static public List<double> N1;
        static public List<double> N2;
        static public List<double> N;
        static public List<double> T;
        static public List<double> xras;
        static public List<double> tras;
        public F_Main()
        {
            InitializeComponent();
            ff = epsilon1;
            Visible = false;
            rashet = new RK();
            masF = new RK.myF[3];
            masF[0] = f1;
            masF[1] = f2;
            masF[2] = ff;
            y = new double[3];
            N1 = new List<double>();
            N2 = new List<double>();
            N = new List<double>();
            T = new List<double>();
            xras = new List<double>();
            tras = new List<double>();
        }
        private void Form6_Load(object sender, EventArgs e)
        {
            myPane = zedGraphControll1.GraphPane;
            xi1 = Convert.ToDouble(hSX1.Value);
            xi2 = Convert.ToDouble(hSXi2.Value);
            xi3 = xi2;
            alfa = Convert.ToDouble(hSalfa.Value) * Math.PI / 180;
            r = Convert.ToDouble(hSr.Value);
            x0 = Convert.ToDouble(hSx0.Value);
            xs0 = Convert.ToDouble(hSomega0.Value);
            A = Convert.ToDouble(NUAA.Value);
            B = Convert.ToDouble(NUBB.Value);
            C = Convert.ToDouble(NUCC.Value);
            ftr = Convert.ToDouble(NUftr.Value);
            omega0 = Convert.ToDouble(NUomega0.Value);
            y[0] = x0;
            y[1] = xs0;
            y[2] = omega0;
            ff = epsilon1;
        }
        public double f1(double x, double[] y, int n)
        {
            return y[1];
        }
        public double f2(double x, double[] y, int n)
        {
            double N1 = (2 * y[2] * y[1]+ff(x, y, n)*y[0]) * Math.Sin(alfa)+ff(x, y, n)*r;
            double N2 = y[2] * y[2] * (r + y[0] * Math.Sin(alfa)) * Math.Cos(alfa) +
                Math.Sin(alfa);
            double rez = ftr * Math.Sqrt(N1 * N1 + N2 * N2);
            if(Mod1.Checked)
                return y[2] * y[2] * (r + y[0] * Math.Sin(alfa)) * Math.Sin(alfa) -
                    Math.Cos(alfa) - xi1 * (y[0] - 1) - xi2 * y[1] - Math.Sign(y[1]) * rez;
        }
    }
}

```

```

else
    return y[2]* y[2] * (r + y[0] * Math.Sin(alfa)) * Math.Sin(alfa) - Math.Cos(alfa)
- xi1 * (y[0] - 1) - Math.Sign(y[1]) * xi2 * y[1] * y[1] - Math.Sign(y[1]) * rez;
}
public double epsilon1(double t, double[] y, int n)
{ return 0; }
public double epsilon2(double t, double[] y, int n)
{ return A; }
public double epsilon3(double t, double[] y, int n)
{ return B * Math.Sin(C * t);}
public void MyScroll()
{
    xi1 = Convert.ToDouble(hSX1.Value);
    xi2 = Convert.ToDouble(hSX2.Value);
    xi3 = xi2;
    alfa = Convert.ToDouble(hSalfa.Value) * Math.PI / 180;
    r = Convert.ToDouble(hSr.Value);
    x0 = Convert.ToDouble(hSx0.Value);
    xs0 = Convert.ToDouble(hSomega0.Value);
    ftr = Convert.ToDouble(NUftr.Value);
    omega0 = Convert.ToDouble(NUomega0.Value);
    y[0] = x0;
    y[1] = xs0;
    y[2] = omega0;
    A = Convert.ToDouble(NUAA.Value);
    B = Convert.ToDouble(NUBB.Value);
    C = Convert.ToDouble(NUCC.Value);
}
void Grafic()
{
    N1.Clear();
    N2.Clear();
    N.Clear();
    T.Clear();
    myPane.CurveList.Clear();
    // Задаем название графика и осей
    myPane.Title.Text = "";
    myPane.XAxis.Title.Text = "Вісь t";
    myPane.YAxis.Title.Text = "Вісь x";
    // Создадим список точек
    PointPairList list = new PointPairList();
    for (double i = 0; i < t_k; i += eps)
    {
        double NN1 = (Math.Sin(alfa) * (2 * y[2] * y[1] + ff(i,y,3) * y[0]) + ff(i,y,3) *
            r);
        double NN2 = y[2] * y[2] * (r + y[0] * Math.Sin(alfa)) * Math.Cos(alfa) +
            Math.Sin(alfa);
        double rez = Math.Sqrt(NN1 * NN1 + NN2 * NN2);
        N1.Add(NN1);
        N2.Add(NN2);
        N.Add(rez);
        T.Add(i);
        y = rashet.Runger(i, y, masF, 3, eps);
        list.Add(i, y[0]);
    }
    LineItem myCurve;
    myCurve = myPane.AddCurve("", list, Color.Black);
    myCurve.Symbol.IsVisible = false;
    zedGraphControl1.AxisChange();
    zedGraphControl1.Invalidate();
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    MyScroll();
    Grafic();
}
private void radioButton1_Click(object sender, EventArgs e)
{
    masF[2]=epsilon1;
    ff = epsilon1;
    MyScroll();
    Grafic();
}
private void radioButton2_Click(object sender, EventArgs e)
{
    masF[2] = epsilon2;
    ff = epsilon2;
    MyScroll();
    Grafic();
}

```

```

}
private void radioButton4_Click(object sender, EventArgs e)
{
    masF[2] = epsilon3;
    ff = epsilon3;
    MyScroll();
    Grafic();
}
private void Mod1_Click(object sender, EventArgs e)
{
    MyScroll();
    Grafic();
}
private void hSX1_ValueChanged(object sender, EventArgs e)
{
    MyScroll();
    Grafic();
}
private void button2_Click(object sender, EventArgs e)
{
    double p_xil = xil;
    xil_min = -1;
    xil_max = 30;
    double i;
    bool flag1 = true;
    bool flag2 = true;
    bool flag3 = true;
    xil = -0.1;
    while (flag2)
    {
        xil += 0.1;
        xi2 = Convert.ToDouble(hSXi2.Value);
        xi3 = xi2;
        alfa = Convert.ToDouble(hSalfa.Value) * Math.PI / 180;
        r = Convert.ToDouble(hSr.Value);
        x0 = Convert.ToDouble(hSx0.Value);
        xs0 = Convert.ToDouble(hSomega0.Value);
        A = Convert.ToDouble(NUAA.Value);
        B = Convert.ToDouble(NUBB.Value);
        C = Convert.ToDouble(NUCC.Value);
        ftr = Convert.ToDouble(NUftr.Value);
        y[0] = x0;
        y[1] = xs0;
        xras.Clear();
        tras.Clear();
        for (i = 0; i < t_k; i += eps)
        {
            y = rashet.RungeR(i, y, masF, 3, eps);
            xras.Add(y[0]);
            tras.Add(i);
            if ((y[0] < 0.5 || y[0] > 10) && flag1 && flag3) { break; }
            if ((y[0] < 0.5 || y[0] > 10) && !flag1 && flag3) { flag3 = false; break; }
        }
        if (i >= 10 && flag1) { xil_min = xil; flag1 = false; }
        if (i >= 10 && !flag3) { xil_max = xil - 0.1; flag2 = false; }
        if (xil > 30) break;
    }
    if (xil_min != xil_max)
        MessageBox.Show(String.Format("min xil:{0:F3}, max xil:{1:F3}", xil_min,
xil_max));
    else
        MessageBox.Show(":(");
    xil = p_xil;
    hSX1.Minimum = Convert.ToDecimal(xil_min);
    hSX1.Maximum = Convert.ToDecimal(xil_max);
    MyScroll();
    Grafic();
}
private void F_Main_Shown(object sender, EventArgs e)
{
    MyScroll();
    Grafic();
}
private void button1_Click(object sender, EventArgs e)
{
    myPane.CurveList.Clear();
}

```

```
// Задаем название графика и осей
myPane.Title.Text = "";
myPane.XAxis.Title.Text = "Вись D";
myPane.YAxis.Title.Text = "Вись n";

// Создадим список точек
PointPairList list = new PointPairList();
for (double D = 1; D < 10; D +=0.1 )
    list.Add(D, G*Math.Pow(d,4)/(8*xil*Math.Pow(D,3)));

LineItem myCurve;
myCurve = myPane.AddCurve("", list, Color.Black);
myCurve.Symbol.IsVisible = false;
zedGraphControl1.AxisChange();
zedGraphControl1.Invalidate();
    }
}
```