

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

О.В.Узун

ОСНОВИ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ФІЗИЧНО РІЗНОРІДНИХ СИСТЕМ ТА ЇХ ЕЛЕМЕНТІВ

Лабораторний практикум

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за освітньою програмою «автоматизовані та роботизовані механічні системи»
спеціальностями 131 «Прикладна механіка», 133 «галузеве машинобудування»

Електронне мережеве навчальне видання

Київ
“КПІ ім. І. Сікорського”
2024

УДК 681.52:[004.896+004.94]

УЗ4

Автор: *Узунов Олександр Васильович*, д-р техн. наук, проф.

Рецензент: Назаренко І.І., д-р техн. наук, проф.
Київський національний університет будівництва і архітектури

Відповідальний редактор: Коваль О.Д., к.т.н., доц.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 6 від 28.03.2024 р.)
за поданням вченої ради навчально-наукового механіко машинобудівного інституту
(протокол № 8 від 25.03.2024 р.)*

Узунов О.В.

УЗ4 Основи математичного моделювання фізично різномірних систем та їх елементів [Електронний ресурс] : лаб. практикум : навч. посіб.для здобувачів ступеня бакалавра за освіт. програмою «автоматизовані та роботизовані механічні системи» спец. 131 «Прикладна механіка», 133 «галузеве машинобудування» ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2024. – 95 с.

Лабораторний практикум містить комплект лабораторних робіт, що рекомендовані в курсі «Основи математичного моделювання фізично різномірних систем», в якому приділено увагу питанням розробки комп'ютерних програм для моделювання статичних та динамічних характеристик, а також питанням дослідження фізичних елементів таких систем та їх співставлення з результатами моделювання. Практикум призначений для здобувачів ступеня бакалавра за спеціальностями 131 «Прикладна механіка», 133 «галузеве машинобудування» і буде також корисним для студентів машинобудівних спеціальностей вищих учбових закладів.

УДК 681.52:[004.896+004.94]

Реєстр. № НП 23/24-426. Обсяг 5.93 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Берестейський, 37, м. Київ, 03056
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

©О.В. Узунов, 2024
© КПІ ім. Ігоря Сікорського, 2024

Зміст

	ВСТУП	5
1.	ЗАГАЛЬНІ ВІДОМОСТІ	6
1.1.	Фізично різнорідні системи та задачі моделювання.....	6
1.2.	Приклади систем, проблеми та підхід до їх вирішення.....	8
1.3.	Загальний процес розроблення програм моделювання.....	10
2	ІНСТРУМЕНТИ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ	13
2.1.	Засоби пакету MatLab для обчислень та моделювання.....	13
2.1.1.	Засоби формування обчислювального процесу.....	13
2.1.1.1.	Засоби виконання обчислювальних операцій.....	13
2.1.1.2.	Засоби забезпечення багаторазового виконання операцій.....	14
2.1.1.3.	Засоби керування ходом обчислювального процесу.....	15
2.1.2.	Забезпечення взаємодії з користувачем.....	16
2.1.2.1.	Засоби отримання інформації від користувача.....	16
2.1.2.2.	Засоби представлення інформації для користувача.....	17
2.1.3.	Побудова програм моделювання процесів.....	22
2.1.3.1.	Формування алгоритму.....	22
2.1.3.2.	Перетворення алгоритму в програму.....	23
2.1.4.	Стандартні функції для вирішення спеціальних задач.....	25
2.1.5.	Чисельне рішення диференціальних рівнянь.....	25
3.	ЗАСОБИ І ОСОБЛИВОСТІ ФІЗИЧНОГО МОДЕЛЮВАННЯ	30
3.1.	Обладнання для фізичного моделювання.....	30
3.2.	Фізичні експерименти та математичні моделі.....	32
4.	ЗАВДАННЯ ДЛЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ	35
4.1.	Лабораторна робота №4.1. Візуалізація математичних функцій.....	35
4.2.	Лабораторна робота №4.2. Розгалужена будова моделі.....	36
4.3.	Лабораторна робота №4.3. Моделювання і дослідження поведінки математичної функції.....	38
4.4.	Лабораторна робота №4.4.Прогнозна модель дії об'єкту.....	40
4.5.	Лабораторна робота №4.5.Моделювання динамічного процесу руху тіла...	42
4.6.	Лабораторна робота №4.6. Моделювання процесів у пружній системі.....	43
4.7.	Лабораторна робота №4.7.Моделювання процесів в гідравлічних компонентах систем.....	45
5.	ЗАВДАННЯ ДЛЯ ФІЗИЧНОГО МОДЕЛЮВАННЯ	47
5.1.	Лабораторна робота №5.1. Дослідження та математичний опис статичних характеристик інформаційних елементів.....	47
5.2.	Лабораторна робота №5.2. Дослідження та математичний опис статичних характеристик підсилюючих елементів.....	48
5.3.	Лабораторна робота №5.3.Розробка, перевірка та математичне представлення модулів	49
5.4.	Лабораторна робота №5.4. Дослідження та математичний опис статичних характеристик обчислювальних елементів	50
5.5.	Лабораторна робота №5.5. Дослідження та математичний опис статичних характеристик виконавчих елементів.....	51
5.6.	Лабораторна робота №5.6. Розроблення, експериментальне дослідження	

	та математичне представлення модулю підсилення.....	52
5.7.	Лабораторна робота №5.7. Розроблення, дослідження та представлення алгоритму дії системи керування електричним приводом.....	54
5.8.	Лабораторна робота №5. Розроблення, дослідження та представлення алгоритму дії системи автоматичного керування електричним двигуном....	55
	Посилання на джерела інформації.....	57
	Рекомендована література.....	57
6.	ДОДАТКИ	59
6.1.	Приклади вирішення задач автоматизації інженерних розрахунків та моделювання.....	59
6.1.1.	Розрахунок площі поперечного перетину, об'єму, маси та вартості стандартного та нестандартного сортаменту.....	59
6.1.2.	Розробка програми моделювання поведінки потоку рідини при зміні параметрів трубопроводу за критерієм Рейнольдса.....	80
6.1.3.	Моделювання дії штучного інтелекту для визначення округлих форм на заданому зображенні	91

ВСТУП

В межах лабораторного практикуму з дисципліни «Основи математичного моделювання фізично різнорідних систем» передбачено проведення лабораторних робіт. Роботи мають виконуватись як на комп'ютерах, так і на лабораторних стендах. При роботі на комп'ютерах вирішують задачі які спрямовані на розроблення програм для моделювання. Лабораторні стенди використовуються для фізичного моделювання і експериментального визначення характеристик елементів та систем.

Метою лабораторних робіт є: поглиблення розуміння принципів будови фізично різнорідних елементів та систем; отримання навичок та досвіду розроблення програм виконання розрахунків та моделювання; моделювання дії і дослідження фізичних елементів та систем; представлення дії фізичних елементів за допомогою математичних моделей.

Завдання для лабораторних робіт сформовані таким чином, щоб студенти, які не мають навичок і досвіду програмування, змогли від ознайомлення з процесом створення програм перейти до вирішення практичних задач.

Серед цих задач є такі, як візуальне представлення результатів; забезпечення взаємодії розробника або користувача з активованою програмою; розробка алгоритмів розрахунків та моделювання; керування ходом обчислювального процесу; використання стандартних процедур моделювання динамічних процесів. Частина завдань лабораторних робіт стосуються фізичного моделювання дії елементів та систем в цілому. Вони мають допомогти студентам зрозуміти зв'язок між теорією та практикою. В ході виконання робіт студенти мають навчитися проводити експерименти для визначення характеристик елементів, навчитися представляти характеристики математичними залежностями, а також перевіряти точність отриманих результатів. Такий підхід має підготувати студентів до вирішення більш складних та спеціальних задач моделювання.

Поступове ускладнення завдань лабораторних робіт передбачає також і самостійну роботу студентів. Вона полягає у попередній домашній підготовці до лабораторних занять, творчий підхід при розробці алгоритмів, програм, принципів схем стендів та методик проведення досліджень. Також студенти мають самостійно оформлювати протоколи виконаних робіт, В протоколах мають бути відображені тема роботи, основні відомості, постановка задачі, отримані результати, їх аналіз та сформульовані висновки. Результати виконання кожної лабораторної роботи з відповідним протоколом мають бути захищені.

1. ЗАГАЛЬНІ ВІДОМОСТІ

1.1. Фізично різні системи та задачі моделювання

Невід'ємною частиною підготовки фахівців з машинобудування є моделювання технічних систем. Під моделюванням зазвичай розуміють імітацію дії системи за допомогою її моделі в штучних умовах. Це може бути як фізичне так і математичне моделювання. У будь-якому випадку вирішення задач моделювання стосується конкретного об'єкту, який, як правило, має певні особливості своєї будови і також особливості моделювання.

В подальшому, будемо розглядати об'єкти, які представляють собою фізично різні технічні системи певного призначення. До фізично різних систем відносять системи, які мають у своєму складі компоненти різної природи: - механічні, газові, рідинні, електричні, електронні та програмні. Складність системи, типи компонентів, їх кількість і зв'язки між ними залежать від функціонального призначення технічної системи. Для формування глибшого уявлення про такі системи треба визначити в чому полягає призначення окремих компонентів. На загальному рівні таких призначень є кілька і вони стосуються дій над потоками інформації і енергії, які проходять через систему. Вказані дії є наступні: отримання інформації або енергії; перетворення; розвітлення потоків; з'єднання потоків; перенесення або транспортування. Розглянемо кілька прикладів виконання дій компонентами різної природи. Наприклад, тверде тіло, як механічний компонент, у разі його підняття, накопичить певну кількість потенційної енергії. В цьому випадку можна стверджувати, що тверде тіло при його підніманні переносить енергію з початкового місця розташування до кінцевої позиції. Крім цього, тверде тіло перетворює або змінює енергію, яка до нього прикладається з кінетичної на потенційну. У випадку, якщо твердому тілу придати певну швидкість, воно змінить потенційну енергію, яка до нього прикладається на кінетичну енергію і також буде переносити її в просторі. Інший фізичний компонент – камера, при її наповненні газом або рідиною під тиском, змінює кінетичну енергію потоку рідини або газу на потенційну енергію тиску в середині камери. Також при витіканні рідини або газу з камери під дією внутрішнього тиску потенційна енергія буде перетворена на кінетичну і вона також буде перенесена на певну відстань по трубопроводу разом з носієм – рідиною або газом. Аналогічним чином, можна розглядати і електричні компоненти, наприклад, електричний конденсатор може перетворювати енергію потоку електронів в електричний

потенціал і створювати можливість для її перенесення по електричним каналам.

Використання подібних компонентів в технічних системах потребує розрахунків або моделювання їх дії. В свою чергу, розрахунки і моделювання ґрунтується на фізичних законах, яким підпорядковується дія компонентів відповідної природи. Наприклад, дія механічних компонентів відповідає законам Ньютона і Гука, а дія рідинних компонентів відповідає законам Паскаля, Бернуллі, Ейлера та ін..

Для виконання певних функцій у складі системи компоненти різної природи можуть поєднуватись у модулі. Для прикладу, розглянемо модуль – гідравлічний циліндр, який комбінує в собі два фізично різнорідних компоненти (рис.1.1). В ньому задіяні гідравлічні та механічні компоненти – рідина та тверде тіло. При подаванні рідини під тиском в ліву порожнину гідроциліндру при нерухомому поршні в ній відбувається збільшення тиску. Тиск за законом Паскаля починає діяти на всі поверхні внутрішнього об'єму лівої порожнини циліндру. Результатом дії цього тиску на площу поршня є осьова сила, яка, відповідно до закону Ньютона, змушує поршень рухатись. При цьому ріднина з правої порожнини буде витискатися до баку. Функцією такого модулю є перетворення гідравлічної енергії в механічну. Аналогічним чином

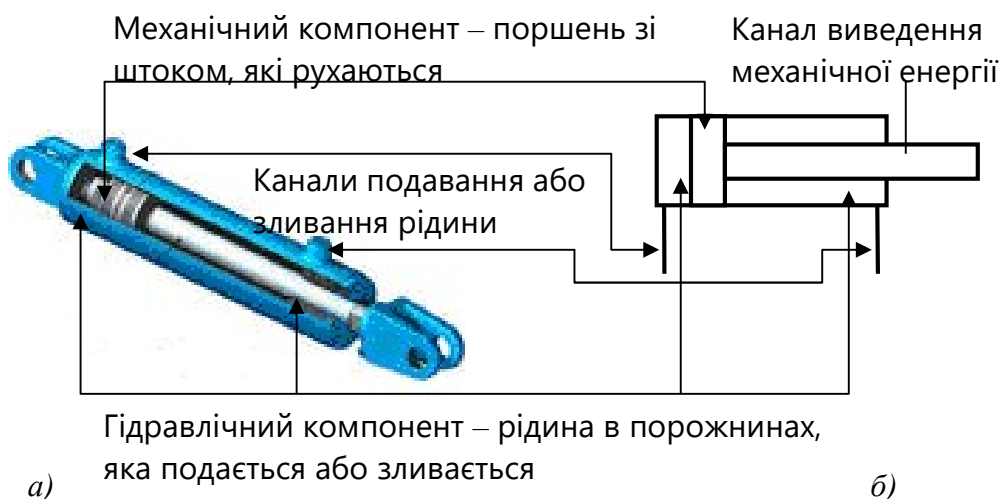


Рис.1.1. Гідравлічний циліндр як приклад фізично різнорідного модулю: а – зовнішній вигляд; б – принципове позначення на схемах

може бути розглянуто також взаємодію компонентів іншої природи. Різнорідність компонентів у складі технічної системи, наприклад, одночасно механічних, гідравлічних та електричних, призводить до необхідності враховувати і відповідний комплект фізичних законів. Це суттєво ускладнює вирішення задач моделювання.

Особливостями фізично різномірних систем, які розглядаються в цьому курсі, є безперервність їх дії – за своїми функціональними ознаками це системи стабілізації, наприклад, рівня води в парогенераторі, або частоти обертання газової турбіни; системи програмного керування, наприклад, система катапультування пілота; системи слідкуючого типу, наприклад система керування рулем висоти літака. Складність проектування таких систем призводить до значних витрат часу та ресурсів. Математичне моделювання дозволяє суттєво скоротити ці витрати.

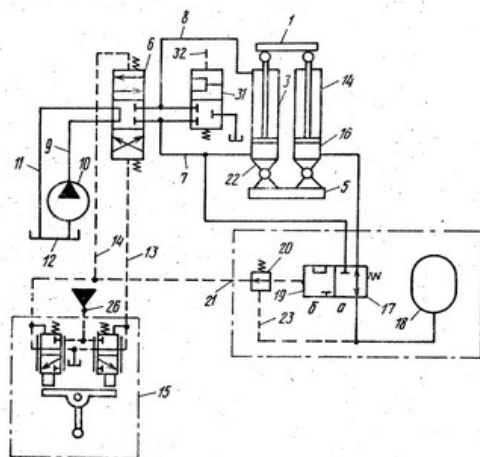
Одною з основних задач моделювання є визначення взаємозв'язку параметрів технічних систем з їх динамічними характеристиками. Дисципліна орієнтована на вирішення задач побудови математичних моделей і дослідження характеристик компонентів в залежності від їх параметрів та умов експлуатації. Також формується підґрунтя до розуміння основ побудови математичних моделей систем в цілому.

1.2. Приклади систем, проблеми та підхід до їх вирішення

Фізично різномірні технічні системи широко використовують в промисловому обладнанні та машинах різного призначення. Наприклад, це екскаватори (рис.1.2 а), транспортні та військові літаки (рис.1.3).



а)



б)

Рис.1.2. Зовнішній вигляд екскаватора (а)[1] і схема його фізично різномірної системи керування (б)[2]

Призначенням таких систем є забезпечення можливостей операторам або пілотам керувати машинами у всіх робочих та екстремальних режимах і можливих умовах експлуатації. Безпосередньо технічні системи забезпечують зв'язок між оператором (пілотом) та керуючими органами машин, або

керування в автоматичному режимі. Для розуміння дії таких систем їх представляють за допомогою принципів схем, які містять в собі позначення фізично різноманітних елементів і відображають зв'язки між ними. Прикладом такої схеми є система керування екскаватором (рис.1.2 б) або одна з систем керування літаком (рис.1.4).

В наведених схемах представлені механічні та гідравлічні компоненти (рис.1.2 б), а також в схемі (рис.1.4) додатково представлені електричні, електронні та програмні компоненти.



а)



б)

Рис.1.3. Вигляд транспортного літака 74TK-200 ("Антонов-Канада")[3 (а) та військового літака F-16[4] (б), які оснащені фізично різноманітними слідкуючими системами керування рулевими поверхнями

Складність проектування таких систем обумовлена суттєвими відмінностями між компонентами різної природи, яка полягає різноманітності процесів, які в них відбуваються як з точки зору фізичних законів, так і з точки зору їх часових характеристик. Це призводить до значних часових витрат на

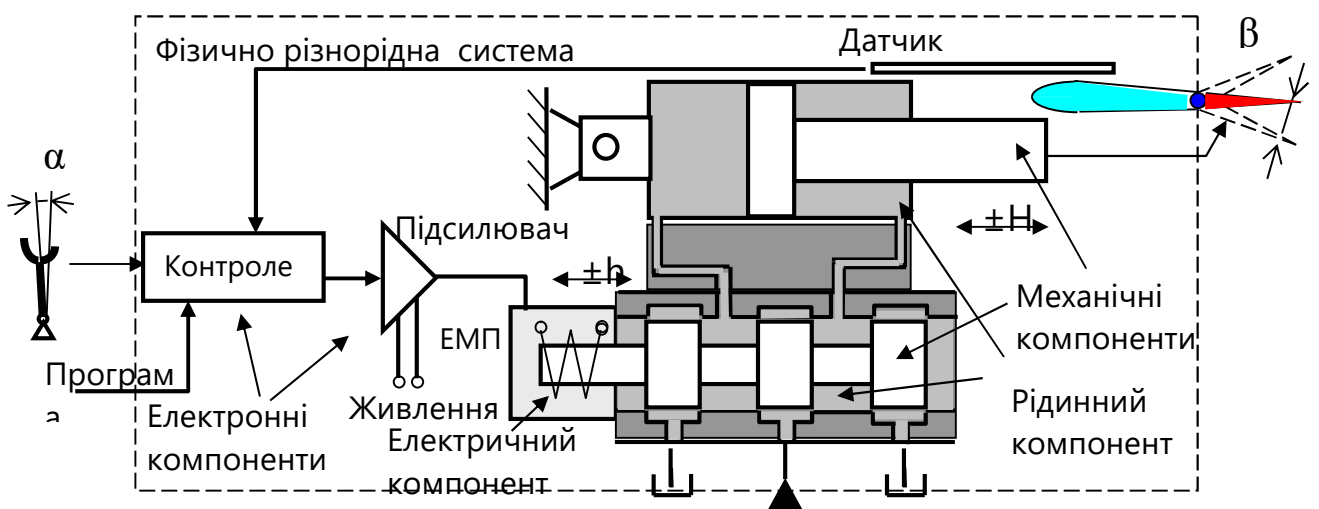


Рис. 1.4. Приклад схеми системи керування рулем літака, яка містить фізично різноманітні компоненти

узгодження процесів роботи різнорідних компонентів в одній технічній системі. Застосуванням методів математичного моделювання дозволяє зменшити або виключити використання фізичних зразків та обладнання при створенні таких систем.

Для моделювання дії компонентів і систем в цілому, а також для виконання інженерних розрахунків широко використовують програмні засоби. Застосування таких засобів потребує розуміння процесу взаємодії розробника з комп'ютером в процесі створення програм.

1.3. Загальний процес розроблення програм моделювання

Задачою дисципліни є навчити студентів основам побудови математичних моделей фізично різнорідних компонентів та систем і за їх допомогою досліджувати процеси їх дії. Для виконання такої роботи потрібно використовувати відповідні інструменти. Таким інструментом є мова програмування, яка дозволяє створювати програми для розрахунків та моделювання. На загальному рівні технологія вирішення таких задач представлена наступним чином (рис.1.5) Розробник програм на основі заданого алгоритму розрахунку або алгоритму роботи моделі створює програму. Після цього програму передають операційній системі комп'ютера і запускають. В цей момент запускається обробник програм, який має

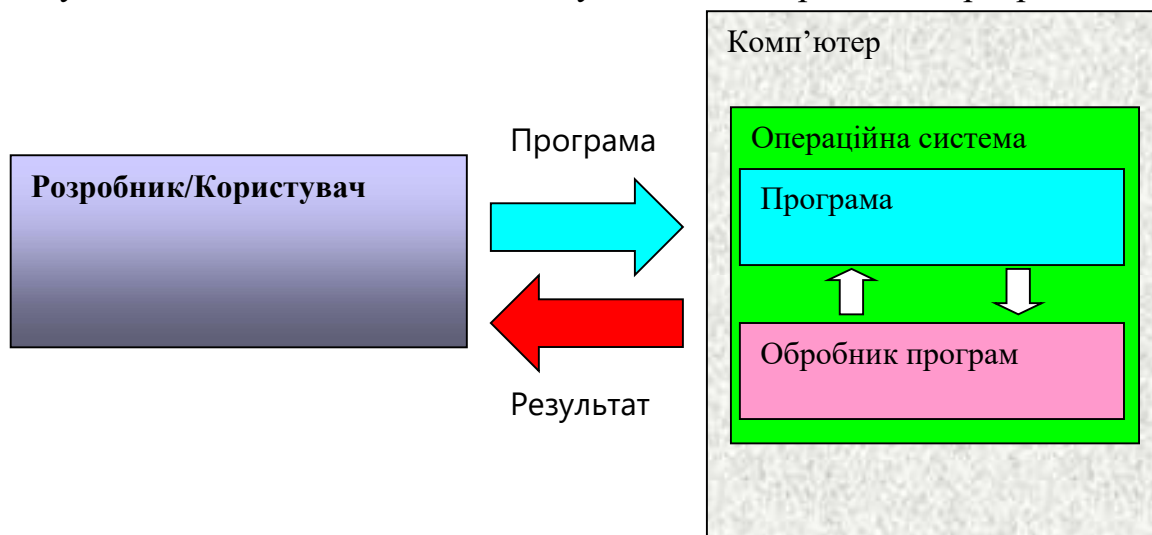


Рис.1.5. Загальна схема взаємодії Розробника програм з Обробником програм в операційному середовищі комп'ютера

виконувати інструкції, які задано в наданій програмі. Від того, наскільки розробник програми і обробник програм порозуміються, залежить результат

виконання програми. В процесі виконання програми можуть відбуватися взаємодії між Розробником та Обробником. В ході взаємодій відбувається налаштування програми. Програма, яка працює коректно, може бути передана Користувачеві для її застосування при вирішенні задач, які відповідають призначенню програми.

Існує значна кількість різноманітних мов програмування. В той же час, важливо зрозуміти який набір засобів потрібен Розробнику для створення програм розрахунків і моделювання дії технічних елементів та систем. Вказані засоби зручно об'єднати у дві групи: - засоби формування обчислювального процесу та засоби забезпечення взаємодії з користувачем (рис.1.6). Засоби формування обчислювального процесу містять засоби виконання математичних операцій, засоби забезпечення багаторазового виконання операцій, засоби керування ходом обчислювального процесу та пам'ять. В свою чергу, засоби взаємодії з користувачем об'єднують засоби отримання інформації від користувача та засоби передачі інформації користувачеві.

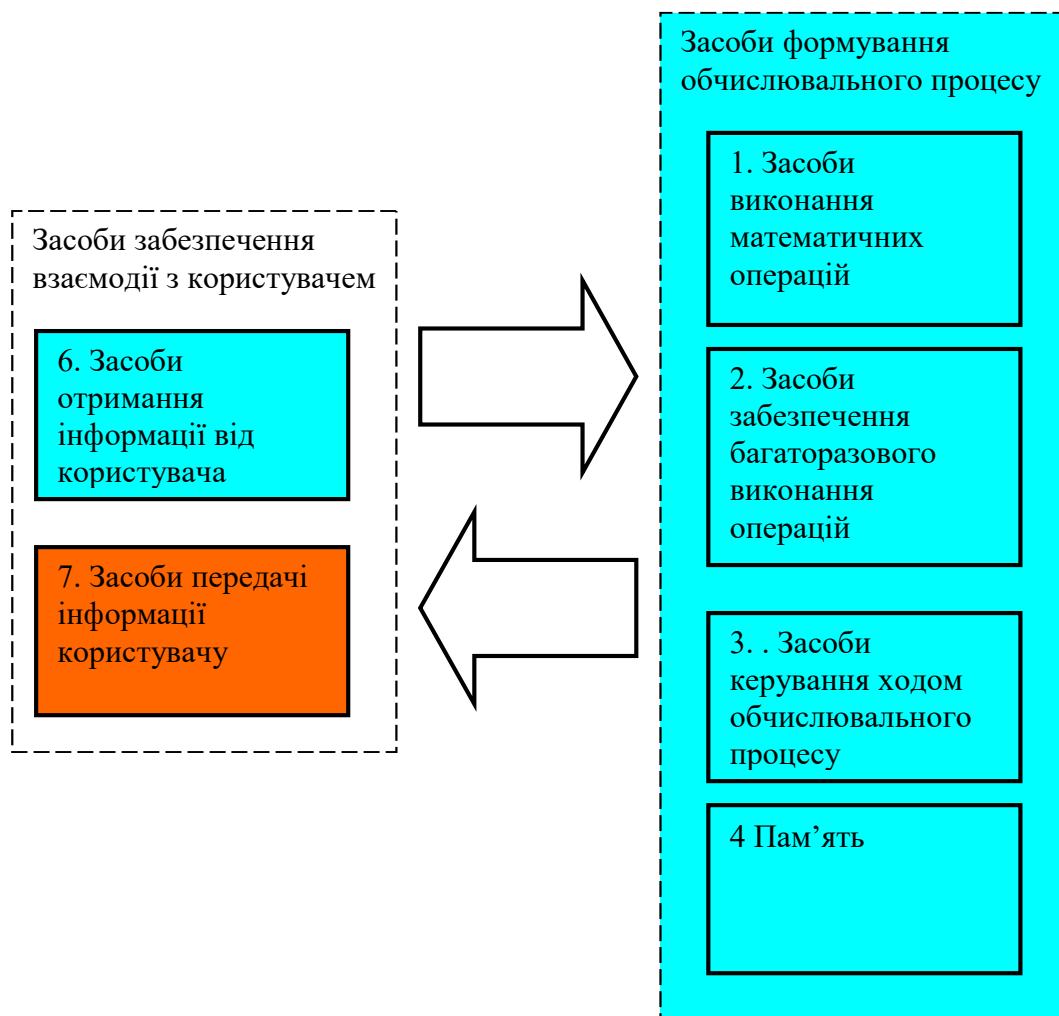


Рис. 1.6. Функціональна схема засобів розроблення програм виконання інженерних розрахунків і моделювання

Для безпосереднього розроблення програм потрібно використовувати конкретну мову програмування, в якій наведені вище засоби реалізуються відповідними командами. Одною з популярних мов програмування для розробки програм є мова програмування С, яка з деякими змінами покладена в основу пакету Матлаб [5].

2. ІНСТРУМЕНТИ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

2.1. Засоби пакету MatLab для обчислень та моделювання

2.1.1. Засоби формування обчислювального процесу

2.1.1.1. Засоби виконання обчислювальних операцій

Автоматизація виконання дій, по перше, потребує наявності засобу, що потенційно може ці дії виконувати, а по друге, мають бути також засоби, що дозволяють однозначно пояснити що, коли і як потрібно робити. Перший засіб умовно може бути названий «Обробником програм», а другий – «мовою спілкування» з Обробником.

Дії які потрібно автоматизувати фіксуються у файлі у вигляді тексту за допомогою мови спілкування. Цей файл передається для виконання обробнику програм. Обробник програм виконує дії, які передбачені у тексті програми. Такими діями можуть бути обчислювальні операції, повторення операцій, аналіз та прийняття рішень до виконання тих, або інших альтернативних дій.

Обчислювальні операції, що можуть використовуватися, відповідають математичним операціям. Кожна математична операція може бути замовлена для виконання відповідним символом. Математичні операції, що можуть використовуватися по відношенню до звичайних констант або змінних параметрів позначаються таким чином:

- + - додавання;
- - віднімання;
- * - множення;
- / - розділення;
- n - взведення у ступінь n .

У разі необхідності застосування математичних операцій для змінних у векторному вигляді необхідно використовувати іншу групу математичних операцій, що позначаються додатковою точкою перед символом операції, наприклад, для взведення вступінь це виглядає наступним чином:

- $\cdot ^n$ - взведення у ступінь n векторної змінної.

2.1.1.2. Засоби забезпечення багаторазового виконання операцій

У разі необхідності багаторазового використання однакових операцій в обумовленій черговості, застосовуються параметри та спеціальні ключові слова, що визначають відповідні дії обробника програми.

У якості параметрів використовуються ідентифікатори. Ідентифікатори представляються у вигляді позначення, що записується як єдиний буквенний символ, кілька символів (без пробілу поміж ними), або символів та цифр.

Приклад: `x`, `x1`, `x3`, `tz` та ін.

Ключові слова які дають змогу передати обробнику програми завдання про багаторазове (циклічне) виконання потрібного переліку операцій застосовуються парами.

Перше ключове слово та заключне ключове слово. У тексті програми вони можуть додатково позначатися іншим кольором. Операції, що розташовані поміж ними обробник програм буде виконувати потрібну кількість разів.

Потрібна кількість обчислень вказується за допомогою запису, що слідує безпосередньо за першим ключовим словом. Вона складається з параметру циклу, якому присвоюється початкове значення, символу, що є роздільним, значення, що завдає шаг прирощення параметру циклу, ще одного символу, що є роздільним, та кінцевого значення параметру циклу.

Приклад.

```
for x=0:1:10  
    y=x^2;  
end
```

Пояснення прикладу. Ключові слова **for** і **end**. Операція, що потребує багаторазового виконання `y=x^2;` Параметр циклу `x`. Початкове значення параметру `0`. Шаг прирощення параметру `1`. Кінцеве значення `10`. У разі виконання наведеного прикладу обробник програми виконає вказану операцію 11 разів.

Існує також інша пара ключових слів (**while**, **end**), що дає можливість передати обробнику програм завдання на багаторазове виконання операцій.

Приклад.

```
x=0;
```

```
while x<13
    y=x^2;
    x=x+1;
end
```

Пояснення прикладу. Вказані ключові слова забезпечують багатократність виконання операції $y=x^2$.

Операції $x=0$; перед початком циклу та $x=x+1$; у середині циклу забезпечують отримання початкового значення та прирощення параметру циклу.

2.1.1.3. Засоби керування ходом обчислювального процесу

Для керування ходом обчислювального процесу застосовуються спеціальні ключові слова.

Такими словами є **if else elseif** та **end**.

Перше ключове слово має використовуватися в комбінації з останніми.

Перше ключове слово передбачає існування додаткової умови для її перевірки, а також опис операції, яка потребує виконання у разі відповідності умови. У разі не відповідності умови операцію буде пропущено.

Наприклад:

```
if x<6 y=x+2; end
```

Пояснення. Обробник програм під час виконання наведеного рядка програми перевіряє значення x . Якщо значення x менше 6, то виконується операція $y=x+2$, в іншому випадку операція пропускається.

Комбінація першого і другого ключових слів дозволяє виконувати альтернативну операцію.

Приклад:

```
if x<6 y=x+2;
else y=x-3; end
```

Пояснення. У разі виконання умови обробляється операція $y=x+2$, у протилежному випадку виконується операція $y=x-3$;

Використання комбінації ключових слів першого і третього можливо продовжити використання умов.

Приклад:

```
if x<6 y=x+2;
elseif x>3 y=x-3; end
```

Пояснення. В наведеному прикладі у разі невиконання першої умови буде перевірятися друга умова. Якщо друга умова виконується обробляється операція $y=x-2$, у протилежному випадку вказана операція ігнорується.

Використання ключового слова **end**, дозволяє забезпечити виконання кількох операцій послідовно для потрібних умов.

Приклад:

```
if x<6 y=x+2;  
  elseif x>3 y=x-3; z=x+16; end
```

Пояснення. У разі невиконання першої умови буде перевірятися друга умова. Якщо друга умова виконується обробляються операції $y=x-2$, і $z=x+16$, у протилежному випадку вказані операції ігноруються.

2.1.2. Забезпечення взаємодії з користувачем

2.1.2.1. Засоби отримання інформації від користувача

В ході виконання програми може бути передбачена можливість завдання користувачем конкретних значень змінних або параметрів. Отримання потрібної інформації від користувача забезпечується за допомогою спеціальних команд **input()** та **menu()**.

Отримання числового значення

Перша команда здійснює запит числового значення, а після його завдання користувачем вказане значення може бути присвоєне необхідній змінній.

Приклад.

```
x =input('x=');
```

Пояснення. Виконання обробником програм вказаного запису призведе до виводу на екран комп'ютеру строки:

```
x=
```

після чого комп'ютер буде знаходитися у стані очікування до моменту завдання значення користувачем. Після отримання значення, наприклад 13, воно буде присвоєно змінній x і воно буде збережено. В оперативній пам'яті комп'ютера змінна x буде зберігати значення 13.

Отримання інформації про прийняте рішення

Команда `menu()` дозволяє представити користувачу інформацію для прийняття рішення, яке може використовуватися, наприклад, для керуванням ходом обчислювального процесу.

Приклад.

```
z=menu('Заголовок','Кнопка_1','Кнопка_2');
```

Пояснення. Вказаний запис призведе до виводу на екран комп'ютера віконця з трьома кнопками, на яких будуть відповідні записи, **Заголовок**, **Кнопка_1** та **Кнопка_2**. Після виводу меню, комп'ютер буде залишатися у стані очікування до надходження дії від користувача.

Користувач має можливість прийняти рішення про яке обробник програм дізнається по номеру натиснутої користувачем кнопки.

У разі натиснутої кнопки 2, змінній **z** буде присвоєно значення **2**. Вказане значення буде використане в програмі для вибору наступних команд, відповідно до бажаного ходу обчислювального процесу.

2.1.2.2. Засоби представлення інформації для користувача

Взаємодія з комп'ютером потребує представлення обробником програм певної інформації для користувача.

Інформація може представлятися у вигляді текстових рядків, чисел, або графіків. Графіки у свою чергу можуть бути виведені у потрібне поле у вікні графічного виводу. Крім того, графіки можуть мати надпис та розмітку по осях координат. Для кожного випадку використовуються відповідні команди.

Вивід інформації у вигляді тексту

Для виводу тексту безпосередньо на екран комп'ютера може використовуватись команда :

```
disp('текст')
```

Пояснення. В результаті виконання команди на екран комп'ютера буде виведено слово

текст

Для виводу текстової стрічки в поле графічного вікна на екрані комп'ютера може використовувється команда

```
h=text(x,y,'Текст для виводу','FontSize',12);
```

Пояснення, x, y – координати лівого нижнього кута поля виводу тексту, 'Текст для виводу' – текст який буде виводитись у поле вікна, 'FontSize',12 – завдає розмір шрифту.

Вивід числової інформації

Числова інформація виводиться у поле вікна аналогічно виводу текстової інформації, але перед виводом числова інформація має бути перетворена у текстову. Для цього використовується команда **num2str(x)**, де x – змінна, що має числове значення.

Вивід інформації у вигляді графіків

Для представлення інформації у графічному вигляді можуть використовуватися операція **plot(x,y)**, де x, y – координати точки.

Приклад.

plot(20,30);

Пояснення. Обробник програм виведе вікно з полем для графіку з розміткою координатних осей. На полі буде зображено точку з координатами $x=20, y=30$.

Виведення надпису для графіка

Виведення надпису для графіка забезпечується командою:

title('Графік залежності...');

Пояснення. У вікні над полем графіка буде виведено напис за аналогією з рис.2.1.

Очистка вікна та осей координат

axis('off');

Пояснення. На полі графіку будуть видалені вісі.

Очистка вікна по за межами поля графіка

delete(gca);

Пояснення. Поле графіку буде очищене від попередньо побудованих графіків.

Зображення координатної сітки **grid**

Пояснення. На поле графіку буде нанесено координатну сітку.

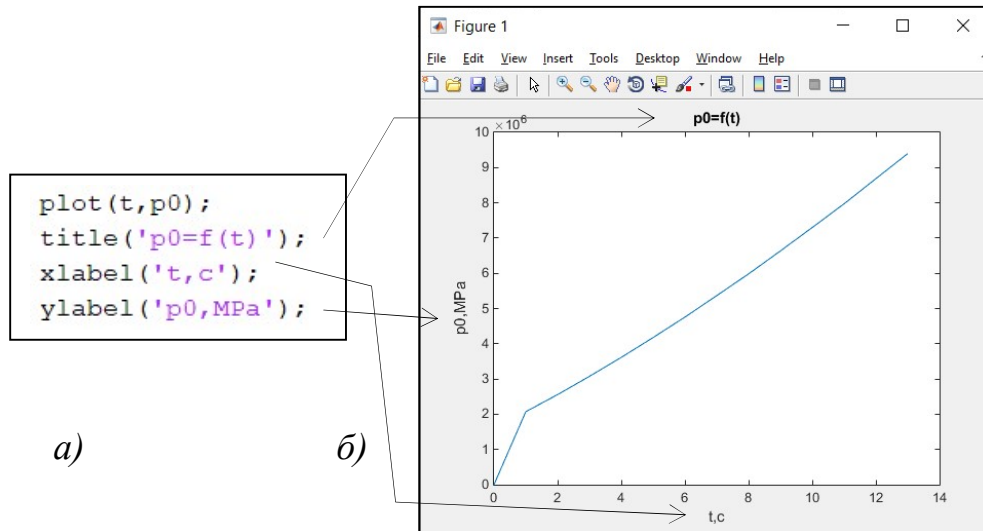


Рис. 2.1. Фрагмент програми (а) та відповідний вигляд вікна (б) при побудові графіків в пакеті Матлаб

Виведення поля графіка у потрібне місце вікна **subplot(2,3,6);**

Пояснення. Розбивка поля вікна виводу на 2 рядки, 3 стовбця, і планування виводу графіку у 6 віконце, яке буде розташоване у нижньому рядку праворуч.

subplot(2,3,[4,5,6]),

Пояснення. Розбивка поля вікна виводу на 2 рядки, 3 стовбця, і планування виводу графіку у 4,5 та 6 віконце одночасно, яке буде розташоване у нижньому рядку.

Збереження попередніх даних на полі графіка **hold on**

Пояснення. При розташуванні цієї команди перед командою **plot()** при виведенні інформації в поле графіка попередня інформація, яку було вже виведено в поле графіку, буде збережена.

Представлення додаткової інформації для користувача

Малювання або видалення координатної сітки на полі графіку

grid on - малювання сітки;
grid off - видалення сітки.

Нанесення підписів по осях X та Y:

xlabel('Час')
ylabel('Переміщення')

Пояснення. Результат виконання команд такий, як наведено в прикладі (рис.2.1).

Розміщення тексту в заданій позиції
text(x,y,'Графік зміни переміщення маси в часі');
або
text(3.05,0.16,'Графік переміщення');

Пояснення. Текст наданий в лапках буде виведено у вікно. Координати виводу відповідають значенням x,y. Відлік відповідає локальній системі координат вікна з початковими значеннями 0,0 у верхньому лівому куту вікна.

Задавання типу, кольору та символів для позначення на лінії графіків
plot (x, y, 'колір, тип лінії, тип точок');
Приклад.
plot (x, y, 'm -- *');

Пояснення. При виведенні графіків для змінних x, y буде використано фіолетовий колір, штрихова лінія, з позначенням точок символом «*».

Нижче наведені варіанти завдань для типів ліній, символів точок та кольорів.

Символи для завдання типу ліній графіків

-	неперервна
--	штрихова
:	пунктирна
-.	штрих-пунктирна

Символи для завдання типу точок на графіку

.	- крапка
+	- плюс
*	- зірочка
o	- коло
x	- хрестик

Символи для завдання потрібного кольору лінії на графіку

c	- блакитний
m	- фіолетовий
y	- жовтий
r	- червоний
g	- зелений
b	- синій
w	- білий
k	- чорний

Побудова тривимірних графіків

Для побудови тримірних графіків в Матлаб можуть бути використані наступні три різні функції **plot3 ()**, **mesh ()** і **surf ()**.

plot3 (X, Y, Z),

mesh(X,Y,Z).

surf(X,Y,Z).

Пояснення. Кожен з аргументів цих функцій **X,Y,Z** є одномірним масивом, при цьому **X,Y** – є незалежними, а **Z** є масивом значень, які відповідають заданим **X,Y**. Для побудови графіків треба попередньо задати вказані масиви, а потім викликати функцію **meshrid(X,Y)** побудови області в координатах **x,y**. Після чого треба задати відповідний ним масив **Z**

і викликати одну з функцій **plot3()**, **mesh()** і **surf()**.

Графік, побудований функцією **plot3()** представляється лініями в координатах **z,y**.

Графіки, побудовані функціями **mesh()** і **surf ()** представляються поверхнями у вигляді сітки, зі значеннями z у вузлах сітки.

Підпис кожної вісі можна зробити відповідними функціями

xlabel('x'), ylabel('y'), zlabel('z').

Приклад. Побудувати графік функції $z(x,y)=y^2-x^3$ для інтервалів значень x та y $x \in [-5,5], y \in [-5,5]$.

Рішення.

Формуємо сітку в площині x, y

[X,Y]=meshgrid(-5:0.4:5,-5:0.4:5);

Розраховуємо значення z для масиву Z

Z=Y.^2-X.^3;

Будуємо тримвимірний графік функції

mesh(X,Y,Z);

Підписуємо вісі

xlabel('x'); ylabel('y'); zlabel('z');

Отримуємо результат

У разі необхідності завдання потрібних масштабів по осях координат, використовуємо функцію

axis([xmin xmax ymin ymax zmin zmax])

Таким чином, розглянутий комплект засобів дозволяє задавати команди на виконання очислювальних операцій, команди багаторазового виконання операцій, команди керування ходом обчислювального процесу, команди використання пам'яті, а також команди для забезпечення взаємодії користувача з обробником програм в ході її виконання. Тепер потрібно навчитися їх використовувати для розробки програм.

2.1.3. Побудова програм моделювання процесів

2.1.3.1.Формування алгоритму

Інженерні розрахунки або моделювання виконуються відповідно до наперед завданих методик. Методики регламентують наявність початкових даних, обумовленої послідовності дій, прийняття рішень відповідно до отриманих проміжних результатів що до вибору варіантів, у випадку

можливих альтернативних дій, а також отримання кінцевих даних. Вказані дії та їх послідовність можуть бути відображені у вигляді алгоритму за допомогою графічних засобів, наприклад (рис. 2.2).

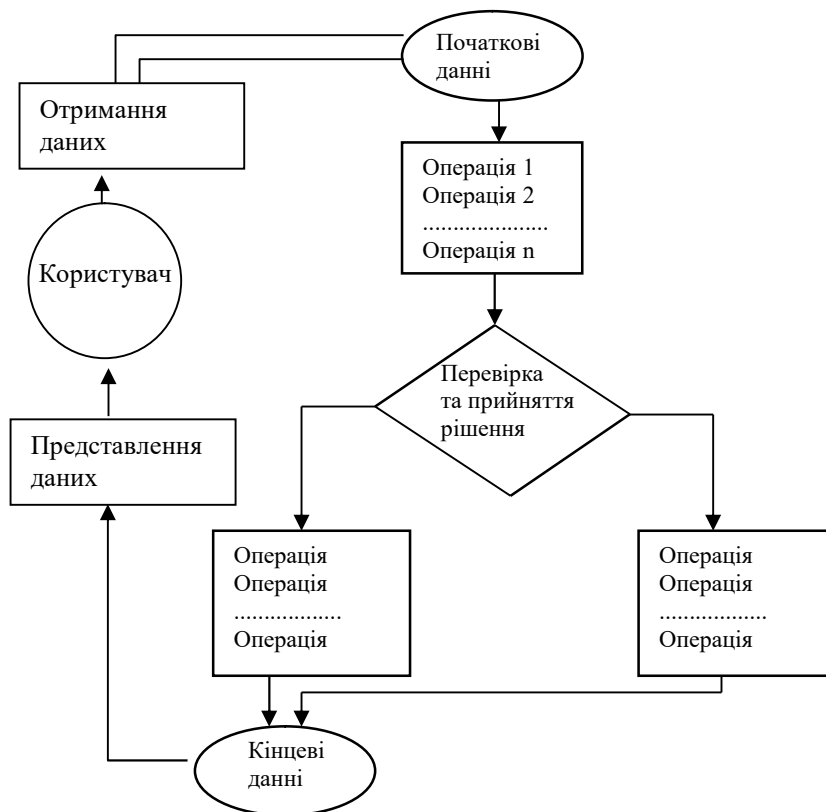


Рис.2.2. Приклад представлення алгоритму виконання розрахунків

Застосування графічних засобів представлення алгоритмів полегшує розуміння задачі, яка вирішується, і дозволяє в подальшому використовувати різні мови програмування, яким володіє розробник. Алгоритмічне представлення також дозволяє розділяти складну задачу на більш прості і вирішувати їх різними розробниками одночасно

2.1.3.2. Перетворення алгоритму в програму

Перетворення алгоритму в програму можна розглядати як наповнення схеми алгоритму змістовними складовими.

Програму формують за допомогою засобів, що були наведені вище. При цьому, використовуються властивості обробника програм виконувати дії у суворій послідовності згідно послідовності їх зчитування. Іншими словами організація заданої послідовності дій передається обробнику програм шляхом послідовного запису потрібних дій безпосередньо у програмі. Також, для

забезпечення потрібної черговості виконання дій використовують засоби керування ходом обчислювального процесу такі як **if else elseif** та **end**. Початкові данні отримуються від користувача, а кінцеві данні представляються користувачу також за допомогою відповідних засобів.

У якості прикладу розглянемо формування програми на основі заданого алгоритму. Покажемо це для одного з його фрагментів. Заданий алгоритм представляє вирішення задачі дослідження поведінки графіку функції в залежності від значень коефіцієнтів (рис.2.2). Програма має забезпечити можливість завдання коефіцієнтів користувачем, автоматизувати обчислення значень заданої функції, та представляти результати у графічній формі. Фрагментом алгоритму, який маємо перетворити в програму є блок реалізації перевірки та прийняття рішення (рис.2.3). Для перетворення цього фрагменту

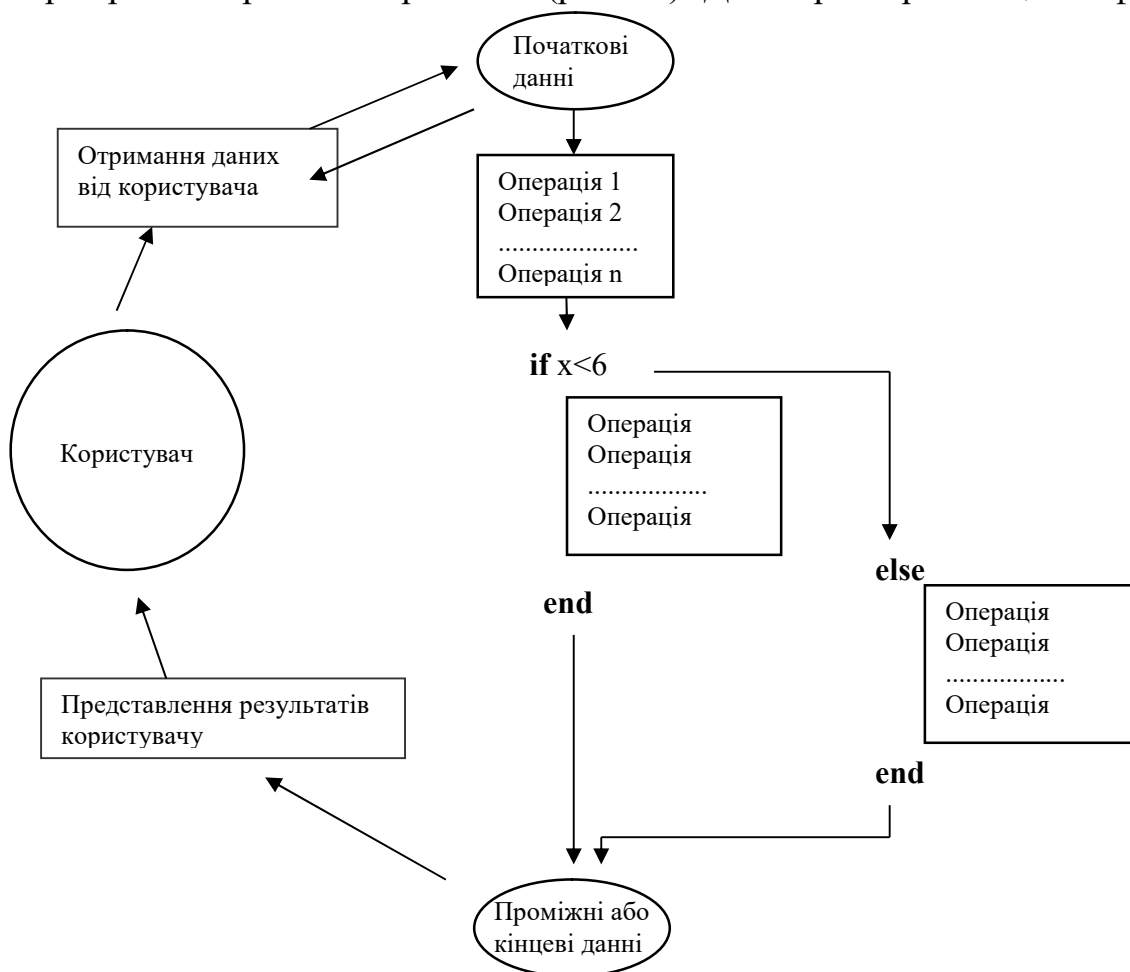


Рис.2.3. Алгоритм проведення розрахунків, що забезпечує інтерактивну взаємодію з користувачем

алгоритму в програму використовуємо засоби організації ходу обчислювального процесу такі як оператори «if», «else» та «end». В результаті

запису цих операторів з врахуванням потрібних операцій, які вказані в алгоритмі для кожної з ситуацій, отримуємо фрагмент програми:

```
if x<6  
Операція 1;  
Операція 2;  
.....  
Операція n;  
end  
else  
Операція 1e;  
Операція 2e;  
.....  
Операція me;  
end
```

Аналогічним чином, кожен блок алгоритму має бути замінено відповідними засобами розробки програм з врахуванням ходу обчислювального процесу. Результатом є код програми, який структуровано відповідно до заданого алгоритму. Як правило, кожен блок має доповнюватись коментарями, які суттєво полегшують роботу з програмою при її налаштуванні та удосконаленні.

2.1.4. Стандартні функції для вирішення спеціальних задач

Стандартними функціями є стандартні математичні операції, наприклад, $\sin(x)$, $\cos(x)$, \sqrt{x} та ін., які можуть бути використані в середині обчислювальних блоків. Існують також стандартні функції вирішення систем диференціальних рівнянь, яким, як правило, описується дія фізично різномірних компонентів та систем в цілому. Для більш глибокого розуміння таких стандартних функцій розглянемо один з основних методів пошуку рішень систем диференціальних рівнянь, які лежать в їх основі.

2.1.5. Чисельне рішення диференціальних рівнянь [6]

Постановка задачі Коші.

Є диференціальне рівняння вигляду

$$y^{(n)} = f(t, y, y^1, \dots, y^{(n-1)});$$

З початковими умовами

$$y(t_0) = y_0, y^1(t_0) = y_1, \dots, y^{(n-1)}(t_0) = f(t, y, y^1, \dots, y^{(n-1)}) = y_{n-1},$$

Треба знайти чисельним методом функцію, яка буде представляти рішення цього рівняння.

Для чисельного вирішення, конкретне задане рівняння вищого порядку необхідно представити у вигляді системи рівнянь першого порядку

$$y^1(t) = F(t, y), y(0) = y_0,$$

де y – функція (якщо система складається з n – рівнянь, то y представлено вектором, а y_0 – її початкові значення (у випадку системи рівнянь y_0 також є вектором)

Для вирішення диференційного рівняння або системи рівнянь чисельним методом в пакеті Матлаб можна використовувати стандартну функцію з ім'ям **ode45**.

Формат використання цієї функції наступний.

$$[T, Y] = \text{ode45}(\text{odefun}, [t_0, \text{tend}], y_0, \text{options})$$

де `odefun` – ім'я функції, яка буде забезпечувати обчислення значень правих частин $F(t, y)$ заданого рівняння або системи рівнянь у формі (1). Результатом виконання цієї функції буде значення або вектор значень y при кожному заданому значенні незалежної змінної t . Функцію `odefun` треба задати і розраховувати перед викликом стандартної функції `ode45`.

Параметри `[t0, tend]` задають початкове і кінцеве значення діапазону змін незалежної змінної t .

y_0 – початкове значення або вектор функції, яка буде визначатись чисельно.

`Options` – необов'язковий аргумент, який може використовуватись для задання точності розрахунку. Його значення задане по умовчанням.

Результатом виконання функції `ode45` є масив значень T – значень незалежної змінної t для яких шукались значення функції і масив відповідних їм значень функції y .

Приклад використання стандартної функції `ode45` пакету Матлаб

Для технічної системи математичний опис її дії представлено диференціальними рівняннями

$$\frac{d^2 h}{dt^2} + \frac{c}{m} \cdot h - g = 0.$$

1. Представляємо математичний опис у формі Коши

1.1. Записуємо змінні під знаком диференціала зліва по відношенню до знака рівності

$$\frac{d^2 h}{dt^2} = -\frac{c}{m} \cdot h + g.$$

1.2. Записуємо рівняння у формі системи диференціальних рівнянь першого ступеню

$$\begin{cases} \frac{dV}{dt} = -\frac{c}{m} \cdot h + g; \\ \frac{dh}{dt} = V. \end{cases}$$

Задаємо початкові умови

$$h(0) = 0; V(0) = 0.$$

2. Підготуємо систему рівнянь для використання стандартної функції `ode45`

2.1. Робимо заміну, позначаємо

$$y_1 = V; y_2 = h.$$

2.2. Пишемо програму моделювання

```
//Код програми
//Функція для розрахунку правих частин системи рівнянь
function F=ode_fnc(x,y)
F=[-c/m*y(2)+g; y(1)];

//Виклик стандартної функції чисельного методу
[T, Y] = ode45('ode_fnc', [0,10], [0,0])
```

2.3. Отримуємо результат моделювання

Формат виводу результату такий:

```
[T, Y] =
t0 y1(t0) y2(t0)
t1 y1 (t1) y2(t1)
.....
tk y1(tk) y2(tk)
```

2.4. Додаємо в програмі команди виводу графіків

```
//Побудова графіків за отриманими результатами
plot(T,Y(:,1)); hold on;
plot(T,Y(:,2));
```

Приклад закінчено.

В залежності від версії пакету Матлаб звернення до стандартної функції інтегрування можуть відрізнятися від наведеного в прикладі.

Ще один варіант звернення наведено нижче

```
odefun=@(t,P0)E*q/V           %Опис функції «odefun», яка задає
                                % праві частини рівняння у формі Коші
[t,P0]=ode45(odefun,t,P0);    %Звернення до стандартної функції
                                %«ode45»
plot(t,P0)                   %Побудова графіку
```

Результатом вирішення таких систем рівнянь є процеси, які можуть бути представленими змінами в часі певних фізичних параметрів, наприклад, швидкості та переміщення маятника після його примусового відхилення від вертикальної і вільному коливанні (рис. 2.4)

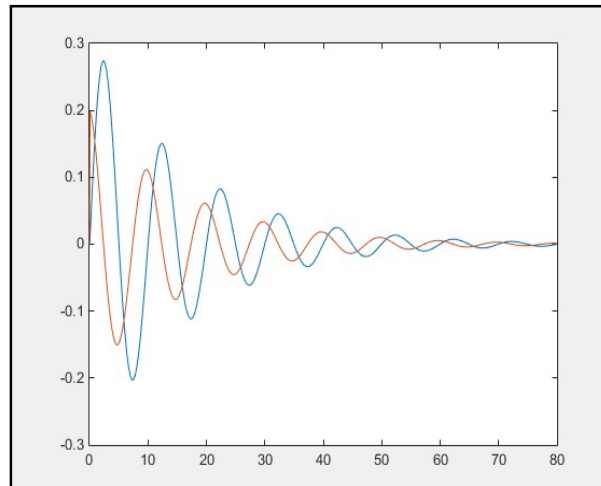


Рис.2.4. Приклад вигляду графіків при моделюванні динамічних процесів

Використання такої стандартної функції дозволяє моделювати процеси в компонентах і системах і суттєво зменшує витрати часу на побудову математичних моделей. Однак при цьому залишаються ще питання про достовірність побудованих математичних моделей, відповіді на які знаходять експериментальним шляхом.

3. ЗАСОБИ І ОСОБЛИВОСТІ ФІЗИЧНОГО МОДЕЛЮВАННЯ

3.1. Обладнання для фізичного моделювання

Вивчення курсу базується на виконанні лабораторних робіт з фізичним обладнанням. При цьому завданнями для студентів є з'ясування принципу дії фізично різномірних елементів та систем, навчання користуватися типовими елементами, з'ясування їх статичних характеристик та створення математичних описів, що дозволяють моделювати дію елементів та систем. Для виконання лабораторних робіт студенти використовують спеціальні набори обладнання, які містять електронні та електромеханічні елементи різного призначення (рис.3.1, рис. 3.2, рис.3.3), стенди для їх складання у функціональні компоненти (рис.3.4), а також комплекти функціональних компонентів систем та вимірювальне обладнання (рис.3.5).

Крім наведених прикладів обладнання при виконанні лабораторних робіт використовуються також вимірювальне обладнання таке як тестери та осцилографи.

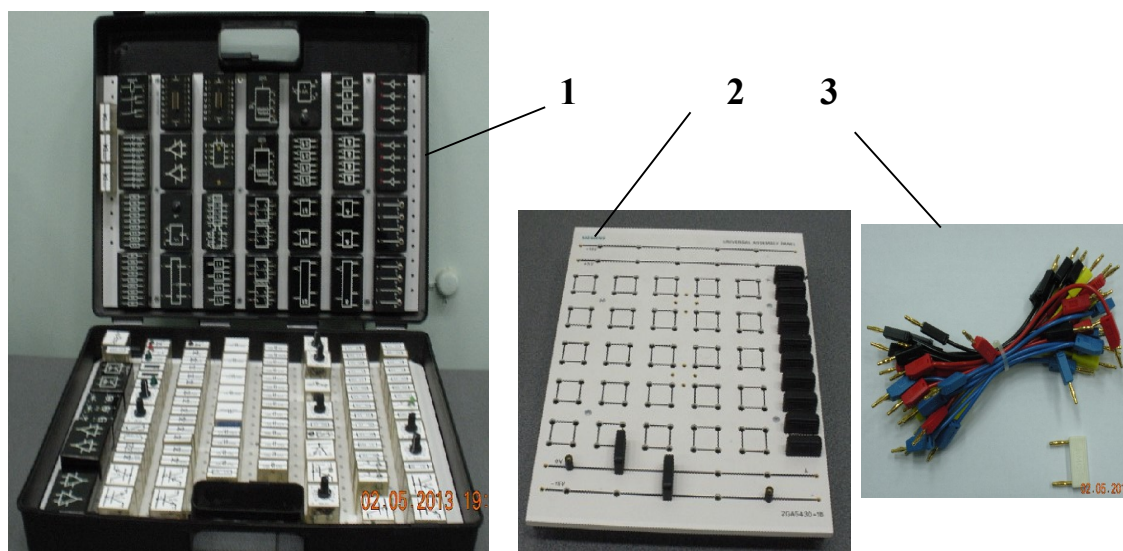


Рис.3.4. Комплект елементів (1), панель для складання (2) та набір дротів (3) для з'єднання елементів, які призначені для макетного моделювання, перевірки роботоспроможності та дослідження елементів та мехатронних систем

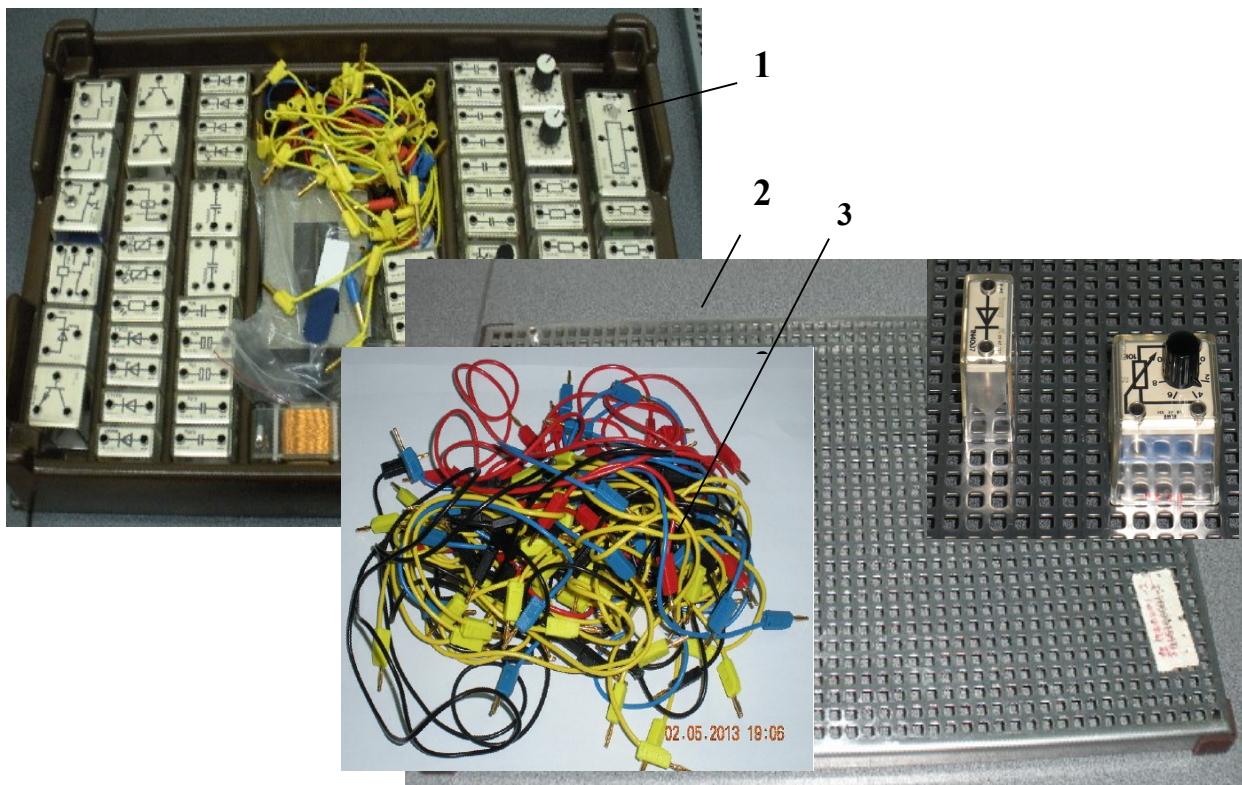


Рис.3.1. Комплект елементів (1), панель для складання (2) та набір дротів (3) для з'єднання елементів, які призначенні для макетного моделювання, перевірки роботоспроможності та дослідження елементів та мехатронних систем

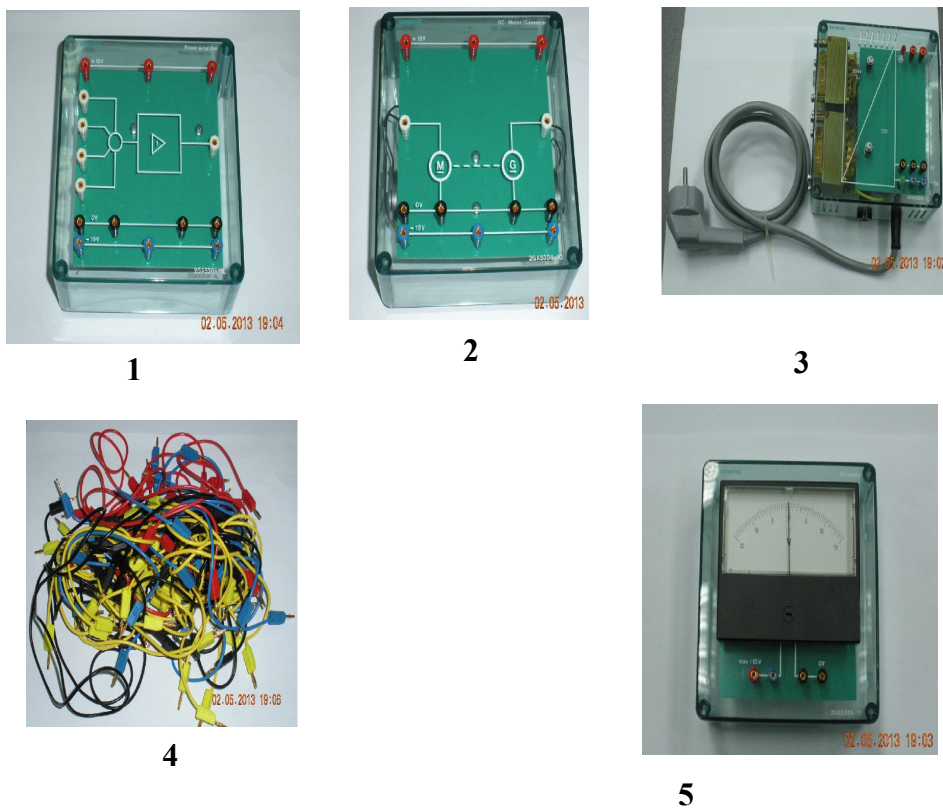


Рис.3.5. Комплект компонентів: - електронний підсилювач з суматором (1), модуль електричний двигун-електричний генератор (2), модуль живлення (3), комплект дротів для забезпечення взаємодії модулів (4), вимірювальний прилад (5)

3.2. Фізичні експерименти та математичні моделі

На відміну від навичок та досвіду побудови програм за допомогою мови програмування фізичні експерименти потребують набуття інших навичок і досвіду. При цьому, важливим є розуміння і дотримання правил техніки безпеки.

Кожен студент перед початком роботи з обладнанням має бути ознайомлений з правилами техніки безпеки, зафіксувати це своїм підписом у відповідному журналі і **ОБОВ'ЯЗКОВО дотримуватись цих правил в ході виконання робіт в лабораторії.**

Основним завданням при виконанні лабораторних робіт з фізичними компонентами систем є встановлення взаємозв'язку між фізичними характеристиками компонентів та їх математичними моделями.

Завдання кожної лабораторної роботи виконуються в наступній черговості:

1. За наданим типом фізичного елемента необхідно зрозуміти принцип його дії та описати його дію реченнями.
2. Визначити, яка фізична величина має подаватися на вхід елемента, а яка є результатом його роботи – вихідною величиною.
3. Пригадати, що є статичною характеристикою, і якою є ця залежність саме для наданого конкретного фізичного елемента.
4. Розробити методичку визначення статичної характеристики конкретного елемента для заданих умов його роботи.
5. Розробити форму таблиці для запису результатів досліджень.
6. Розробити принципову схему стенду для визначення статичних характеристик заданого елемента.
7. Підібрати обладнання та скласти експериментальний стенд.
8. Провести дослідження, фіксуючи результат в таблиці.
9. Побудувати статичну характеристику досліджуваного елемента.
10. Побудувати математичний опис дії досліджуваного елемента.
11. Змодельовати дію досліджуваного елемента для заданих вхідних сигналів та порівняти отримані результати з експериментальними характеристиками елемента.
12. Зробити висновки по результатах виконаної роботи.
13. Оформити протокол та захистити роботу.

Особливості роботи з електронними компонентами та вимірювальним обладнанням

В ході постановки експериментів і проведення досліджень необхідно вміти представляти кожен експериментальний зразок компоненту технічної системи, коректно задавати умови його роботи та виконувати вимірювання.

Для цього необхідно дотримуватись наступних рекомендацій.

1. При постановці експериментів потрібно підготувати схему проведення досліджень (рис.3.6)

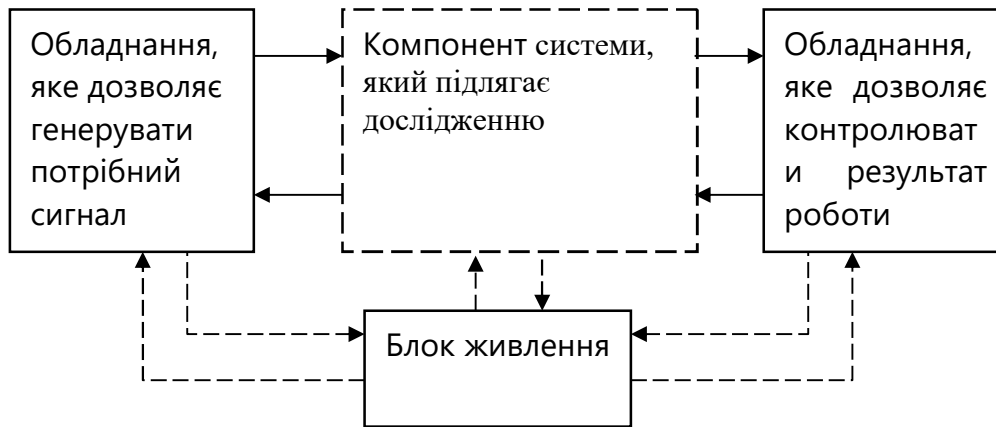


Рис. 3.6. Загальна схема для проведення досліджень компонентів фізично різномірних систем

2. Вибрати обладнання для задання та вимірювання сигналів, а також джерело живлення. У якості обладнання для генерування сигналу керування, може бути використано генератор сигналів або ручне задання сигналу.

Обладнанням для контролю результатів роботи можуть бути вимірювальні прилади: - амперметр, вольтметр, омметр, або комбінований пристрій - тестер, а також осцилограф.

3. Скласти стенд, провести дослідження та обробити результати.

Для прикладу наведено схему для дослідження інформаційного елемента, який дозволяє отримувати інформацію про механічне переміщення. При цьому, результатом роботи елемента є електрична напруга на його виході (рис.3.7.)

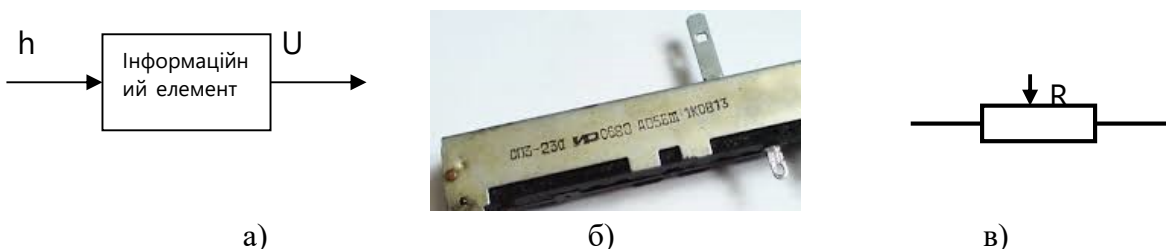


Рис. 3.7. Загальне представлення компоненту (а), його зовнішній вигляд (б), позначення на принципових схемах (в)

Принципова схема для дослідження інформаційного елементу (рис.3.8), таблиця для фіксації результатів і графік (рис.3.9 а) ілюструють етапи виконання роботи.

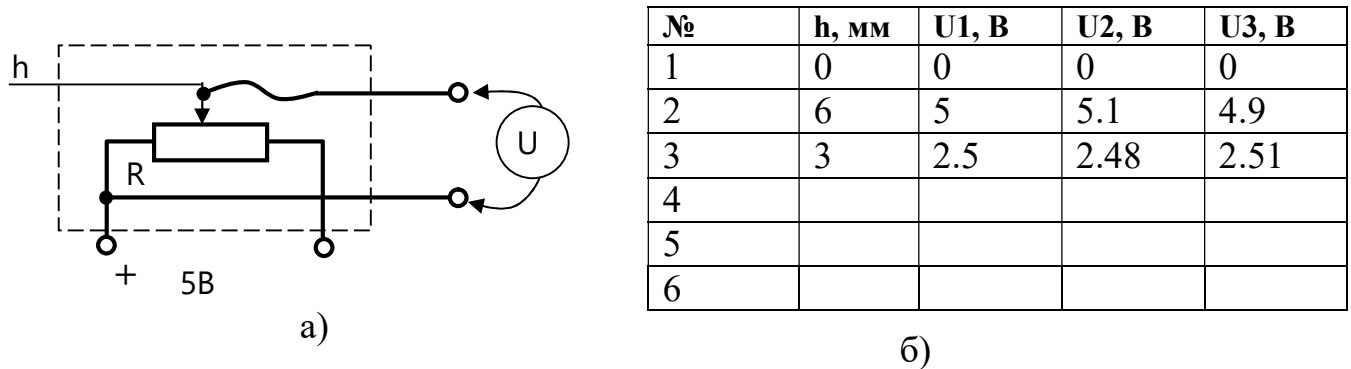


Рис.3.8. Етапи виконання лабораторної роботи (а- принципова схема стану для дослідження, б – таблиця з результатами вимірювань, графічне представлення результат

4.Наступним етапом виконання цієї роботи є побудова математичного опису для імітації дії інформаційного елементу в комп'ютерному середовищі (рис.3.9).

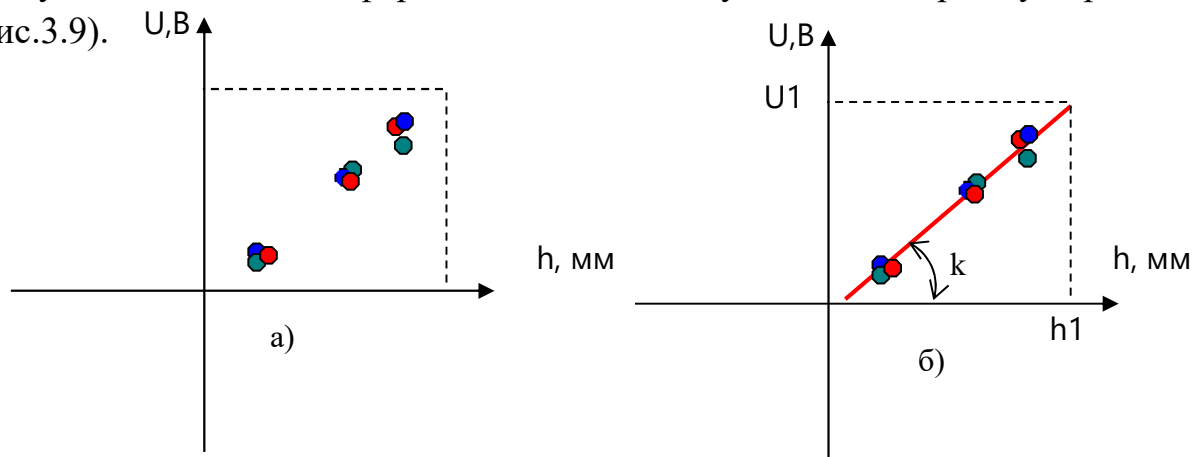


Рис.3.9. Етапи виконання лабораторної роботи (а- графічне представлення результатів вимірів, б – апроксимація експериментальних результатів за допомогою математичної залежності

Математичний опис будують за шляхом апроксимації експериментальних даних за допомогою вибору відповідної математичної залежності. Вона має відповідати набору експериментальних з адаптованими до них коефіцієнтами. Наприклад, для експериментальних результатів визначення статичної характеристики розглянутого вище інформаційного елементу такою є лінійна залежність

$$U = k \cdot h,$$

де k – коефіцієнт лінійності, який визначається за формулою

$$k = \text{tg}(U_1 / h_1).$$

Аналогічним чином мають виконуватись експериментальні роботи, наведені в лабораторних завданнях.

4. ЗАВДАННЯ ДЛЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

4.1. Лабораторна робота №4.1. Візуалізація математичних функцій

Мета: Отримання навичок автоматизації обчислювальних операцій та побудови графіків.

Основні відомості для виконання операцій у пакеті MatLAB.

Математичні операції пакету.

Корінь квадратний: **sqrt(x)**.

Модуль: **abs(x)**.

Додавання, віднімання, множення: +, -, *.

Ділення зліва на право: **./**

Ділення справа вліво: **.**

Взведення у ступінь: **.^**

Завдання циклу: **for** z=0 : 0.05 : 1.5 <оператори> **end**;

де 0 – початкове значення, 0.05 – шаг приращення, 1.5- кінцеве значення.

Вивід графіку: **plot(x,y,x1,y1,x2,y2)**;

де x,y – координати точки кривої 1, x1,y1- координати точки кривої 2, x2,y2- координати точки кривої 3.

Вивід надпису: **title('Графік функцій.....')**;

Збереження значень індексованої змінної:

b(1)=20; b(2)=43; и т.д., де b – індексована змінна, 1, 2... - індекси; 20, 43 – значення.

Завдання на лабораторну роботу. Написати програму для вирішення системи рівнянь графічним способом.

Завдання

№ за списком	Система рівнянь
1.	$Y_1 = \sqrt{(25 - x^2)}; Y_2 = -3/4 \cdot x.$
2.	$Y_1 = \sqrt{(x + 4)}; Y_2 = 2 \cdot \sqrt{(6 - x)}.$
3.	$Y_1 = 7 \cdot \sqrt{(5 + 4x - x^2)}; Y_2 = 14 - 2 \cdot x.$
4.	$Y_1 = x^2; Y_2 = 7 \cdot x - 10.$

5.	$Y_1 = x^2; Y_2 = 3 \cdot x + 4.$
6.	$Y_1 = x^2; Y_2 = 7 \cdot x - 12.$
7.	$Y_1 = x^2; Y_2 = 2 \cdot x + 5 \cdot x + 7.$
8.	$Y_1 = \sqrt{(x-2)}; Y_2 = x - 3.$
9.	$Y_1 = \sqrt{(4-x^2)}; Y_2 = -1/x.$
10.	$Y_1 = \sqrt{(x^2 - 3 \cdot x - 10)}; Y_2 = 8 - x.$
11.	$Y_1 = x^3; Y_2 = 3 \cdot x + 4.$
12.	$Y_1 = x^2; Y_2 = -1/x.$

Порядок захисту. Робота захищається за наявності програми, що працює шляхом відповідей на контрольні питання.

Додаткові відомості. Для налагодження програми іноді можуть контролюватись проміжні результати виконання окремих операторів. Для виведення таких результатів потрібно в програмі після оператора прибрати символ «;».

4.2 Лабораторна робота №4.2 Розгалужена будова моделі

Мета: Отримання навичок розробки розгалужених програм.

Основні відомості. Програма може складатися з кількох М-файлів, які можуть взаємодіяти один з одним у процесі обчислень. Зазвичай один із М-файлів є головним (іноді його називають файлом-сценарієм) який керує ходом обчислень. Головний файл в своєму тілі може використовувати інші М-файли, які є зовнішні по відношенню до нього. Такі М-файли, як правило містять в собі частини програми, які можуть бути використані багатократно з різними вхідними параметрами. Називають такі М-файли М-функціями. Звернення з головного файлу до М-файлів функцій може виконуватись за

допомогою їх імен. У цьому випадку М-файл, що представляє собою функцію, повинен мати ряд формальних ознак. Такими ознаками є:

-
- ключове слово – **function**;
- символ, що позначає змінну яка повертається;
- та ім'я функції зі списком значень.

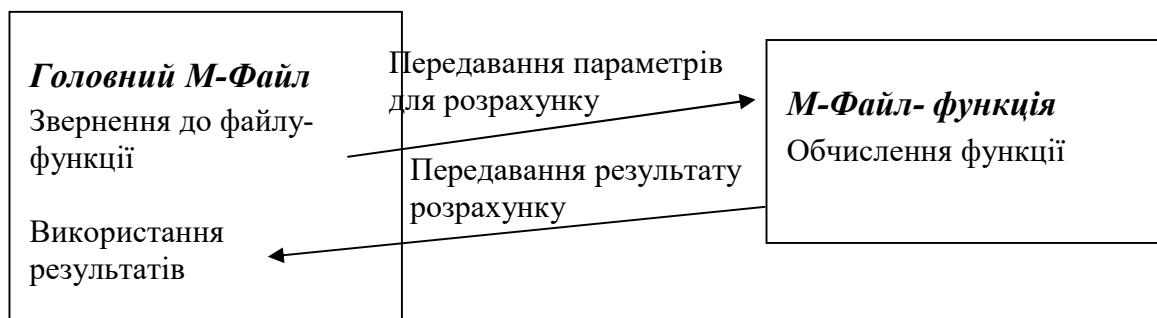
Звернення до М-файлу функції з головного файлу виконується шляхом запису імені функції зі списком змінних, що потрібні для її виконання.

Приклад запису М-файлу функції:

-----Приклад функції-----
function y=F1(x,d) % “x” та “d” є параметрами функції, “y” – змінна, яка буде
y=x+d; % отримувати результат виконання функції
%Кінець

Можлива структура програми:

-----Приклад структури програми-----



*В тексті програми після символу «%» надається коментар, який обробником програм не виконується.

Важливо: При збереженні функції в файлі, його ім'я має бути таким самим, як і ім'я функції, але з розширенням «.m».

Приклад. Для функції «**function** y=F1(x,d)» фал треба зберігати з ім'ям **F1.m**

Приклад звернення до файлу-функці F1 з головного файлу

Головний М-Файл

```
x=2;  
d=16;  
k=21;  
y = F1(x,d); %Так виглядає звернення до функції  
%Нижче може бути записана формула, яка використовує результат  
виконання функції  
z=y+k;
```

При цьому, треба розуміти, що в файлах функціях всі імена змінних, що знаходяться всередині М-файлу, а також змінні, визначені в заголовку є локальними і їх дія не розповсюджується на головний файл.

Всі змінні, які використовуються у головному файлі-сценарії створюють деяку робочу область, в якій їх значення зберігаються протягом всього сеансу роботи з системою. Якщо в програмі одночасно використовуються кілька файлів-сценаріїв, то її робоча область є єдиною для всіх файлів-сценаріїв, що викликаються під час роботи з системою.

Завдання на лабораторну роботу. Написати програму для вирішення системи рівнянь графічним способом з використанням М-файлів. При цьому програма повинна складатися з трьох частин: - М-файла головної програми; - М-файлу функції обчислення значень для першого графіку; - М-файлу функції обчислення значень для другого графіку. Система рівнянь береться з попередньої лабораторної роботи.

Порядок захисту. Робота захищається при наявності програми, що працює шляхом відповідей на контрольні питання.

4.3. Лабораторна робота №4.3. Моделювання і дослідження поведінки математичної функції

Мета: Отримання навичок математичних моделей з використанням діалогового режиму та графічної інтерпретації результатів.

Основні відомості. Засоби вводу та виводу інформації пакету **MatLAB**. Вивід тексту на екран: **disp('текст')**

Формування текстової строки яка вміщує числове значення:
sprintf('параметр %g', x)

Запроси на ввід числового значення з клавіатури: **x=input('x=')**

Організація діалогу з користувачем за допомогою меню:
z=menu('Заголовок','Кнопка_1','Кнопка_2'.....)

Завдання паузи: **pause**

Очищення поля виводу у вікні: **delete(gca)**

Перетворення числового значення у строкову змінну: **y=num2str(x)**, де x – числове значення, y – строкова змінна.

Формування полів виводу графічної та текстової інформації у вікні виводу: **subplot(m,n,p)**, де m – число полів по вертикалі, n- число полів по горизонталі, p- номер підвікон, що призначені для виводу інформації.

Приклад 1. **subplot(2,3,6)** – вікно буде розділено на 6- полів, по три поля на кожному з двох рівней. Вивод інформації буде відбуватися у праве нижнє поле.

Приклад 2. **subplot(2,3,[4,5,6])** – Вивод інформації буде відбуватися у три поля, що розташовані знизу.

Виведення тексту у поле вікна: **h=text(x,y,'Початкові данні:', 'FontSize',12)** , де x,y –координати для виводу тексту, 'FontSize',12 – визначає розмір шрифту у пікселях.

Очищення вікна від зображення координатних осей Очистка вікна від зображення та надписів на них: **axis('off')**

Завдання на лабораторну роботу. Написати програму для побудови та виводу на екран результатів у графічному вигляді для заданої функції (відповідно до номеру бригади). При цьому необхідно забезпечити можливість завдання параметрів функції у діалоговому режимі та вивід в перше поле вікна - графіка функції при початкових значеннях параметрів, у друге поле вікна – графіку функції для нових значень параметрів, що задаються користувачем, а у третє поле – текстової інформації, про тип та параметри заданої функції.

Завдання

№ за списком	Вид функції
1.	$Y = a + b \cdot x^m$.
2.	$Y = a + (1/b) \cdot x^m$.
3.	$Y = a + (b/c) \cdot x^m$.

4.	$Y = a + b / x^m.$
5.	$Y = (a + b \cdot x^m) \cdot c.$
6.	$Y = a + b \cdot \sin(F + \omega \cdot t).$
7.	$Y = a / x + b.$
8.	$Y = a \cdot (1 - \sin(\omega \cdot t)).$
9.	$Y = b - c \cdot \sin(F + \omega \cdot t).$
10.	$Y = a + (1/b) \cdot x^m.$
11.	$Y = (a + b \cdot x^m) \cdot c.$
12.	$Y = A \cdot x^n - c.$

Порядок захисту. Робота захищається за наявності програми, що працює шляхом відповідей на контрольні питання.

Рекомендації. Для написання програми необхідно використовувати операції з циклами та М-файли.

4.4. Лабораторна робота №4.4. Прогнозна модель дії об'єкту

Мета: Отримання навичок побудови математичних моделей на основі експериментальних даних з використанням діалогового режиму та графічної інтерпретації результатів.

Основні відомості. Математична модель на основі експериментальних даних може бути побудована шляхом їх апроксимації за допомогою поліномів виду: $y(x) = A_3 \cdot x^3 + A_2 \cdot x^2 + A_1 \cdot x + A_0$, де $A_3..A_0$ - коефіцієнти поліному.

Спосіб завдання вектору у середовищі MatLab: $v = [1.5 \ -2.4 \ 4.9e3]$;

Функція апроксимації даних, що були отримані експериментальним шляхом: $\text{polyfit}(X, Y, n)$,

де X - вектор аргументів, Y – вектор експериментальних значень, n - ступінь апроксимуючого поліному.

Функція для обчислення значень поліному за заданими значеннями його аргументів: `polyval(P,x)`,

де P – заданий вектор коефіцієнтів поліному, x - значення аргументу.

Завдання вектору: `x=0 : 0.05 : 1.5`;

де 0 – початкове значення, 0.05 – шаг прирощення, 1.5 - кінцеве значення.

Вивід графіку: `plot(x,y,x1,y1,x2,y2)`;

де x,y – координати точки кривої 1, $x1,y1$ - координати точки кривої 2, $x2,y2$ - координати точки кривої 3.

Вивід надпису: `title('Графік апроксимації результату експериментів')`;

Завдання на лабораторну роботу. Написати програму для апроксимації] завданих експериментальних даних за допомогою поліномів 1-го, 2-го та 3-го ступеню. Забезпечити можливість вводу даних у діалоговому режимі. Вивести результати у вигляді графіків, порівняти їх та зробити висновки.

Завдання

№ за списком	X	1	2	3	4	5	6	7	8
1	Y	-1.1	0.2	0.4	0.8	0.7	0.6	0.4	0.1
2	Y	-3,1	0.8	0.9	0.7	0.3	0.1	0.15	0.01
3	Y	-1.1	0.2	0.4	0.5	0.8	0.3	0.25	0.2
4	Y	2.1	1.2	3.6	6.1	5.3	4.0	3.05	2.0
5	Y	-0.5	1.0	1.8	2.3	1,5	1.0	0.8	0.2
6	Y	-0.7	0.1	0.7	0.75	0.6	0.4	0.25	0.1
7	Y	-0.9	0.2	0.3	0.9	0.8	0.7	0.3	0.1
8	Y	-0.5	0.1	0.7	0.8	0.9	0.3	0.4	0.3
9	Y	-1.6	1.1	3.4	5.1	5.0	3,8	2.1	2.5
10	Y	-1.3	1.8	2.1	3.4	4.2	2.1	1.1	0.8

Порядок захисту. Робота захищається за наявності програми, що працює шляхом відповідей на контрольні питання.

4.5. Лабораторна робота 4.5. Моделювання динамічного процесу руху тіла

Постановка задачі. На рисунку (рис.1) представлена схема об'єкту моделювання. На тверде тіло, що лежить на плоскій поверхні, діє сила F . Результатом є рух тіла.

Математичний опис процесу дії об'єкту

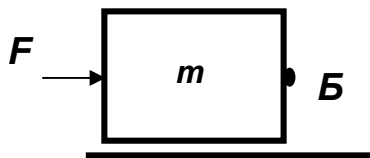


Рис.1.Схема об'єкту моделювання

Завдання на лабораторну роботу. Скласти математичний опис процесу руху тіла під дією зовнішньої сили. При цьому треба врахувати силу спротиву від в'язкого тертя тіла по поверхні руху (прийняти значення коефіцієнту в'язкого тертя $b=0.03$). Перетворити математичний опис в модель в пакеті Simulink. Завдання 1. Змоделювати процес руху твердого тіла при різних законах зміни зовнішньої сили в часі (спостерігати за змінами в часі прискорення, швидкості та переміщення т.Б, яка розташована на твердому тілі). Завдання 2. Визначити вплив величини маси тіла на час його руху при імпульсній зміні зовнішньої сили F . Результати представити у вигляді графіку залежності $t=f(m)$, де t – час руху тіла, m -- маса.

Графік, який ілюструє отримані результати

Завдання

№ за списком	Значення сили F, N	Час дії сили, с (імпульс)
1.	0.5	0.1
2.	3.5	0.05
3.	0.2.5	0.1
4.	6	0.05
5.	3	0.1
6.	4	0.05
7.	35	0.1
8.	30	0.05
9.	120	0.1
10.	63	0.05

11.	10	0.1
12.	45	0.05

План виконання роботи. 1.Скласти математичний опис процесу руху тіла під дією зовнішньої сили. 2.Побудувати математичну модель. 3.Задати значення параметрів (m , b , $F(t)$). 4. Виконати завдання I. Змодельовати процес для різних законах зміни F від часу. 5. Провести аналіз отриманих результатів. 6. Виконати завдання II. Намалювати систему координат для побудови графіку $t=f(m)$, До проведення експериментів на цій системі координат намалювати графік вказаної залежності відповідно до вашого розуміння. Після цього провести послідовно серію експериментів для різних значень мас тіла та однакових імпульсах сили F . В кожному експерименті визначати час руху маси та заносити отримані значення в таблицю. На основі отриманих даних побудувати графік $t=f(m)$. Проаналізувати результат, порівняти його зі своїм попереднім розумінням та зробити висновки. **Порядок захисту.** Робота захищається за наявності працюючої програми та оформленого протоколу шляхом відповідей на контрольні запитання.

Висновки.

4.6. Лабораторна робота 4.6. Моделювання процесів у пружній системі

Потрібні відомості. На рисунку (рис.1.) представлено схему об'єкту моделювання. На пружину у т. А діє сигнал X . Маса під дією сили, що виникає у пружині за рахунок її стиснення починає рухатись. Швидкість та положення точки Б, що розташована на масі, залежить від сигналу X , жорсткості пружини C та величини маси m .

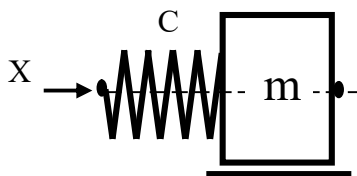


Рис.1. Схема об'єкту моделювання

Математичний опис роботи об'єкту

Завдання на лабораторну роботу. Скласти математичний опис процесу роботи об'єкту. Перетворити математичний опис у модель. Змодельовати роботу об'єкту для двох різних типів вхідних сигналів.

1. Для ступінчатої зміни положення т. А ($x=x_0$, де x - переміщення, x_0 - значення переміщення).

2. Для вхідного сигналу виду: $x=x_0*\sin(\omega*t)$, де x - переміщення, x_0 - максимальна амплітуда переміщення, ω - частота, t - час.

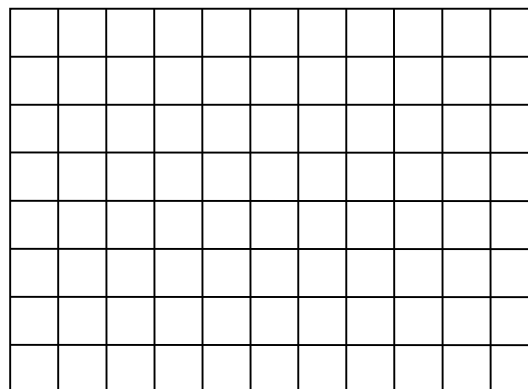
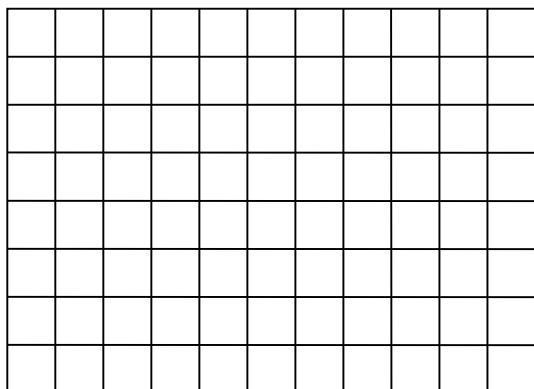
Результати представити у формі графіків: графік залежності швидкості т. Б від часу для при ступінчатого переміщення т. А на завдану величину; графік залежності максимальної швидкості руху т. Б, для різних частот коливань т. А.

Завдання

№ за списком	1	2	3	4	5	6	7	8	9	10	11	12
m, кг	5	3.5	2..5	0.6	0.3	4.6	0.5	0.7	10	1.5	7.0	0.8

План виконання роботи. 1.Скласти для об'єкту опис процесів, що вивчаються. 2. Провести моделювання процесу. 3. Результати роботи оформити у вигляді протоколу.

Результати моделювання



Порядок захисту. Робота захищається за наявності протоколу. Протокол повинен вміщувати розрахункову схему об'єкту моделювання, опис моделі, значення параметрів для яких проведено дослідження, графіки залежності: графік залежності швидкості т. Б від часу для ступінчатого зміщення т. А; графік залежності максимальної швидкості руху т. Б, для різних частот коливань т. А.

Додаткові відомості. Значення параметрів, що не вказані, обираються самостійно.

4.7. Лабораторна робота №4.7. Моделювання процесів в гідравлічних компонентах систем.

Завдання на лабораторну роботу

Виконав:

Об'єкт моделювання. На рисунку (рис.1) представлена схема об'єкту моделювання. Ззовні камери існує тиск рідини p більший ніж в її середині. Відбувається процес додаткового наповнення камери рідиною. Результатом процесу є збільшення тиску в середині камери.

Математичний опис процесу*

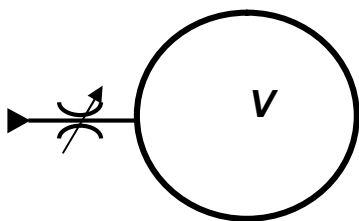


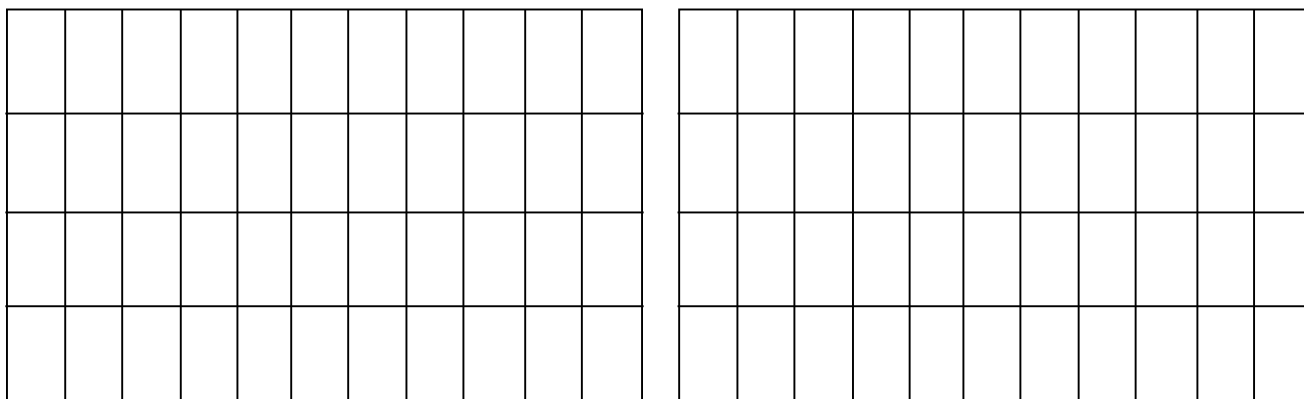
Рис.1. Схема об'єкту моделювання

Завдання. Розробити математичний опис процесу зміни тиску в середині камери під дією зовнішнього тиску перетворити математичний опис в математичну модель в програмі MatLab. Змоделювати процес при різних параметрах об'єкту. Визначити вплив параметрів об'єкту та зовнішнього середовища на час зміни тиску в середині камери при постійному тиску ззовні. Результати представити у формі графіків залежностей 1: $p=f(t)$; 2: $t_1=f(V)$; 3: $t_1=f(d)$; де p – тиск в камері, t – час процесу, V – об'єм, d – діаметр отвору, t_1 – час зміни тиску в камері.

Індивідуальне завдання.

№	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
p, М Па	2.0	5.0	10.0	12.0	15.0	17.0	20.0	25.0	30.0	35.0	40.0	45. 0

Графіки, що ілюструють отримані результати



План виконання роботи. 1. Розробити математичний опис процесу та трансформувати опис в модель. 2. Намалювати графік зміни тиску в камері з часом, за своїм розумінням, до проведення експериментів. 3. Задати значення параметрів. 4. Провести моделювання і за результатами побудувати графіки. 5. Провести аналіз отриманих результатів та зробити висновки. Порівняти отриманий графік процесу з графіком, який був намальований до проведення експериментів.

Порядок захисту роботи. Робота захищається при наявності робочої версії моделі та оформленого протоколу шляхом обґрунтованих відповідей на контрольні запитання.

Додаткові пояснення

*для математичного опису процесу наповнення камери використовувати математичні залежності, які визначають витрату через дросель

$q = \mu \cdot f \cdot \sqrt{2 \cdot \rho \cdot (p_2 - p_1)}$, де μ – коефіцієнт витрати (прийняти 0.6), f – площа поперечного перерізу дроселя (треба задавати), ρ – щільність мінеральної олії, p_1, p_2 – тиск на вході в дросель і тиск на виході дроселя відповідно.

Тиск в камері при подаванні до неї витрати визначається за залежністю

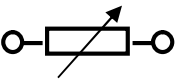
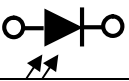
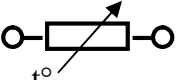
$$\frac{dp}{dt} = \frac{E}{V} q, \text{ де}$$

p – тиск в камері, E – адіабатичний модуль пружності робочої рідини, V – об'єм камери, q – витрата, яка потрапляє в камеру.

5. ЗАВДАННЯ ДЛЯ ФІЗИЧНОГО МОДЕЛЮВАННЯ

5.1. Лабораторна робота 5.1. Дослідження та математичний опис статичних характеристик інформаційних елементів

Тема. Інформаційні елементи фізично різнорідних систем.

Тип елемента	Схема принципу дії елемента	Математичний опис дії елемента
Механічне переміщення-електрика		
Світло-електрика		
Температура-електрика		

Завдання та послідовність виконання роботи

1. Розробити схему визначення статичних характеристик для заданого типу інформаційного елемента.
2. Визначити, яка змінна визначає вхідну дію, та яка змінна визначає вихідний результат.
3. Розробити методику проведення дослідження статичної характеристики
4. Розробити форму таблиці для запису результатів.
5. Зібрати схему на стенді.
6. Провести дослідження, фіксуючи результат в таблиці.
7. Побудувати статичну характеристику інформаційного елемента
8. Побудувати математичний опис статичної характеристик.
9. Сформулювати висновок.

Зміст протоколу. Протокол має містити:

прізвище та назву учбової групи студента, який її виконав, назви вказаних вище пунктів та результати їх виконання.

5.2 Лабораторна робота 5.2. Дослідження та математичний опис статичних характеристик підсилюючих елементів

Тема. Підсилюючі елементи фізично різномірних систем.

Загальні відомості. Існують елементи, які дозволяють підсилювати вхідний сигнал за потужністю (рис.1). При цьому зміни вихідного сигналу при зміні вхідного сигналу залежать від характеристик підсилюючого елементу. Для технічних систем, в яких цей елемент використовується характеристики елементу суттєво впливають на характеристики всієї системи. Тому потрібно знати їх статичні характеристики. У якості прикладу наведено принципову схему релейного підсилювача (рис.2).



Рис.1. Загальна схема підсилюючого елементу

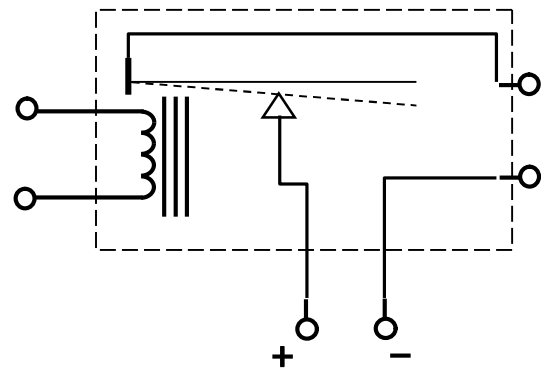


Рис.2. Принципова схема підсилювача релейного типу

Завдання та послідовність виконання роботи

1. Розробити схему визначення статичних характеристик для заданого типу підсилюючого елементу.
2. Визначити, яка змінна визначає вхідну дію, та яка змінна визначає вихідний результат.
3. Розробити методичку проведення дослідження статичної характеристики
4. Розробити форму таблиці для запису результатів.
5. Зібрати схему на стенді.
6. Провести дослідження, фіксуючи результат в таблиці.
7. Побудувати статичну характеристику підсилюючого модулю
8. Побудувати математичний опис статичної характеристик.
9. Сформулювати висновок.

Зміст протоколу. Протокол має містити:

прізвище та назву учбової групи студента, який її виконав, назви вказаних вище пунктів та результати їх виконання.

5.3. Лабораторна робота 5.3. Розробка, перевірка та математичне представлення модулів

Загальні відомості. Джерело електричної енергії 220 В змінного струму є в кожному приміщенні, але автоматичні прилади для свого живлення часто потребують іншої напруги та струму. У якості узгоджувачих елементів застосовують трансформатори та випрямлювачі струму. Трансформатори будують на використанні ефекту утворення електромагнітного поля навколо провідників по яким тече змінний струм та ефекту виникнення електричного струму у провіднику у разі, якщо він знаходиться в межах дії електромагнітного поля (електромагнітна індукція). Коефіцієнт трансформації визначається як співвідношення кількості витків провідника у вихідній та вхідній обмотках. Для перетворення змінного струму в постійний (випрямлення) застосовують напівпровідникові діоди. Для стабілізації напруги використовують конденсатори (рис.1).

Завдання 1. 1. Розробити принципову схему узгоджувачого елемента для

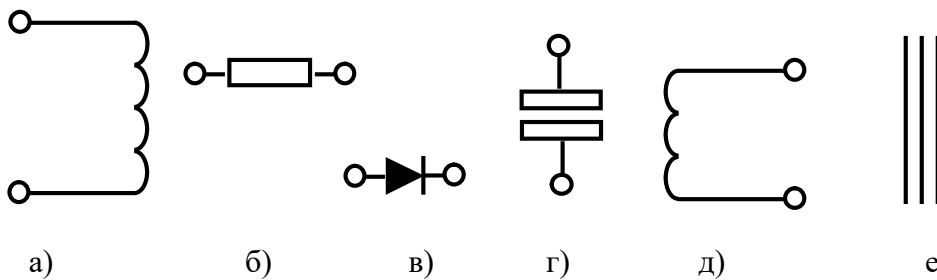


Рис.1. Позначення елементів автоматичних систем. а – вхідна обмотка трансформатора, б – опір, в- діод, г- конденсатор, д – вихідна обмотка трансформатора, е – провідник електромагнітного поля (серцевина трансформатора)

узгодження напруги джерела (24В 50 Гц) з напругою, яка потрібна споживачеві (12В). При цьому необхідно забезпечити його безпечну роботу.

2. Провести розрахунки і визначити параметри елементів.

3. Перевірити роботу спроможність узгоджувачого елемента на стенді.

Завдання 2. Розробити принципову схему елемента узгодження з випрямленням вихідного струму. Розрахувати її та перевірити.

Завдання 3. Розробити принципову схему елемента узгодження з випрямленням вихідного струму та стабілізацією напруги. Розрахувати її та перевірити.

Вимоги до системи. 1. Передбачити можливість включення і виключення системи у роботу в ручному режимі.

Хід виконання роботи. 1. Вибрати потрібні елементи системи.

2. Розробити принципову схему системи.

3. Перевірити логіку роботи системи.

4. Скласти макет системи. Живлення **НЕ** вмикати!

5. Запросити викладача.

6. Перевірити роботу спроможність системи.

7. Скласти протокол лабораторної роботи.*

- Протокол має містити принципову схему системи з описом її роботи і висновки за результатами практичної перевірки роботи спроможності системи.

5.4. Лабораторна робота 5.4. Дослідження та математичний опис статичних характеристик обчислювальних елементів

Тема. Обчислювальні елементи мехатронних систем.

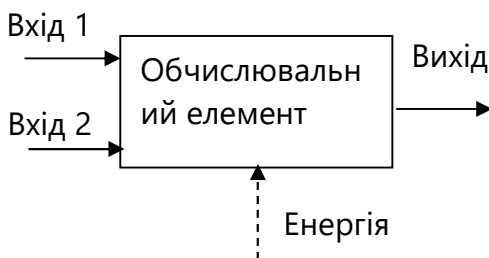


Рис.1. Загальна схема обчислювального елемента

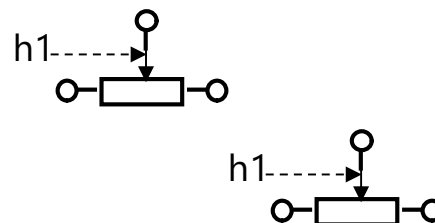


Рис.2. Заготовка до схеми обчислювального елемента на електричних потенціометрах

Завдання та послідовність виконання роботи

1. Розробити схему обчислювального елемента для алгебраїчного додавання механічних сигналів, використовуючи електричні потенціометри (рис.1,2).

2. Розробити схему визначення статичних характеристик для заданого типу обчислювального елемента.
3. Визначити, яка змінна визначає вхідну дію, та яка змінна визначає вихідний результат.
4. Розробити методика проведення дослідження статичної характеристики
5. Розробити форму таблиці для запису результатів.
6. Зібрати схему на стенді.
7. Провести дослідження, фіксуючи результат в таблиці.
8. Побудувати статичну характеристику, що ілюструє роботу обчислювального елемента.
9. Побудувати математичний опис статичної характеристик.
10. Сформулювати висновок.

Зміст протоколу. Протокол має містити: прізвище та назву учбової групи студента, який її виконав, назви вказаних вище пунктів та результати їх виконання.

4.5. Лабораторна робота 5.5. Дослідження та математичний опис статичних характеристик виконавчих елементів

Тема. Виконавчі елементи фізично різномірних систем.

Загальні відомості. Існують елементи систем, які дозволяють виконувати команди керування, наприклад рухати робочі органи технічних об'єктів. Одним з таких елементів є електричний двигун (рис.1,2). На вхід двигуна подається електричного напруга. При цьому вал двигуна обертається. Якщо вал двигуна поєднати з робочим органом, то він буде приводити його в рух і виконувати механічну роботу. Зміна вхідної напруги призводить до зміни частоти обертання валу двигуна.

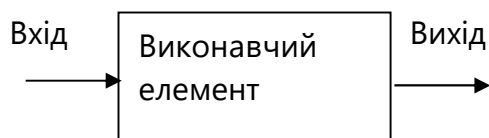


Рис.1. Загальна схема виконавчого елемента

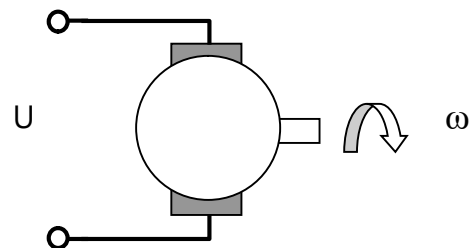


Рис.2. Принципова позначення електричного двигуна постійного струму

Завдання та послідовність виконання роботи

1. Розробити схему визначення статичних характеристик для заданого типу виконавчого елемента.
2. Визначити, яка змінна визначає вхідну дію, та яка змінна визначає вихідний результат.
3. Розробити методику проведення дослідження статичної характеристики
4. Розробити форму таблиці для запису результатів.
5. Зібрати схему на стенді.
6. Провести дослідження, фіксуючи результат в таблиці.
7. Побудувати статичну характеристику виконавчого елемента.
8. Побудувати математичний опис статичної характеристик.
9. Сформулювати висновок.

Зміст протоколу. Протокол має містити:

прізвище та назву учбової групи студента, який її виконав, назви вказаних вище пунктів та результати їх виконання.

5.6. Лабораторна робота 5.6. Розроблення, експериментальне дослідження та математичне представлення модулю підсилення

Загальні відомості. Зазвичай потужність сигналів керування значно менше потужності, яка потрібна для виконання технологічних операцій. Для узгодження потужностей використовують підсилювачі (електричні, пневматичні, гідравлічні та ін.). Електричні підсилювачі можуть створюватись на основі напівпровідникових пристроїв – транзисторів (рис.1,2). Транзистор

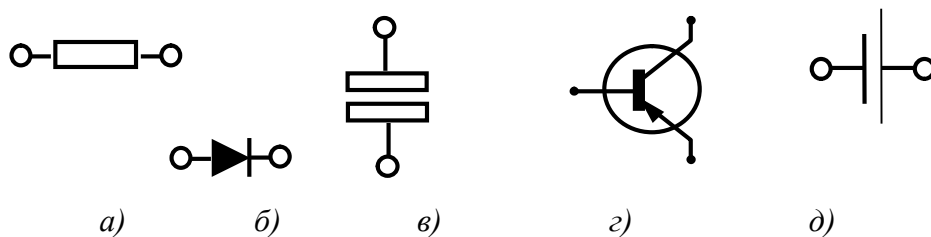


Рис.1. Позначення елементів автоматичних систем. а – електричний опір, б – діод, в- конденсатор, г – транзистор зі структурою р-п-р, д – джерело постійного струму

містить два напівпровідникові **р-п** переходи: емітерний та колекторний. Електричний опір переходу емітер-колектор залежить від потенціалу бази. Якщо база відключена, то опір переходу емітер-колектор є нескінченним.

Завдання 1. 1. Розробити принципову схему транзисторного підсилювача з використанням джерела постійного струму 12В.

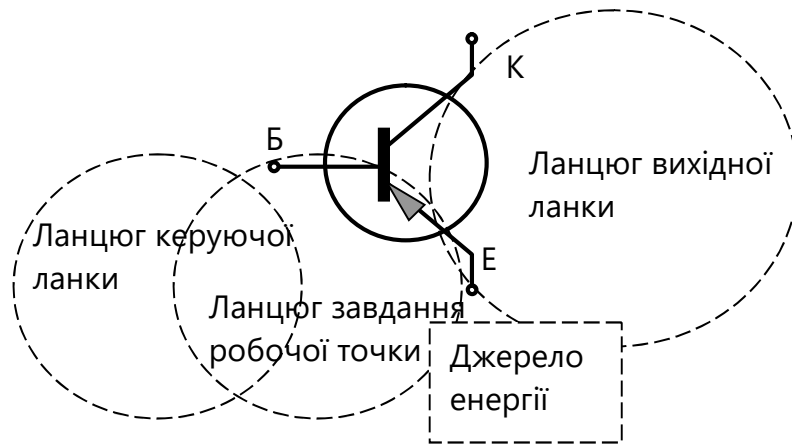


Рис.2. Схема циркуляції струму в транзисторі для забезпечення підсилюючої функції

При цьому необхідно забезпечити можливість вимірювань струму та напруги у керуючій та вихідній ланках для перевірки його функціонування.

2. Провести розрахунки і визначити параметри елементів.
3. Перевірити роботу спроможність підсилювача на стенді.
4. Зробити висновки.

Вимоги до системи. 1. Передбачити можливість налаштування величини керуючого сигналу.

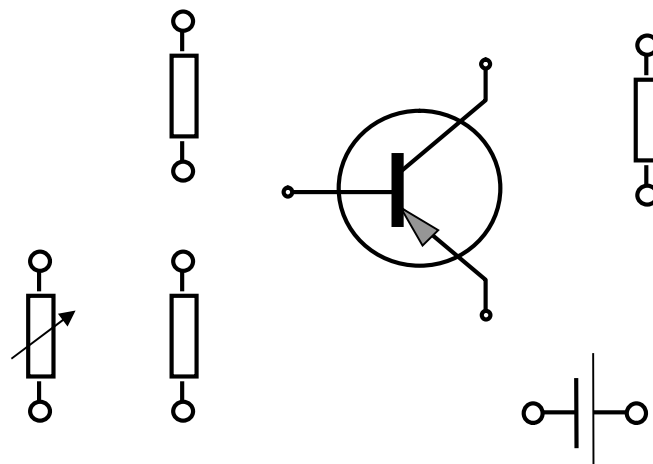


Рис.3. Схема транзисторного підсилювача та схема вимірювань для підтвердження виконання функції підсилення

Хід виконання роботи.

1. Вибрати потрібні елементи системи.
2. Розробити принципову схему системи (рис.3).

3. Перевірити логіку роботи системи.
4. Скласти макет системи. Живлення **НЕ вмикати!** Запросити викладача.
5. Перевірити роботу спроможність системи.
6. Розробити математичне представлення модулю підсилення.
6. Скласти протокол лабораторної роботи.*

* Протокол має містити принципову схему системи з описом її роботи і висновки за результатами практичної перевірки роботоспроможності системи.

5.7 Лабораторна робота 5.7. Розробка, дослідження та представлення алгоритму дії системи керування електричним приводом

Тема. Розробка системи дискретного керування електричним двигуном.

Загальні відомості. Існують системи ручного керування які забезпечують обертовий рух робочого органу з заданою частотою обертання. Наприклад система керування повітряним вентилятором аеродинамічної труби. При цьому необхідно забезпечувати рух повітря як в прямому так і в зворотному напрямку. Частота обертання вентилятору задана. Така система складається з типових елементів (рис.1).

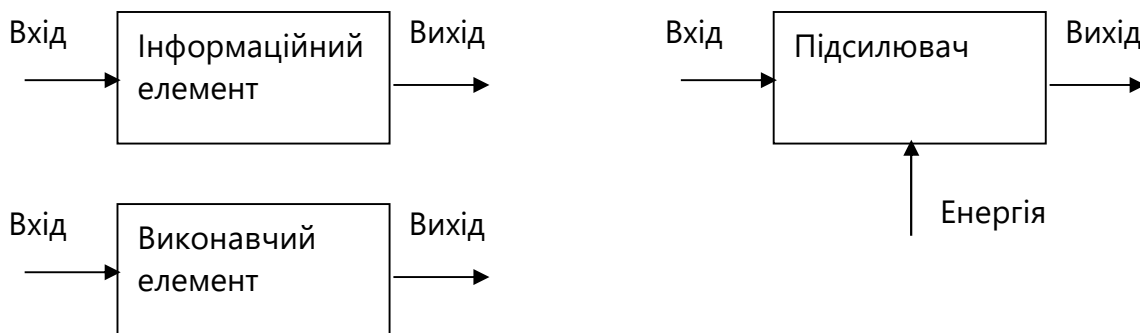


Рис. 1. набір типових елементів мехатронних систем керування

Завдання та послідовність виконання роботи

1. Розробити структурну та принципову схеми системи ручного керування обертанням повітряного вентилятора аеродинамічної труби з забезпеченням заданої постійної частоти та можливістю реверсу. Для системи використати релейний підсилювач.

2. Визначити, яка змінна визначає вхідну дію, та яка змінна визначає вихідний результат.
3. Розробити методику проведення дослідження регулюючої характеристики системи (залежності частоти обертання валу електродвигуна від керуючого сигналу).
4. Розробити форму таблиці для запису результатів.
5. Зібрати схему на стенді.
6. Провести дослідження, фіксуючи результат в таблиці.
7. Побудувати статичну характеристику виконавчого елемента.
8. Побудувати математичний опис статичної характеристик.
9. Розробити алгоритм дії фізично різномірної системи
10. Сформулювати висновок.

Зміст протоколу. Протокол має містити:

прізвище та назву учбової групи студента, який її виконав, назви вказаних вище пунктів та результати їх виконання.

5.8. Лабораторна робота 5.8. Розробка, дослідження та представлення алгоритму дії системи автоматичного дискретного керування електричним двигуном

Тема. Розробка системи автоматичного дискретного керування електричним двигуном.

Загальні відомості. В технічних об'єктах широко використовуються системи автоматичного керування. Наприклад система автоматичного керування вентилятором охолодження процесора комп'ютера. При цьому необхідно забезпечувати включення або відключення вентилятора при підвищенні температури вище заданої та його відключення при її зниженні. Частота обертання вентилятору задана. Система складається з типових елементів (рис.1).

Завдання та послідовність виконання роботи

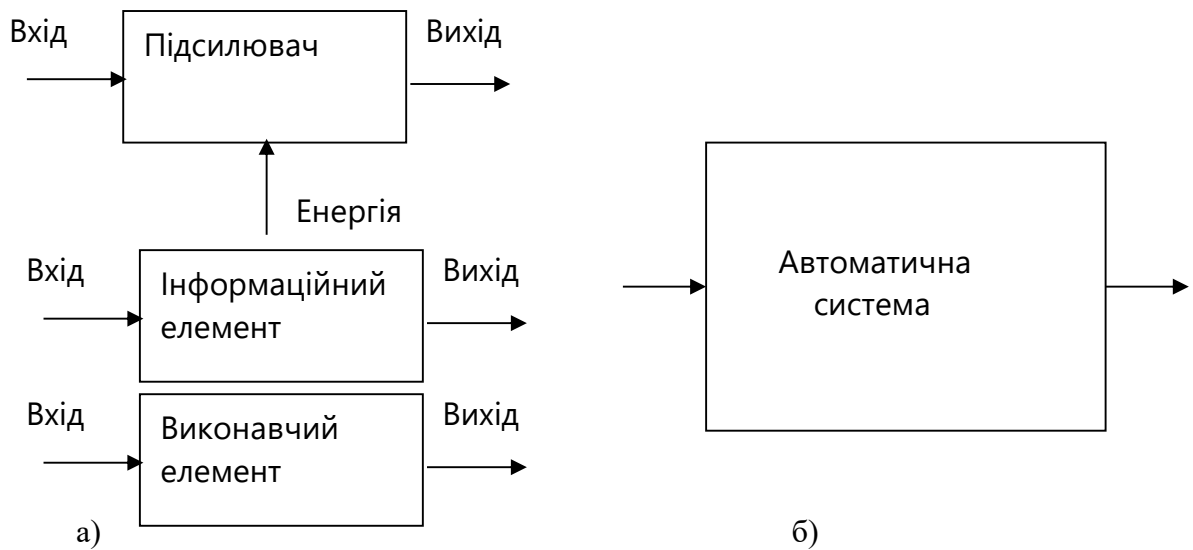


Рис. 1. Представлення типових елементів автоматичних систем (а) та автоматичної системи (б)

1. Розробити структурну та принципову схеми системи автоматичного керування вентилятором охолодження з забезпеченням заданої постійної частоти обертання.
2. Визначити, яка змінна визначає вхідну дію, та яка змінна визначає вихідний результат.
3. Розробити методику проведення дослідження регулюючої характеристики системи (залежності частоти обертання валу електродвигуна від керуючого сигналу).
4. Розробити форму таблиці для запису результатів.
5. Зібрати схему на стенді.
6. Провести дослідження, фіксуючи результат в таблиці.
7. Побудувати статичну характеристику системи.
8. Розробити алгоритм дії автоматичної системи
9. Сформулювати висновок.

Зміст протоколу. Протокол має містити:

прізвище та назву учбової групи студента, який її виконав, назви вказаних вище пунктів та результати їх виконання.

Посилання на джерела інформації

1. Фото екскаватору. URL:<https://zlochiv.net/try-porady-i-aki-dopomozhutybraty-ekskavator/> (дата звернення: 15.03.2024).
2. Схема гідросистеми екскаватору. URL: <https://patents.su/8-1745844-gidroprivod-strely-ehkskavatora.html> (дата звернення: 19.03.2024).
3. Фото літака «Антонов-Канада» 74ТК-200. URL: <https://www.epravda.com.ua/news/2021/07/26/676259/> (дата звернення: 11.03.2024).
4. Фото літака F-16. URL: <https://www.epravda.com.ua/news/2021/07/26/676259/> (дата звернення: 19.03.2024).
5. Ю.Ф. Лазарев МатЛаб 5.x - :К.:Видавнича група BNV, 2000. –384 с. (Серія «Бібліотека студента») ISBN 566-522-06-7, ISBN 5-7315-0096-7.
6. Рішення диференціальних рівнянь чисельними методами в середовищі Matlab. URL: <http://um.co.ua/6/6-7/6-7923.html> (дата звернення: 18.03.2024).

Рекомендована література

1. Яхно О.М., Узунов О.В., Луговський О.Ф., Ковалев В.А., Мовчанюк А.В., Коц І.І., Губарев О.П. Прикладна гідроаеромеханіка і механотроніка. Вінниця: ВНТУ, 2017. – 712с. id: 1683
2. Попович М.Г., Ковальчук О.В. Теорія автоматичного керування: Підручник. – 2-ге вид. перероб. І доп. – К.: Либідь, 2007. -656 с. ISBN 978-966-06-0447-6.
URL:http://pdf.lib.vntu.edu.ua/books/Popovich_2007_656.pdf (дата звернення: 18.03.2024).
3. Ю.Ф. Лазарев МатЛаб 5.x - :К.:Видавнича група BNV, 2000. – 384 с. (Серія «Бібліотека студента») ISBN 566-522-06-7, ISBN 5-7315-0096-7.
4. Назаренко І.І., Бернік І.М. Основи проектування і конструювання машин та обладнання виробництв. Навчальний посібник для вищих навчальних закладів. Видавництво «Аргар Медіа Груп». -К.: - 2013, - 544.

5. Томашевський, В.М. Моделювання систем/ В.М. Томашевський. - К. : Видавнича група ВНУ, 2005.- 352с.
6. Струтинський, В.Б. Математичне моделювання процесів та систем механіки: Підручник/В.Б. Струтинський. - Житомир: ЖШТШ, 2001.- 612с.

6. ДОДАТКИ

6.1. Приклади вирішення задач автоматизації інженерних розрахунків та моделювання

У якості прикладу наведемо програми, які було розроблено студентами в ході виконання розрахунково графічних робіт після прослуховування курсу лекцій з Основ математичного моделювання фізично різнорідних систем та виконання лабораторних робіт.

6.1.1. Розрахунок площі поперечного перерізу, об'єму, маси та вартості стандартного та нестандартного профілів сортаменту

(розробник програми Шевель Андрій, Гр.МА-02)

Призначення програми

Програма призначена для автоматизації розрахунків пов'язаних з визначенням площі поперечного перерізу, об'єму, маси та ціни заготовок нестандартного та стандартного сортаменту, а саме: труби круглого перерізу, швелера з прямими полками, рівнобокого кутика та двотавра з рівними полками..

Цільова група: студенти та працівники машинобудівних підприємств.

Користь від програми

Програма дозволяє автоматизувати, а відповідно і скоротити час для розрахунку площі поперечного перерізу, об'єму, маси та ціни заготовок нестандартного та стандартного сортаменту.

Визначення площі поперечного перерізу може бути використане у розрахунках на статичну та динамічну міцність заготовок.

Визначення об'єму та маси дозволить зорієнтуватися виробнику у визначенні розмірів упаковки. А також дозволить визначити масу, що може бути використана при транспортуванні виробів чи заготовок.

Визначення ціни заготовок дозволить зорієнтуватися користувача у економічному стані при купівлі сортаменту.

Постановка задачі

Вхідними даними в даній задачі були геометричні розміри прокатного сортаменту, а саме:

- труби круглого перерізу (внутрішній діаметр d , товщина стінки s та довжина ℓ)
- Швелера з прямими полками (висота h_1 , товщина стійки s , ширина швелера b , товщина полки t та його довжина ℓ)
- Рівнобокого кутика (висота (є одночасно і шириною) b , товщина стінки t та довжина ℓ)
- Двотавр з прямими полками (висота h_1 та товщина ніжки s , ширина полки b та товщина полки t та довжина прокату ℓ)

Отримані значення використовуються для розрахунку площі поперечного перетину, об'єму, маси та ціни цих профілів.

Самі значення розрахунків отримуємо підставивши вхідні дані у розрахункові формули, що будуть приведені далі.

Методика розрахунку

Розрахунок площі поперечного перетину профілів

1. Труба круглого перерізу

Розрахункова схема (див. рис. 1)

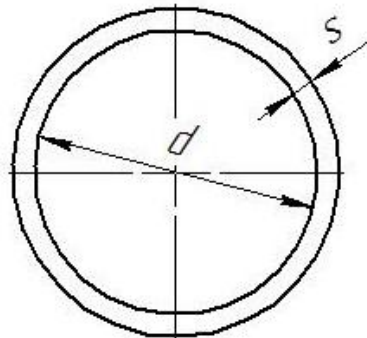


Рис.1. Розрахункова схема труби круглого перерізу

Значення площі поперечного перетину визначаємо за формулою:

$$S = \frac{\pi \cdot ((d + 2s)^2 - d^2)}{4}$$

2. Швелер з прямими полками

Розрахункова схема (див. рис. 2):

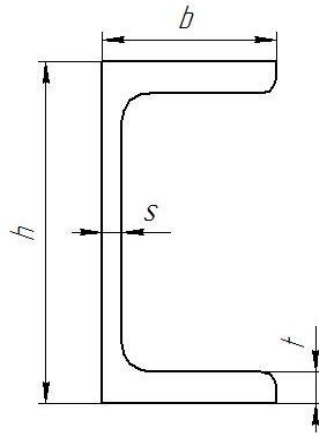


Рис.2. Розрахункова схема швелера з рівними полками

Значення площі поперечного перетину визначаємо за формулою:

$$S = (h_1 \cdot s + 2 \cdot (b - s) \cdot t)$$

3. Рівнобокий кутик

Розрахункова схема (див. рис. 3)

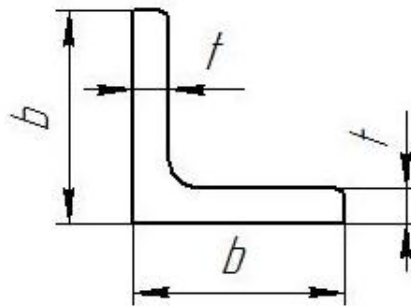


Рис.3. Розрахункова схема рівнобокого кутика

Значення площі поперечного перетину визначаємо за формулою:

$$S = (b \cdot t + (b - t) \cdot t)$$

4. Двотавр з прямими полками

Розрахункова схема (рис. 4)

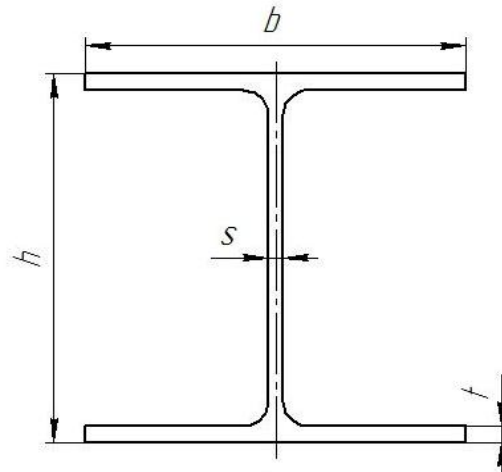


Рис .4. Розрахункова схема двотавра з прямими полками

Значення площі поперечного перетину визначаємо за формулою:

$$S = ((h1 - 2 \cdot t) \cdot s + 2 \cdot b \cdot t)$$

Розрахунок об'єму профілів

Для кожного з випадків об'єм профілів розраховуємо за формулою:

$$V = S \cdot \ell$$

де, ℓ - довжина профілю (заготовки).

Розрахунок маси

Для кожного з випадків маса профілів розраховуємо за формулою:

$$m = \rho \cdot V$$

де, ρ - густина матеріалу з якого виготовлено профіль (вибирається користувачем).

Розрахунок ціни

Для кожного з випадків ціна профілю розраховується за формулою:

$$C = p \cdot \ell$$

де, p - вартість одного погонного метру прокату (задається користувачем).

Наведемо приклад розрахунку маси швелера з прямими полками, що виготовлений із звичайної сталі:

1. Визначаємося з вибором розрахунку (вибираємо розрахунок маси заготовки).
2. Визначаємося з матеріалом з якого виготовлений профіль (вибираємо сталь звичайна)
3. Визначаємося з профілем заготовки (вибираємо швелер з прямими полками)
4. Задати геометричні розміри профілю:
 - 4.1 Задати значення ширини швелера b ;
 - 4.2 Задати значення товщини полки t ;
 - 4.3 Задати значення висоти швелера h ;
 - 4.4 Задати значення товщини стійки швелера s ;
 - 4.5 Задати значення довжини профілю ℓ ;
 - 4.6 Розрахувати значення маси профілю за формулою:

$$m = \rho \cdot (h \cdot s + 2 \cdot (b - s) \cdot t) \cdot \ell$$

5. Вивести значення маси на екран.

Для інших варіантів розрахунків і профілів схема розрахунку буде подібною.

Алгоритм розрахунку

Наведемо алгоритм розрахунку для методики представленої у пункті 4.

Алгоритм розрахунку див. рис. 5.

В наведеному алгоритмі блок вибору типу розрахунку, матеріалу та профілю є складним. Він дозволяє змінювати як введені дані так і умови та типи розрахунків.

2. Текст програми.

```
while i<3
```

```
Вибір типу розрахунку:
```

```
z=menu ('Вибір розрахунку', 'Розрахунок площі  
поперечного перерізу заготовки', 'Розрахунок об'єму  
заготовки', 'Розрахунок маси заготовки', 'Розрахунок  
ціни заготовки', 'Вихід')
```

```
switch z
```

Розрахунок площі поперечного перетину профілів:

```
case 1
```

```
    k5=menu ('Виберіть профіль', 'Труба  
кругла', 'Швелер з прямими полками', 'Кутик  
рівнобокий', 'Двотавр з прямими полками')
```

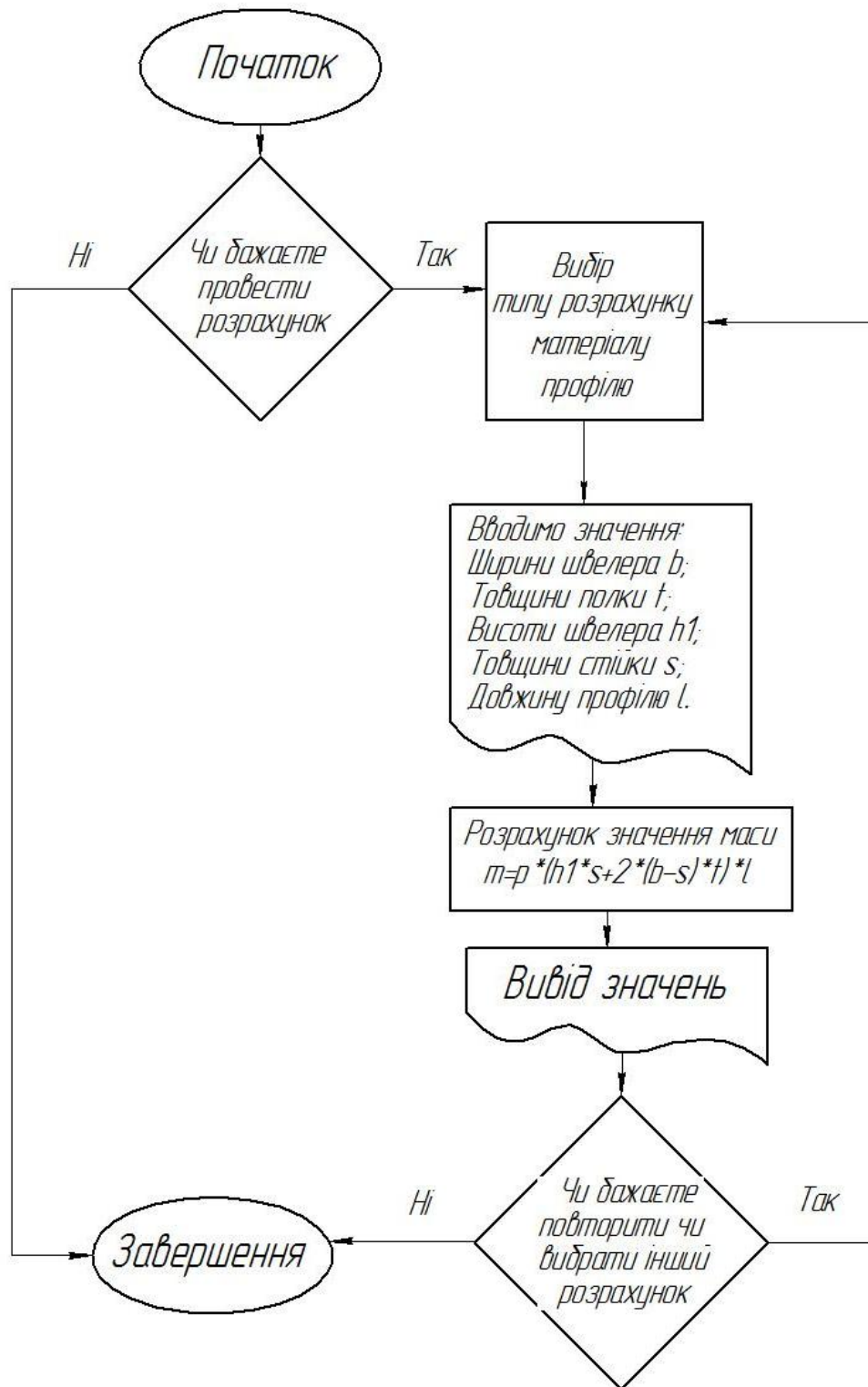


Рис.5. Схема алгоритму розрахунку площі поперечного перетину, об'єму, маси та ціни стандартного та нестандартного сортаменту

switch k5

Розрахунок площі поперечного перетину труби круглого перерізу:

case 1


```

        delete (gca)
        subplot (1,1,1)
        axis ('off')
    disp ('Введіть значення внутрішнього діаметра
труби в мм')
    d=input ('d=')
    disp ('Введіть товщину стінки в мм')
    s=input ('s=')
    S=(3.14*((d+2*s)^2-d^2)/4)/100
    disp ('Площа поперечного перерізу труби кв.см')
    h=text (0,1,'Розрахунок площі поперечного перетину
труби','FontSize',15)
    h=text (0.1,0.5,'Внутрішній діаметр в
мм','FontSize',12)
    d1=num2str (d)
    h=text (1,0.5,d1,'FontSize',12)
    h=text (0.1,0.45,'Товщина стінки в
мм','FontSize',12)
    s1=num2str (s)
    h=text (1,0.45,s1,'FontSize',12)
    h=text (0.1,0.40,'Площа поперечного перетину
профілю в кв см','FontSize',12)
    S1=num2str (S)
    h=text (1,0.40,S1,'FontSize',12)

```

Розрахунок площі поперечного перетину швелера з прямими полками:

```

        case 2
        delete (gca)
        subplot (1,1,1)
        axis ('off')
    disp ('Введіть значення ширини швелера в мм')
    b=input ('b=')
    disp ('Введіть товщину полки в мм')
    t=input ('t=')
    disp ('Введіть висоту швелера в мм')
    h=input ('h=')
    disp ('Введіть товщину стійки в мм')
    s=input ('s=')
    S=(h*s+2*((b-s)*t))/100
    disp ('Площа поперечного перерізу швелера кв.см')
    h=text (-0.1,1,'Розрахунок площі поперечного перерізу
швелера','FontSize',15)
    h=text (0.15,0.9,' з прямими
полками','FontSize',15)

```

```

h=text (0.1,0.5,'Ширина швелера в мм','FontSize',12)
    b1=num2str (b)
    h=text (1,0.5,b1,'FontSize',12)
    h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
    t1=num2str (t)
    h=text (1,0.45,t1,'FontSize',12)
    h=text (0.1,0.40,'Висота швелера в
мм','FontSize',12)
    h2=num2str (h1)
    h=text (1,0.40,h2,'FontSize',12)
    h=text (0.1,0.35,'Товщина стійки в
мм','FontSize',12)
    s1=num2str (s)
    h=text (1,0.35,s1,'FontSize',12)
    h=text (0.1,0.30,'Площа поперечного перерізу
профілю в кв см','FontSize',12)
    S1=num2str (S)
    h=text (1,0.30,S1,'FontSize',12)

```

Розрахунок площі поперечного перетину рівнобокого кутика :

```

        case 3
            delete (gca)
            subplot (1,1,1)
            axis ('off')
            disp ('Введіть значення ширини кутика в мм')
            b=input ('b=')
            disp ('Введіть товщину полки в мм')
            t=input ('t=')
            S=(b*t+(b-t)*t)/100
            disp ('Площа поперечного перерізу кутика кв.см')
            h=text (0.1,1,'Розрахунок площі поперечного
перетину','FontSize',15)
            h=text (0.3,0.9,' рівнобокого кутика','FontSize',15)
            h=text (0.1,0.5,'Ширина кутика в мм','FontSize',12)
            b1=num2str (b)
            h=text (1,0.5,b1,'FontSize',12)
            h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
            t1=num2str (t)
            h=text (1,0.45,t1,'FontSize',12)
            h=text (0.1,0.40,'Площа поперечного перерізу
профілю в кв см','FontSize',12)
            S1=num2str (S)

```

```
h=text (1,0.40,s1,'FontSize',12)
```

Розрахунок площі поперечного перетину двотавра з прямими полками:

```
    case 4
    delete (gca)
    subplot (1,1,1)
    axis ('off')
    disp ('Введіть значення ширини двотавра в мм')
    b=input ('b=')
    disp ('Введіть товщину полки в мм')
    t=input ('t=')
    disp ('Введіть висоту двотавра в мм')
    h=input ('h=')
    disp ('Введіть товщину стійки в мм')
    s=input ('s=')
    S=(b*t*2+(h-2*t)*s)/100
    disp ('Площа поперечного перерізу двотавра
кв.см')
    h=text (-0.1,1,'Розрахунок площі поперечного
перерізу двотавра','FontSize',15)
    h=text (0.15,0.9,' з прямими
полками','FontSize',15)
    h=text (0.1,0.5,'Ширина двотавра в мм','FontSize',12)
    b1=num2str (b)
    h=text (1,0.5,b1,'FontSize',12)
    h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
    t1=num2str (t)
    h=text (1,0.45,t1,'FontSize',12)
    h=text (0.1,0.40,'Висота двотавра в
мм','FontSize',12)
    h2=num2str (h1)
    h=text (1,0.40,h2,'FontSize',12)
    h=text (0.1,0.35,'Товщина стійки в
мм','FontSize',12)
    s1=num2str (s)
    h=text (1,0.35,s1,'FontSize',12)
    h=text (0.1,0.30,'Площа поперечного перерізу
профілю в кв см','FontSize',12)
    S1=num2str (S)
    h=text (1,0.30,S1,'FontSize',12)
end
```

Розрахунок об'єму профілів:

```
case 2
```

```

        k=menu ('Виберіть профіль', 'Труба
кругла', 'Швелер з прямими полками', 'Кутик
рівнобокий', 'Двотавр з прямими полками')
switch k

```

Розрахунок об'єму труби круглого перерізу:

```

    case 1
        delete (gca)
        subplot (1,1,1)
        axis ('off')
        disp ('Введіть значення внутрішнього діаметра
труби в мм')
        d=input ('d=')
        disp ('Введіть товщину стінки в мм')
        s=input ('s=')
        disp ('Введіть довжину труби в мм')
        l=input ('l=')
        V=(3.14*((d+2*s)^2-d^2)*l/4)/1000
        disp ('Об'єм труби куб.см')
        h=text (0,1, 'Розрахунок об'єму
труби', 'FontSize', 15)
        h=text (0.1,0.5, 'Внутрішній діаметр в
мм', 'FontSize', 12)
        d1=num2str (d)
        h=text (0.8,0.5,d1, 'FontSize', 12)
        h=text (0.1,0.45, 'Товщина стінки в
мм', 'FontSize', 12)
        s1=num2str (s)
        h=text (0.8,0.45,s1, 'FontSize', 12)
        h=text (0.1,0.40, 'Довжина профіля в
мм', 'FontSize', 12)
        l1=num2str (l)
        h=text (0.8,0.40,l1, 'FontSize', 12)
        h=text (0.1,0.35, 'Об'єм профілю в куб
см', 'FontSize', 12)
        V1=num2str (V)
        h=text (0.8,0.35,V1, 'FontSize', 12)

```

Розрахунок об'єму швелера з прямими полками:

```

    case 2
        delete (gca)
        subplot (1,1,1)
        axis ('off')
        disp ('Введіть значення ширини швелера в мм')
        b=input ('b=')
        disp ('Введіть товщину полки в мм')

```

```

t=input ('t=')
disp ('Введіть висоту швелера в мм')
h=input ('h=')
disp ('Введіть товщину стійки в мм')
s=input ('s=')
disp ('Введіть довжину швелера в мм')
l=input ('l=')
V=(h*s+2*((b-s)*t))*l/1000
disp ('Об'єм швелера куб.см')
h=text (0,1, 'Розрахунок об'єму швелера з прямими
полками', 'FontSize',15)
h=text (0.1,0.5, 'Ширина швелера в
мм', 'FontSize',12)
b1=num2str (b)
h=text (0.8,0.5,b1, 'FontSize',12)
h=text (0.1,0.45, 'Товщина полки в
мм', 'FontSize',12)
t1=num2str (t)
h=text (0.8,0.45,t1, 'FontSize',12)
h=text (0.1,0.40, 'Висота швелера в
мм', 'FontSize',12)
h2=num2str (h1)
h=text (0.8,0.40,h2, 'FontSize',12)
h=text (0.1,0.35, 'Товщина стійки в
мм', 'FontSize',12)
s1=num2str (s)
h=text (0.8,0.35,s1, 'FontSize',12)
h=text (0.1,0.30, 'Довжина профіля в
мм', 'FontSize',12)
l1=num2str (l)
h=text (0.8,0.30,l1, 'FontSize',12)
h=text (0.1,0.25, 'Об'єм профілю в куб
см', 'FontSize',12)
V1=num2str (V)
h=text (0.8,0.25,V1, 'FontSize',12)
Розрахунок об'єму рівнобокого кутика:
case 3
delete (gca)
subplot (1,1,1)
axis ('off')
disp ('Введіть значення ширини кутика в мм')
b=input ('b=')
disp ('Введіть товщину полки в мм')
t=input ('t=')

```

```

        disp ('Введіть довжину кутика в мм')
        l=input ('l=')
        V=(b*t+(b-t)*t)*l/1000
        disp ('Об'єм кутика куб.см')
        h=text (0,1,'Розрахунок об'єму рівнобокого
кутика','FontSize',15)
        h=text (0.1,0.5,'Ширина кутика в
мм','FontSize',12)
        b1=num2str (b)
        h=text (0.8,0.5,b1,'FontSize',12)
        h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
        t1=num2str (t)
        h=text (0.8,0.45,t1,'FontSize',12)
        h=text (0.1,0.40,'Довжина профіля в
мм','FontSize',12)
        l1=num2str (l)
        h=text (0.8,0.40,l1,'FontSize',12)
        h=text (0.1,0.35,'Об'єм профілю в куб
см','FontSize',12)
        V1=num2str (V)
        h=text (0.8,0.35,V1,'FontSize',12)
Розрахунок об'єму двотавра з прямими полками:
        case 4
            delete (gca)
            subplot (1,1,1)
            axis ('off')
            disp ('Введіть значення ширини двотавра в мм')
            b=input ('b=')
            disp ('Введіть товщину полки в мм')
            t=input ('t=')
            disp ('Введіть висоту двотавра в мм')
            h=input ('h=')
            disp ('Введіть товщину стійки в мм')
            s=input ('s=')
            disp ('Введіть довжину двотавра в мм')
            l=input ('l=');
            V=(b*t*2+(h-2*t)*s)*l/1000
            disp ('Об'єм двотавра куб.см')
            h=text (0,1,'Розрахунок об'єму двотавра з прямими
полками','FontSize',15)
            h=text (0.1,0.5,'Ширина двотавра в
мм','FontSize',12)
            b1=num2str (b)

```

```

        h=text (0.8,0.5,b1,'FontSize',12)
        h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
        t1=num2str (t)
        h=text (0.8,0.45,t1,'FontSize',12)
        h=text (0.1,0.40,'Висота двотавра в
мм','FontSize',12)
        h2=num2str (h1)
        h=text (0.8,0.40,h2,'FontSize',12)
        h=text (0.1,0.35,'Товщина стійки в
мм','FontSize',12)
        s1=num2str (s)
        h=text (0.8,0.35,s1,'FontSize',12)
        h=text (0.1,0.30,'Довжина профіля в
мм','FontSize',12)
        l1=num2str (l)
        h=text (0.8,0.30,l1,'FontSize',12)
        h=text (0.1,0.25,'Об'єм профілю в куб
см','FontSize',12)
        V1=num2str (V)
        h=text (0.8,0.25,V1,'FontSize',12)
    end

```

Розрахунок маси профілів:

```

    case 3

```

Вибір матеріалу для розрахунку маси:

```

        k1=menu ('Виберіть матеріал','Сталь
звичайна','Сталь нержавіюча','Алюміній')

```

```

        switch k1

```

```

            case 1

```

```

                Ro=7.85

```

```

            case 2

```

```

                Ro=7.7

```

```

            case 3

```

```

                Ro=2.7

```

```

        end

```

```

        k2=menu ('Виберіть профіль','Труба
кругла','Швелер з прямими полками','Кутик
рівнобокий','Двотавр з прямими полками')

```

```

        switch k2

```

Розрахунок маси труби круглого перерізу:

```

    case 1

```

```

        delete (gca)

```

```

        subplot (1,1,1)

```

```

        axis ('off')

```

```

disp ('Введіть значення внутрішнього діаметра
труби в мм')
d=input ('d=')
disp ('Введіть товщину стінки в мм')
s=input ('s=')
disp ('Введіть довжину труби в мм')
l=input ('l=')
m=(3.14*((d+2*s)^2-d^2)*l*Ro/4)/(1000*1000)
disp ('Маса труби кг')
h=text (0,1,'Розрахунок маси труби','FontSize',15)
h=text (0.1,0.5,'Внутрішній діаметр в
мм','FontSize',12)
d1=num2str (d)
h=text (0.8,0.5,d1,'FontSize',12)
h=text (0.1,0.45,'Товщина стінки в
мм','FontSize',12)
s1=num2str (s)
h=text (0.8,0.45,s1,'FontSize',12)
h=text (0.1,0.40,'Довжина профіля в
мм','FontSize',12)
l1=num2str (l)
h=text (0.8,0.40,l1,'FontSize',12)
h=text (0.1,0.35,'Густина матеріалу в г/куб см
','FontSize',12)
Ro1=num2str (Ro)
h=text (0.8,0.35,Ro1,'FontSize',12)
h=text (0.1,0.30,'Маса профілю в
кг','FontSize',12)
m1=num2str (m)
h=text (0.8,0.30,m1,'FontSize',12)
Розрахунок маси швелера з прямими полками:
    case 2
        delete (gca)
        subplot (1,1,1)
        axis ('off')
disp ('Введіть значення ширини швелера в мм')
b=input ('b=')
disp ('Введіть товщину полки в мм')
t=input ('t=')
disp ('Введіть висоту швелера в мм')
h=input ('h=')
disp ('Введіть товщину стійки в мм')
s=input ('s=')
disp ('Введіть довжину швелера в мм')

```



```

    l=input ('l=')
    m=(h*s+2*((b-s)*t))*l*Ro/(1000*1000)
    disp ('Маса швелера кг')
    h=text (0,1,'Розрахунок маси швелера з прямими
полками','FontSize',15)
    h=text (0.1,0.5,'Ширина швелера в
мм','FontSize',12)
    b1=num2str (b)
    h=text (0.8,0.5,b1,'FontSize',12)
    h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
    t1=num2str (t)
    h=text (0.8,0.45,t1,'FontSize',12)
    h=text (0.1,0.40,'Висота швелера в
мм','FontSize',12)
    h2=num2str (h1)
    h=text (0.8,0.40,h2,'FontSize',12)
    h=text (0.1,0.35,'Товщина стійки в
мм','FontSize',12)
    s1=num2str (s)
    h=text (0.8,0.35,s1,'FontSize',12)
    h=text (0.1,0.30,'Довжина профіля в
мм','FontSize',12)
    l1=num2str (l)
    h=text (0.8,0.30,l1,'FontSize',12)
    h=text (0.1,0.25,'Густина матеріалу в г/куб см
','FontSize',12)
    Ro1=num2str (Ro)
    h=text (0.8,0.25,Ro1,'FontSize',12)
    h=text (0.1,0.20,'Маса профілю в
кг','FontSize',12)
    m1=num2str (m)
    h=text (0.8,0.20,m1,'FontSize',12)
Розрахунок маси рівнобокого кутика:
    case 3
        delete (gca)
        subplot (1,1,1)
        axis ('off')
        disp ('Введіть значення ширини кутика в мм')
        b=input ('b=')
        disp ('Введіть товщину полки в мм')
        t=input ('t=')
        disp ('Введіть довжину кутика в мм')
        l=input ('l=')

```

```

m=(b*t+(b-t)*t)*l*Ro/(1000*1000)
disp ('Маса кутика кг')
h=text (0,1,'Розрахунок маси рівнобокого
кутика','FontSize',15)
h=text (0.1,0.5,'Ширина кутика в
мм','FontSize',12)
b1=num2str (b)
h=text (0.8,0.5,b1,'FontSize',12)
h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
t1=num2str (t)
h=text (0.8,0.45,t1,'FontSize',12)
h=text (0.1,0.40,'Довжина профіля в
мм','FontSize',12)
l1=num2str (l)
h=text (0.8,0.40,l1,'FontSize',12)
h=text (0.1,0.35,'Густина матеріалу в г/куб см
','FontSize',12)
Ro1=num2str (Ro)
h=text (0.8,0.35,Ro1,'FontSize',12)
h=text (0.1,0.30,'Маса профілю в
кг','FontSize',12)
m1=num2str (m)
h=text (0.8,0.30,m1,'FontSize',12)

```

Розрахунок маси двотавра з прямими полками:

```

case 4
    delete (gca)
    subplot (1,1,1)
    axis ('off')
    disp ('Введіть значення ширини двотавра в мм')
    b=input ('b=')
    disp ('Введіть товщину полки в мм')
    t=input ('t=')
    disp ('Введіть висоту двотавра в мм')
    h=input ('h=')
    disp ('Введіть товщину стійки в мм')
    s=input ('s=')
    disp ('Введіть довжину двотавра в мм')
    l=input ('l=')
    m=(b*t*2+(h-2*t)*s)*l*Ro/(1000*1000)
    disp ('Маса двотавра кг')
    h=text (0,1,'Розрахунок маси двотавра з прямими
полками','FontSize',15)

```

```

        h=text (0.1,0.5,'Ширина двотавра в
мм','FontSize',12)
        b1=num2str (b)
        h=text (0.8,0.5,b1,'FontSize',12)
        h=text (0.1,0.45,'Товщина полки в
мм','FontSize',12)
        t1=num2str (t)
        h=text (0.8,0.45,t1,'FontSize',12)
        h=text (0.1,0.40,'Висота двотавра в
мм','FontSize',12)
        h2=num2str (h1)
        h=text (0.8,0.40,h2,'FontSize',12)
        h=text (0.1,0.35,'Товщина стійки в
мм','FontSize',12)
        s1=num2str (s)
        h=text (0.8,0.35,s1,'FontSize',12)
        h=text (0.1,0.30,'Довжина профіля в
мм','FontSize',12)
        l1=num2str (l)
        h=text (0.8,0.30,l1,'FontSize',12)
        h=text (0.1,0.25,'Густина матеріалу в г/куб см
','FontSize',12)
        Ro1=num2str (Ro)
        h=text (0.8,0.25,Ro1,'FontSize',12)
        h=text (0.1,0.20,'Маса профілю в
кг','FontSize',12)
        m1=num2str (m)
        h=text (0.8,0.20,m1,'FontSize',12)
    end

```

Розрахунок ціни прокату:

```

    case 4
        k3=menu ('Виберіть профіль','Труба
кругла','Швелер з прямими полками','Куттик
рівнобокий','Двотавр з прямими полками')
    switch k3

```

Розрахунок ціни труби круглого перерізу:

```

    case 1
        delete (gca)
        subplot (1,1,1)
        axis ('off')
        disp ('Введіть довжину труби в м')
        l=input ('l=')
        disp ('Введіть ціну за погонний метр в грн')
        p=input ('p=')

```

```

Cina=l*p
disp ('Ціна сортаменту в грн')
h=text (0,1,'Розрахунок ціни труби круглого
перерізу','FontSize',15)
h=text (0.1,0.5,'Довжина профіля в
м','FontSize',12)
l1=num2str (l)
h=text (0.8,0.5,l1,'FontSize',12)
h=text (0.1,0.45,'Ціна погонного метра
грн/м','FontSize',12)
p1=num2str (p)
h=text (0.8,0.45,p1,'FontSize',12)
h=text (0.1,0.40,'Ціна профіля в
грн','FontSize',12)
Cina1=num2str (Cina)
h=text (0.8,0.40,Cina1,'FontSize',12)
Розрахунок ціни швелера з прямими полками:
case 2
delete (gca)
subplot (1,1,1)
axis ('off')
disp ('Введіть довжину швелера в м')
l=input ('l=')
disp ('Введіть ціну за погонний метр в грн')
p=input ('p=')
Cina=l*p
disp ('Ціна сортаменту в грн')
h=text (0,1,'Розрахунок ціни швелера з прямими
полками','FontSize',15)
h=text (0.1,0.5,'Довжина профіля в
м','FontSize',12)
l1=num2str (l)
h=text (0.8,0.5,l1,'FontSize',12)
h=text (0.1,0.45,'Ціна погонного метра
грн/м','FontSize',12)
p1=num2str (p)
h=text (0.8,0.45,p1,'FontSize',12)
h=text (0.1,0.40,'Ціна профіля в
грн','FontSize',12)
Cina1=num2str (Cina)
h=text (0.8,0.40,Cina1,'FontSize',12)
Розрахунок ціни рівнобокого кутика:
case 3
delete (gca)

```

```

        subplot (1,1,1)
        axis ('off')
        disp ('Введіть довжину кутика в м')
        l=input ('l=')
        disp ('Введіть ціну за погонний метр в грн')
        p=input ('p=')
        Cina=l*p
        disp ('Ціна сортаменту в грн')
        h=text (0,1,'Розрахунок ціни рівнобокого
кутика', 'FontSize',15)
        h=text (0.1,0.5,'Довжина профіля в
м', 'FontSize',12)
        l1=num2str (l)
        h=text (0.8,0.5,l1,'FontSize',12)
        h=text (0.1,0.45,'Ціна погонного метра
грн/м', 'FontSize',12)
        p1=num2str (p)
        h=text (0.8,0.45,p1,'FontSize',12)
        h=text (0.1,0.40,'Ціна профіля в
грн', 'FontSize',12)
        Cina1=num2str (Cina)
        h=text (0.8,0.40,Cina1,'FontSize',12)
Розрахунок ціни двотавра з прямими полками:
        case 4
            delete (gca)
            subplot (1,1,1)
            axis ('off')
            disp ('Введіть довжину двотавра в м')
            l=input ('l=')
            disp ('Введіть ціну за погонний метр в грн')
            p=input ('p=')
            Cina=l*p
            disp ('Ціна сортаменту в грн')
            h=text (0,1,'Розрахунок ціни двотавра з прямими
полками', 'FontSize',15)
            h=text (0.1,0.5,'Довжина профіля в
м', 'FontSize',12)
            l1=num2str (l)
            h=text (0.8,0.5,l1,'FontSize',12)
            h=text (0.1,0.45,'Ціна погонного метра
грн/м', 'FontSize',12)
            p1=num2str (p)
            h=text (0.8,0.45,p1,'FontSize',12)

```

```

h=text (0.1,0.40,'Ціна профіля в
грн','FontSize',12)
Cina1=num2str (Cina)
h=text (0.8,0.40,Cina1,'FontSize',12)
end

```

Вихід із програми:

```

case 5
    delete (gcf)
    exit
end
end

```

3. Приклад розрахунку.

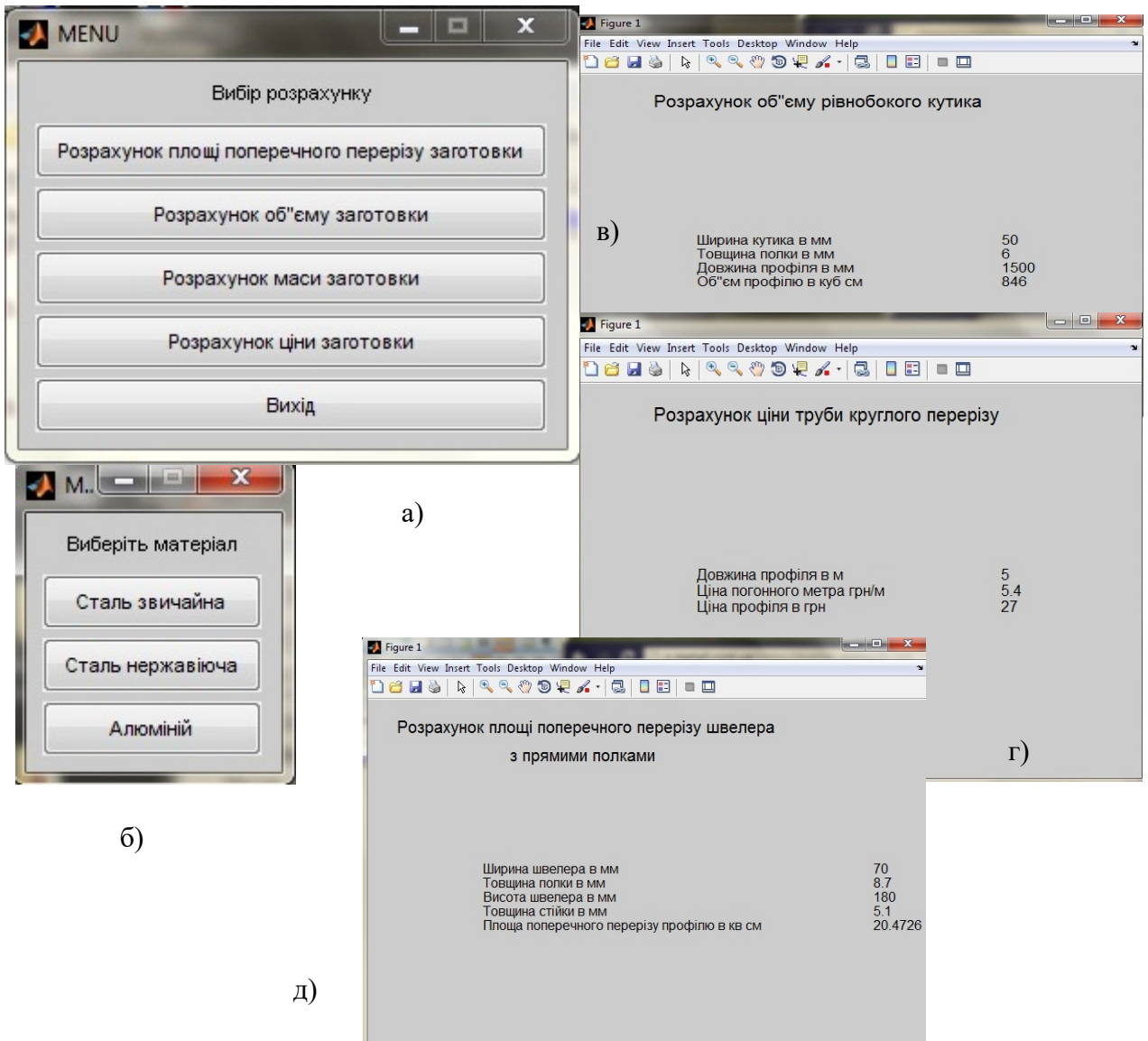


Рис. 6. 1. Вигляд вікон при роботі програми: а- головне вікно; б – вікно, що дозволяє вибрати матеріал; в – вікно з результатами розрахунку об'єму кутка; г – вікно з результатом розрахунку ціни; д – вікно з результатом розрахунку площі поперечного

Висновок. За допомогою цього курсу, я навчився використовувати програмний пакет MATLAB для автоматизації інженерних розрахунків. Навчився складати на основі заданої методики складати алгоритми розрахунків, а також розробив програму, яка спрощує виконання розрахунків та дозволяє скоротити часові витрати.

6.1.2. Розробка програми моделювання поведінки потоку рідини при зміні параметрів трубопроводу за критерієм Рейнольдса

(Розробник програми Глушенко Олег, група МА-02)

Призначення програми

Програма призначена для моделювання поведінки потоку рідини за допомогою автоматичного визначення числа Рейнольдса.

Цільова група: студенти та працівники машинобудівних підприємств

Користь від програми

Програма дозволяє автоматизувати, а відповідно і скоротити час для розрахунку числа Рейнольдса. Також є можливість побудови графіків залежності числа Рейнольдса від (l) та (d).

Постановка задачі

Вхідними даними в даній задачі були діаметр, довжина прохідного пререрізу, коефіцієнт кін.в'язкості, густина.

Самі значення розрахунків отримуємо підставивши вхідні дані у розрахункові формули, що будуть приведені далі.

Методика розрахунку

а) За формулою Дарсі

$$Re = \frac{64}{\lambda}$$

б) За формулою Пуазеля

$$Re = \frac{128\nu\rho l}{\pi D^4}$$

в)

$$Re = \frac{V * D}{\nu}$$

г) Для гнучких трубопроводів

$$Re = \frac{68}{\lambda}$$

Де λ - коефіцієнт кін.в'язкості

1. Визначаємося з вибором розрахунку (вибираємо формулу)
2. Задати геометричні розміри профілю:
3. Вивести значення числани екран.

Текст програми

```
%Очищення оперативної пам'яті
clear;
%Підготовка текстової інформації для виведення у вікно
програми
text1='                                     Самостійна
робота                                     ';
text2='          з дисципліни "Основи автоматичного
проектуння"                               ';
text3='          Виконав студент 3го курсу
гр.МА-02 ММІ                               ';
text4='Глущенко Олег';
text5={text1, text2, text3, text4};
m1=0, m2=0,m3=0,m4=0,m5=0;
f1=0, f2=0, f3=0, f4=0;
t1='', t2='', t3='', t4='', t5='', t6='';
%Запуск головного циклу програми
while m1~=6
%Виведення графічного інтерактивного меню на екран
    m1=menu('Виберіть
формулу', 'Re= $\lambda/64$ ', 'Re= $128 \cdot \nu \cdot p \cdot l / \pi \cdot d^4$ ', 'Re= $v \cdot d / \nu$ 
', 'Re= $\lambda/68$ ', 'Про програму', 'Вихід')
%Блок реагування на команди користувача при натискання
на кнопки інтерфейсу
%Якщо вибрали 1-у формулу, то
if m1==1
    m2=0;
    f1=0;
%Запуск внутрішнього циклу програми
    while m2~=3
%Блок реагування на команди користувача при натискання
на кнопки інтерфейсу внутрішнього циклу
        m2=menu('Re= $\lambda/64$ ', 'Ввести
lamda', 'Графік', 'Головне меню')
        %Якщо вибрали ввести ламда, то
        if m2==1
            lamda1=inputdlg('Lamda', 'Dani', 1, {'200'});
            Re1=str2num(lamda1{1})/64;
            f1=1;
        end
        %Якщо вибрали Графік, то
        if m2==2
%Виведення графіку у вікно
            clf;
```

```

        hold all
        subplot(2,1,1)
        lamda=0:5:70000;
        Re=lamda/64;
        p1=plot(lamda,Re)
        title('Re=lamda/64');
        xlabel('lamda');
        ylabel('Re');
        grid on
        if f1==1
            subplot(2,1,2)
                axis off
                %Виведення текстової інформації у вікно
                t1=text(0,1, strcat('lamda=', lamda1{1}), 'fontsize', 14);
                t2=text(0,0.8, strcat('Re=', num2str(Re1)), 'fontsize', 14);
                %Перевірка значень чисел Рейнольдса та виведення типу
                режиму течії
                    if (Re1>=2320)
                        t3=text(0,0.6, 'Turbulentnyy
potik', 'fontsize', 14);
                    end
                    if (Re1<2320)
                        t3=text(0,0.6, 'Laminarnyy
potik', 'fontsize', 14);
                    end
                    else
                        warndlg('Введіть дані!');
                    end
                end
            end
        end
        %Якщо вибрали 4-у формулу, то
        if m1==4
            m5=0;
            f4=0;
            %Запуск внутрішнього циклу програми
            while m5~=3
                %Блок реагування на команди користувача при натискання
                на кнопки інтерфейсу внутрішнього циклу
                m5=menu('Re=lamda/68', 'Ввести
lamda', 'Графік', 'Головне меню')
                %Якщо вибрали ввести ламда, то
                if m5==1
                    lamda1=inputdlg('Lamda', 'Dani', 1, {'200'});

```

```

        Re1=str2num(lamda1{1})/68
        f4=1;
end
%Якщо вибрали графік, то
    if m5==2
%Виведення графіку у вікно
        clf;
        hold all
        subplot(2,1,1)
        lamda=0:5:70000;
        Re=lamda/68;
        p2=plot(lamda,Re)
        title('Re=lamda/68');
        xlabel('lamda');
        ylabel('Re');
        grid on
        if f4==1
            subplot(2,1,2)
            axis off
%Виведення текстової інформації у вікно
            t1=text(0,1,strcat('lamda=',lamda1{1}),'fontsize',14);
            t2=text(0,0.8,strcat('Re=',num2str(Re1)), 'fontsize',14);
            if (Re1>=2320)
                t3=text(0,0.6,'Turbulentnyy
potik','fontsize',14);
            end
%Перевірка значень чисел Рейнольдса та виведення типу
режиму течії
            if Re1<2320
                t3=text(0,0.6,'Laminarnyy
potik','fontsize',14);
            end
            else
                warndlg('Введіть дані!');
            end
        end
    end
end
%Якщо вибрали 2-у формулу, то
if m1==2
    m3=0;
    f2=0;
%Запуск внутрішнього циклу програми
    while m3~=4

```

%Блок реагування на команди користувача при натискання на кнопки інтерфейсу внутрішнього циклу

```
m3=menu('Re=128*nu*p*l/pi*d^4','Ввести дані','Графік Re(d)','Графік Re(l)','Головне меню')
```

%Якщо вибрали введення даних, то

```
if m3==1
```

```
data=inputdlg({'nu','p','l','d'},'Dani',1,{'120','1000','20','0.05'});
```

```
Re1=128*str2num(data{1})*str2num(data{2})*str2num(data{3})/pi*(str2num(data{4}))^4
```

```
f2=1;
```

```
end
```

%якщо вибрали графік Re(d), то

```
if m3==2
```

%Виведення графіку у вікно

```
clf;
```

```
if f2==1
```

```
hold all
```

```
subplot(2,1,1)
```

```
d=0:0.1:1;
```

```
Re=128*str2num(data{1})*str2num(data{2})*str2num(data{3})/pi*d.^4;
```

```
p3=plot(d,Re)
```

```
title('Re(d)=128*nu*p*l/pi*d^4');
```

```
xlabel('d');
```

```
ylabel('Re');
```

```
grid on
```

```
subplot(2,1,2)
```

```
axis off
```

%Виведення текстової інформації у вікно

```
t1=text(0,1,strcat('nu=',data{1}),'fontsize',14);
```

```
t2=text(0,0.8,strcat('p=',data{2}),'fontsize',14);
```

```
t3=text(0,0.6,strcat('l=',data{3}),'fontsize',14);
```

```
t4=text(0,0.4,strcat('d=',data{4}),'fontsize',14);
```

```
t5=text(0,0.2,strcat('Re=',num2str(Re1)), 'fontsize',14);
```

%Перевірка значень чисел Рейнольдса та виведення типу режиму течії

```
if (Re1>=2320)
```

```
t6=text(0,0,'Turbulentnuy potik','fontsize',14);
```

```
end
```

```
if Re1<2320
```

```
t6=text(0,0,'Laminarnuy potik','fontsize',14);
```

```

        end
        else
            warndlg('Введіть дані!');
        end
    end
    %якщо вибрали графік Re(1), то
    if m3==3
%Виведення графіку у вікно
        clf;
        if f2==1
            hold all
            subplot(2,1,1)
            l=0:1:100;
            Re=128*str2num(data{1})*str2num(data{2})*l/pi*(str2num(d
            ata{4}))^4;
            p4=plot(l,Re)
            title('Re(1)=128*nu*p*l/pi*d^4');
            xlabel('l');
            ylabel('Re');
            grid on
            subplot(2,1,2)
            axis off
%Виведення текстової інформації у вікно
            t1=text(0,1,strcat('nu=',data{1}),'fontsize',14);
            t2=text(0,0.8,strcat('p=',data{2}),'fontsize',14);
            t3=text(0,0.6,strcat('l=',data{3}),'fontsize',14);
            t4=text(0,0.4,strcat('d=',data{4}),'fontsize',14);
            t5=text(0,0.2,strcat('Re=',num2str(Re1)),'fontsize',14);
%Перевірка значень чисел Рейнольдса та виведення типу
режиму течії
            if (Re1>=2320)
                t6=text(0,0,'Turbulentnuy
                potik','fontsize',14);
            end
            if Re1<2320
                t6=text(0,0,'Laminarnuy
                potik','fontsize',14);
            end
            else
                warndlg('Введіть дані!');
            end
        end
    end
end
end
end

```

```

%Якщо вибрали 3-у формулу, то
if m1==3
    m4=0;
    f3=0;
%Запуск внутрішнього циклу програми
    while m4~=4
%Блок реагування на команди користувача при натискання
на кнопки інтерфейсу внутрішнього циклу
        m4=menu('Re=v*d/nu', 'Ввести дані', 'Графік
Re(d)', 'Графік Re(v)', 'Головне меню')
        %Якщо вибрали ввести дані, то
        if m4==1
data=inputdlg({'v', 'd', 'nu'}, 'Дані', 1, {'70', '0.05', '25.6
'});
Re1=str2num(data{1})*str2num(data{2})/str2num(data{3})
            f3=1;
        end
        %якщо вибрали графік Re(d), то
        if m4==2
%Виведення графіку у вікно
            clf;
            if f3==1
                hold all
                subplot(2,1,1)
                d=0:0.1:2;
                Re=str2num(data{1})*d/str2num(data{3});
                p5=plot(d,Re)
                title('Re(d)=Re=v*d/nu');
                xlabel('d');
                ylabel('Re');
                grid on
                subplot(2,1,2)
                axis off
            %Виведення текстової інформації у вікно
            t1=text(0,1,strcat('v=',data{1}), 'fontsize',14);
            t2=text(0,0.8,strcat('s=',data{2}), 'fontsize',14);
            t3=text(0,0.6,strcat('nu=',data{3}), 'fontsize',14);
            t4=text(0,0.4,strcat('Re=',num2str(Re1)), 'fontsize',14);
%Перевірка значень чисел Рейнольдса та виведення типу
режиму течії
            if (Re1>=2320)
                t5=text(0,0.2, 'Turbulentnuy
potik', 'fontsize',14);
            end
        end
    end
end

```

```

        if Re1<2320
            t5=text(0,0.2,'Laminarnuy
potik','fontsize',14);
        end
    else
        warndlg('Введіть дані!');
    end
end
    %якщо вибрали графік Re(v), то
    if m4==3
%Виведення графіку у вікно
        clf;
        if f3==1
            hold all
            subplot(2,1,1)
            v=0:1:1000;
            Re=v*str2num(data{2})/str2num(data{3});
            p6=plot(v,Re)
            title('Re(v)=Re=v*d/nu');
            xlabel('v');
            ylabel('Re');
            grid on
            subplot(2,1,2)
            axis off

%Виведення текстової інформації у вікно
            t1=text(0,1,strcat('v=',data{1}),'fontsize',14);
            t2=text(0,0.8,strcat('s=',data{2}),'fontsize',14);
            t3=text(0,0.6,strcat('nu=',data{3}),'fontsize',14);
            t4=text(0,0.4,strcat('Re=',num2str(Re1)),'fontsize',14);
            if (Re1>=2320)
                t5=text(0,0.2,'Turbulentnuy
potik','fontsize',14);
            end
%Перевірка значень чисел Рейнольдса та виведення типу
режиму течії
            if Re1<2320
                t5=text(0,0.2,'Laminarnuy
potik','fontsize',14);
            end
            else
                warndlg('Введіть дані!');
            end
        end
    end
end
end

```

```
end
  if m1==5
    msgbox(text5, 'Про програму')
  end
end
```

6.Алгоритм розрахунку

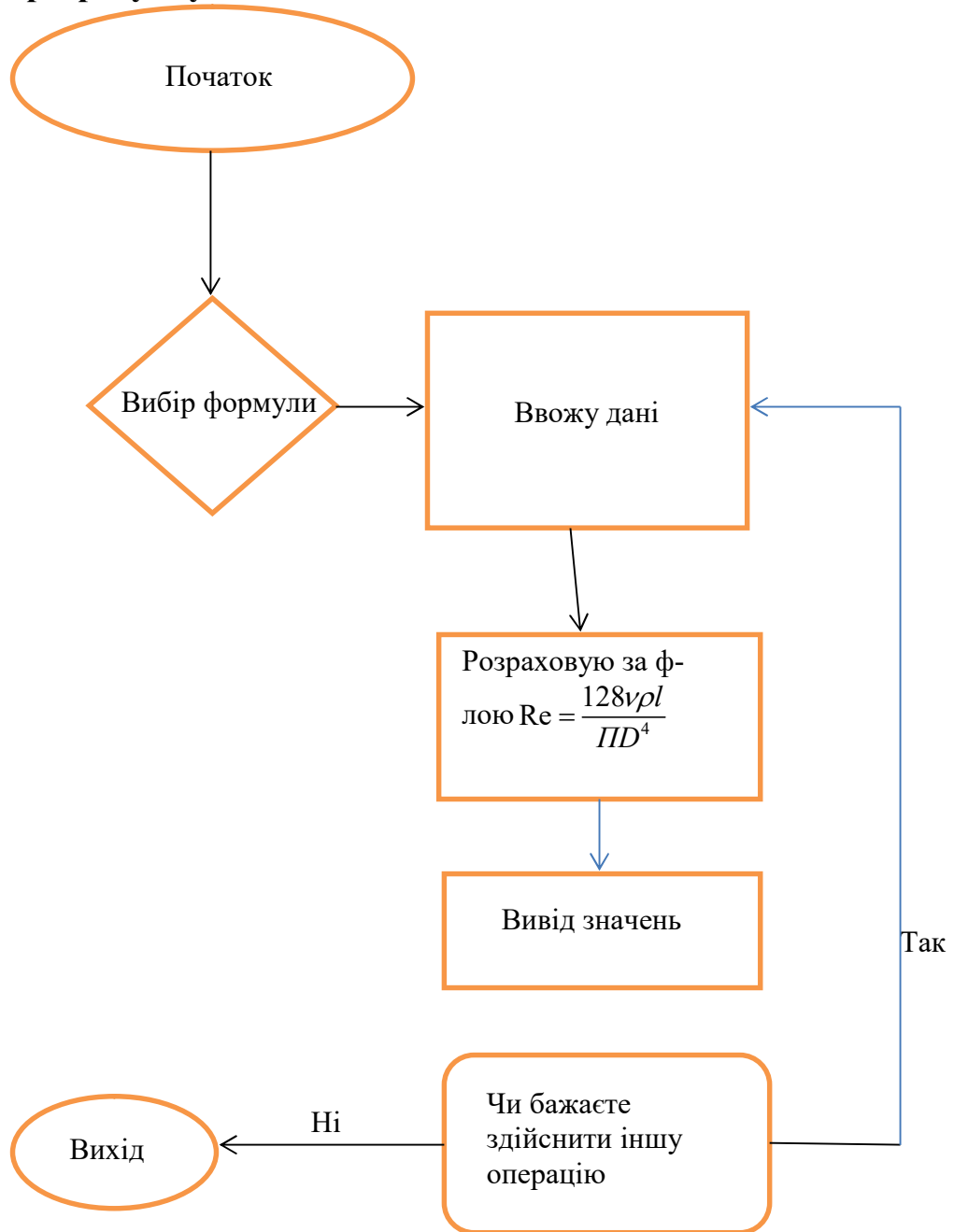
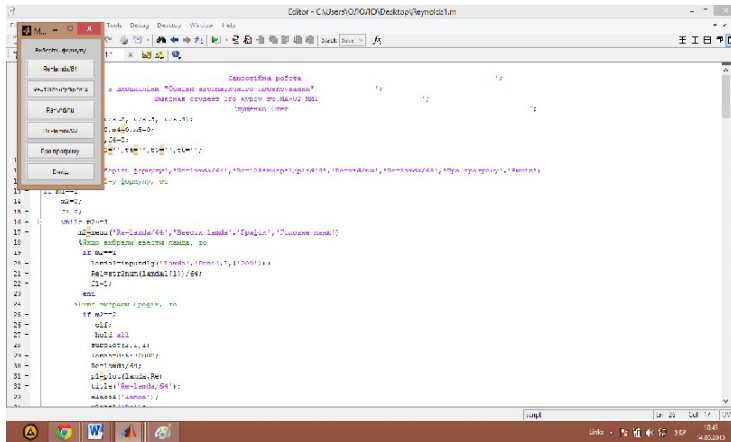
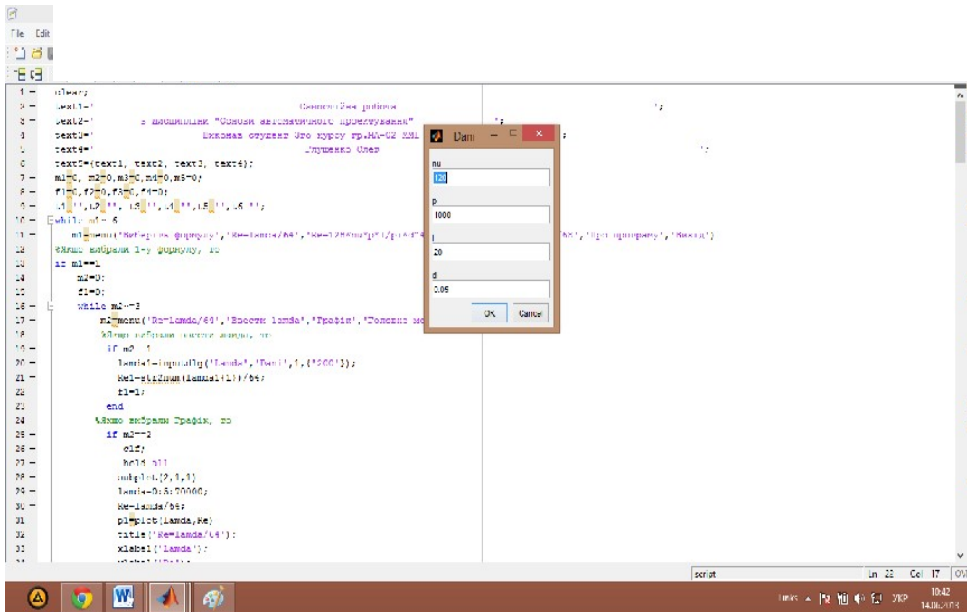


Рис.6.2. Алгоритм моделювання поведінки потоку рідини при зміні параметрів трубопроводу за критерієм Рейнольдса

Приклад розрахунку



a)



б)

Рис.6.3. Вигляд вікон вводу даних для моделювання поведінки потоку: а – меню вибору базової формули; б – меню завдання параметрів потоку.

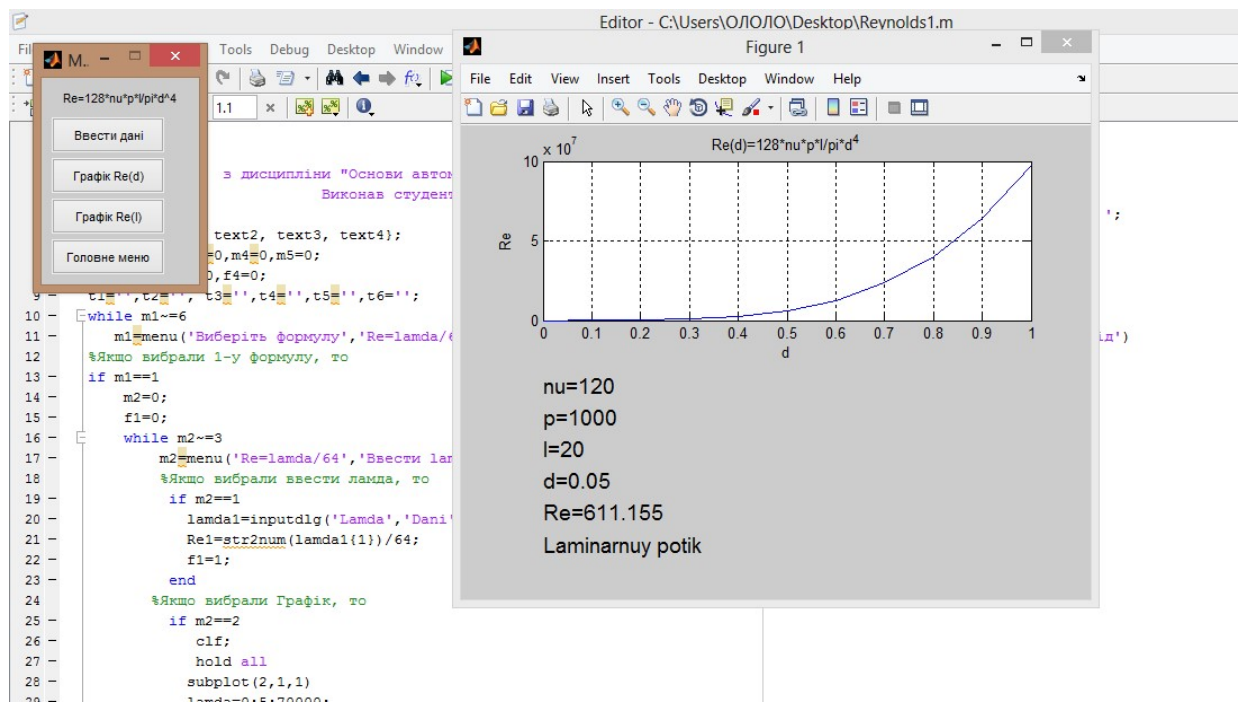


Рис.6.4. Вигляд вікон вводу результату моделювання поведінки потоку

Висновок. При вивченні курсу я навчився використовувати MATLAB у практичній цілях, навчився складати алгоритми розрахунків, а також розробив програму яка спрощує розрахунки числа Рейнольдса.

6.2.3. Моделювання дії штучного інтелекту для визначення округлих форм на заданому зображенні (розробник програми Ключев Василь, гр. МА-01)

Призначення програми.

Програма призначена для розпізнання округлих форм на зображенні (бульбашки, кола, отвори та інші) із подальшим графічним виводом результатів аналізу та необхідного графіку.

Цільова група: наукова гілка спільноти (студенти, лаборанти, вчені, викладачі), а також дослідники та дизайнери.

1. Користь від програми.

Програма має досить широкий спектр застосування, від дизайнерських потреб до дослідження явищ кавітації. Так, людина, що досліджує це явище, маючи декілька зображень, на яких відображено у певні моменти часу кавітаційні бульбашки та їх утворення, зможе робити дослідження у цьому напрямку, завантаживши ці зображення у програму та проаналізувавши їх. Програма автоматично розпізнає та позначить на зображенні бульбашки, їх розміри і побудує графік.

Також програма несе в собі демонстраційні властивості потужності пакету MatLab Image Toolbox, що має користь при початковій стадії опанування цього програмного пакету студентами чи учнями. Це дозволить популяризувати програмування не лише серед студентів інженерних спеціальностей, а також серед інших верств населення.

2. Постановка задачі.

- Вхідними даними є зображення, на якому відображені округлі форми, що мають бути розпізнані автоматично. Критеріями для зображення є: чіткість, розмір не більш як 1600x1800px, округлість правильної форми (коловидної), відсутність відблисків або інших дефектів зображення. Формат зображення: jpg, png.

- Вихідні дані: проаналізоване зображення із нанесеними на ньому діаметрами округлостей і виводом графіку.

3. Методика роботи програми.

Використовуючи потенціал спеціальних операторів та функцій, що входять до пакету MatLab Image Toolbox ми маємо прості та водночас якісні

процеси розпізнання форм на зображеннях, що не потребують поглиблених знань у програмуванні, мають початкові оптимізації, що позитивно впливає на швидкодії програми.

4.Алгоритм програми.

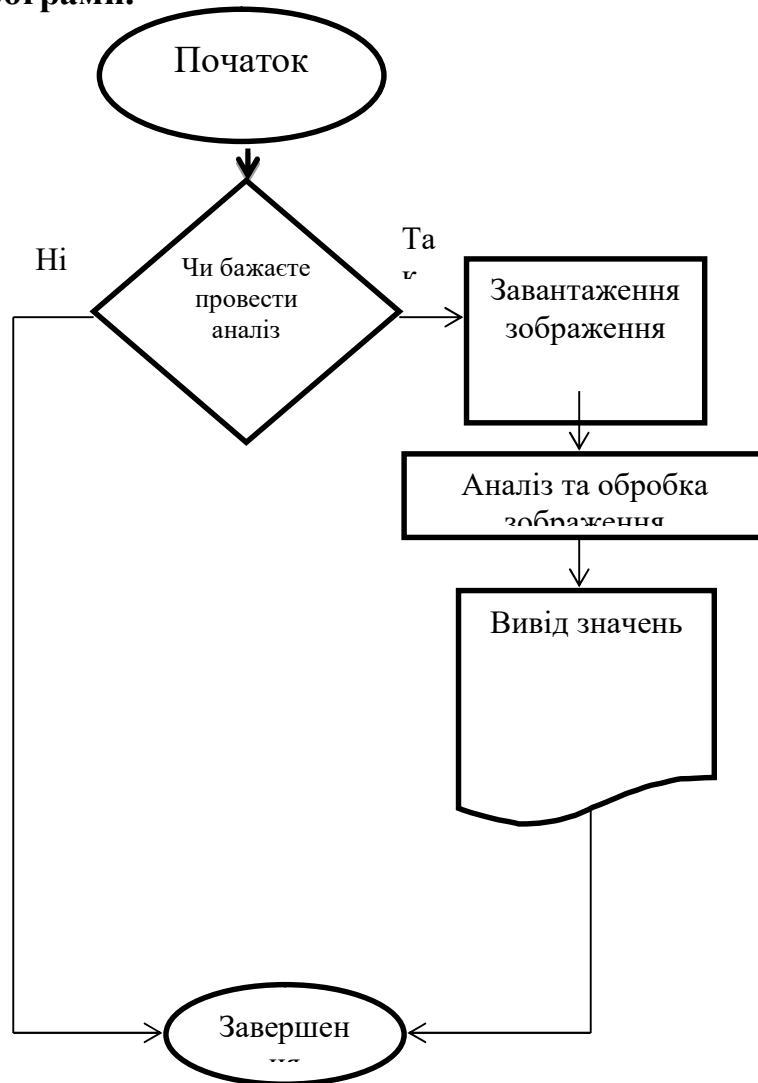


Рис.6.5. Алгоритм моделювання дії штучного інтелекту для визначення округлих

Текст програми

```
clear all;close all;clc;

img = imread('bubbles.jpg');
%відкриття зображення. У поле введіть свою назву
зображення
grayimg = rgb2gray(img);
%переведення зображення у відтінки сірого
grayimg = imadjust(grayimg);
%контрастування
bw = edge(grayimg,'canny', 0.15, 2);
%виділення меж методом "Канні". Також можливі методи
'sobel', 'prewitt', 'roberts'
bw = imfill(bw,'holes');
%заливання замкнутих областей
se = strel('disk',1);
%формування структурного елемента
bw = imopen(bw,se);
%морфологічне відкриття (позбавлення тонких ліній)

[B,L] = bwboundaries(bw);
%відслідковує зовнішні межі об`єкту та виводить 2
результата. B-структурний
%масив розміром Px1, P-кількість об`єктів на зображенні.
Елементи масиву B
%- матриці розміром Qx2, де Q - кількість пікселів, що
належать границі
%об`єкта. L-матриця, розмір котрої рівен розміру
зображення, маюча %невід`емні числа, котрі відповідають
замкнутим областям.
stats = regionprops(L,'Centroid','EquivDiameter');
%утворює масив структур stats, поля елементів котрого
Centroid i
%EquivDiameter мають координати центру області і діаметр
області
figure, imshow(img)
hold on
%Виведення результатів
for k = 1:length(B)
    boundary = B{k};
    radius = stats(k).EquivDiameter/2;
    xc = stats(k).Centroid(1);
    yc = stats(k).Centroid(2);
    theta = 0:0.01:2*pi;
```

```

Xfit = radius*cos(theta) + xc;
Yfit = radius*sin(theta) + yc;
plot(Xfit, Yfit, 'g');
text(boundary(1,2)-15,boundary(1,1)+15,
num2str(radius,3), 'Color', 'y', ...
'FontSize', 8);
end

```

%виведення графіку

```

radius2=zeros (length(B));
xc2=zeros (length(B));
yc2=zeros (length(B));
for k = 1:length(B)
boundary = B{k};
radius2(k) = stats(k).EquivDiameter/2;
xc2(k) = stats(k).Centroid(1);
yc2(k) = stats(k).Centroid(2);
theta = 0:0.01:2*pi;
Xfit2 = radius2(k).*cos(theta) + xc2(k);
Yfit2 = radius2(k).*sin(theta) + yc2(k);

end
figure;
plot(radius2);
title('Radius');

```

7. Приклад аналізу.

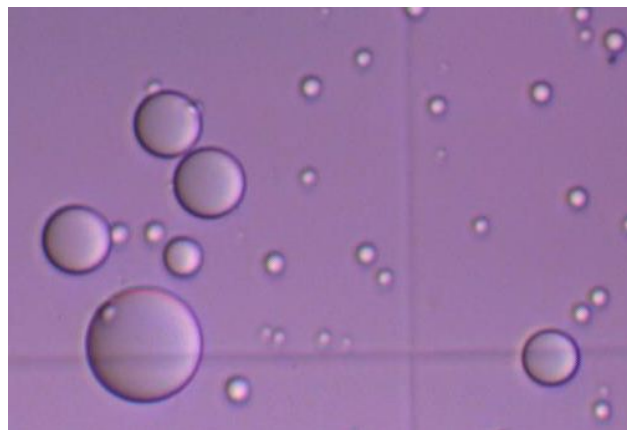
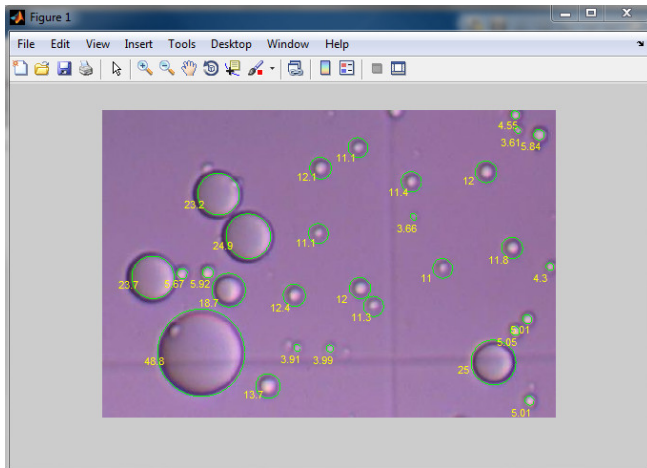
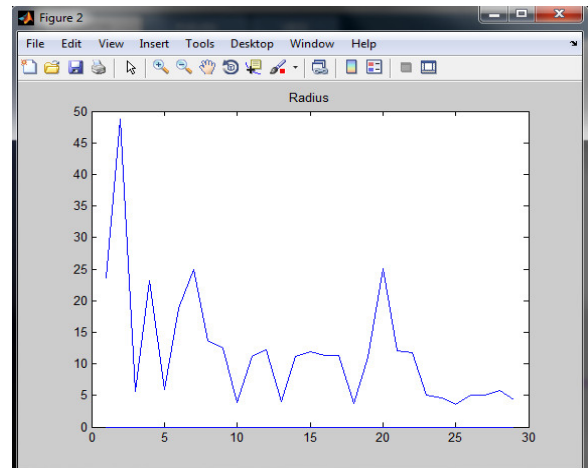


Рис.6.6. Приклад фото для аналізування



а)



б)

Рис.б.6. Приклад результату роботи програми: а – оброблено фото з визначеними параметрами круглих форм; б – графік розподілення круглих форм за їх розмірами

Висновок. при вивченні цього курсу, я навчився використовувати пакет MATLAB у практичних цілях, відкрив для себе потужне доповнення Matlab Image Toolbox, навчився складати алгоритми та розв'язувати різного типу задачі.