

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ПРОГРАМНІ ЗАСОБИ МОДЕЛЮВАННЯ СИСТЕМ КЕРУВАННЯ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання робіт комп'ютерного практикуму для студентів напряму
підготовки «Автоматизація та комп'ютерно-інтегровані технології»

Рекомендовано Вченою радою інженерно-хімічного факультету

Київ
НТУУ «КПІ»
2012

Програмні засоби моделювання систем керування : Метод. вказівки до викон. робіт комп. практ. для студ. напр. „Автоматизація та комп'ютерно-інтегровані технології” / Уклад.: М. В. Коржик. – К. : НТУУ «КПІ», 2012. – 44 с.

*Гриф надано Вченою радою ІХФ
(Протокол № 3 від 27 лютого 2012 р.)*

Навчальне видання

ПРОГРАМНІ ЗАСОБИ МОДЕЛЮВАННЯ СИСТЕМ КЕРУВАННЯ
Методичні вказівки до виконання робіт комп'ютерного практикуму для студентів
напрямку підготовки «Автоматизація та комп'ютерно-інтегровані технології»

Укладач	Коржик Михайло Володимирович, канд. техн. наук
Відповідальний редактор	А. І. Жученко, докт. техн. наук, проф.
Рецензент	С. Ю. Олійник, канд. техн. наук, доц.

За редакцією укладача

Зміст

	Стор.
Вступ	4
Робота № 1. Інтерполяція даних	5
Робота № 2. Символьні розрахунки	8
Робота № 3. Розв'язання звичайних диференціальних рівнянь	11
Робота № 4. Розв'язання диференціальних рівнянь в частинних похідних	13
Робота № 5. Дослідження lti-об'єктів в часовій області	17
Робота № 6. Дослідження lti-об'єктів в частотній області	19
Робота № 7. Властивості lti-об'єктів	22
Робота № 8. Ідентифікація динамічних об'єктів	25
Робота № 9. Синтез систем методом кореневого годографа	30
Робота № 10. Розрахунок систем на заданий показник коливності	33
Робота № 11. Імітаційне моделювання динамічних об'єктів	35
Робота № 12. Імітаційне моделювання замкнених систем	37
Контрольна робота	39
Список рекомендованої літератури	39
Додатки	41

Вступ

Дисципліна “Програмні засоби моделювання систем керування” має на меті ознайомлення студентів бакалаврської програми навчання із сучасними програмними продуктами прикладної математики, які здатні полегшити розв’язання широкого спектра професійних задач, передбачених напрямом підготовки “Автоматизація та комп’ютерно-інтегровані технології”.

Цей практикум призначено для опанування студентами базових можливостей високорівневої мови програмування та системи технічних і наукових розрахунків MatLab, а також для набуття практичних навичок в моделюванні, аналізі та синтезі лінійних стаціонарних систем керування та окремих їх елементів засобами спеціалізованих пакетів розширення MatLab: Control System Toolbox і System Identification Toolbox, та за допомогою спеціалізованого інструменту імітаційного моделювання – Simulink. Крім того в практикумі використовуються інші стандартні пакети розширення MatLab для розв’язання деяких суміжних задач.

Порядок виконання робіт

Роботи виконуються шляхом програмування і виконання поставлених задач та аналізу отриманих результатів. Кожна задача розв’язується у відповідності з наступним алгоритмом:

1. Записати у протокол роботи поставлену задачу.
2. Сформулювати список команд, функцій та інших засобів MatLab для розв’язання поставленої задачі.
3. Користуючись довідковою системою, навести короткі відомості щодо призначення та форматів застосування нових (таких, що не використовувались у попередніх роботах) засобів MatLab.
4. Навести програмний код командного вікна та/або тексти скриптів і функцій користувача для розв’язання поставленої задачі.

5. За необхідності навести журнал виконання задачі.
6. Навести у протоколі результати розв'язання задачі.
7. Зробити висновки з виконаної роботи.

Результати розв'язання задач кожної роботи наводяться у вигляді тексту виводу у командне вікно MatLab, роздруківок графічних вікон та графіків, побудованих у відповідності з вимогами, зазначеними викладачем.

Повністю оформлений протокол виконаної роботи подається для перевірки перед початком наступної роботи.

Робота № 1

ІНТЕРПОЛЯЦІЯ ДАНИХ

Мета роботи: набути практичних навичок у використанні базових засобів MatLab для візуалізації, інтерполяції та згладжування даних.

Теоретичні відомості

Для візуалізації даних MatLab пропонує велику кількість гнучких засобів, які забезпечують побудову і редагування діаграм, тривимірних поверхонь та геометричних фігур, а також графіків пласких і просторових кривих в різних системах координат [1, 9, 15]. Зокрема, для побудови графіків в декартових координатах використовується функція `plot` та супутні команди і функції, що дозволяють змінювати деякі атрибути графічного вікна: `xlabel`, `ylabel`, `legend`, `grid` тощо.

Інтерполяція – це визначення проміжних значень величини за відомим дискретним набором даних (зокрема, отриманим експериментально).

Для інтерполяції вибірки даних розмірністю n можна застосувати поліноміальну інтерполяцію, де інтерполяційний поліном степеня не вище $n - 1$ має вид:

$$P(x) = p_1 x^{n-1} + p_2 x^{n-2} + \dots + p_{n-1} x + p_0. \quad (1.1)$$

Зазвичай поліноміальну інтерполяцію застосовують для невеликих вибірок і степінь поліному, як правило, не перевищує 6.

В системі MatLab для поліноміальної інтерполяції застосовується функція `polyfit`, яка за вихідними даними розраховує коефіцієнти полінома (1.1) методом найменших квадратів [9]. При цьому для розрахунку значень полінома можна скористатись функцією `polyval`.

Для підвищення якості апроксимації при роботі з великими вибірками застосовується кусково-поліноміальна інтерполяція [9, 15], яка в системі MatLab реалізується функцією `interp1`. При цьому функція забезпечує наступні методи інтерполяції: `'linear'` – лінійна, `'cubic'` – інтерполяція ковзним кубічним поліномом, `'spline'` – сплайнова, та деякі інші.

Для застосування всіх можливостей теорії сплайнів можна скористатися додатковими бібліотеками `Curve Fitting Toolbox` або `Spline Toolbox` [8], що містять велику кількість відповідних спеціалізованих функцій. Наприклад, якщо виникає необхідність згладжування експериментальних даних, зручно використовувати функцію `csaps`, яка розраховує кубічний сплайн із заданим фактором згладжування.

Завдання на роботу

1. Побудувати в графічному вікні графіки функції `sin`, як це зображено на рис. 1. При цьому графік y_1 (із кроком 1 за x) побудувати лише точками (в подальшому їх ми будемо вважати результатами експериментальної), а графік y_2 (із кроком 0.1 за x) побудувати безперервною лінією і вважати етлонною функцією.

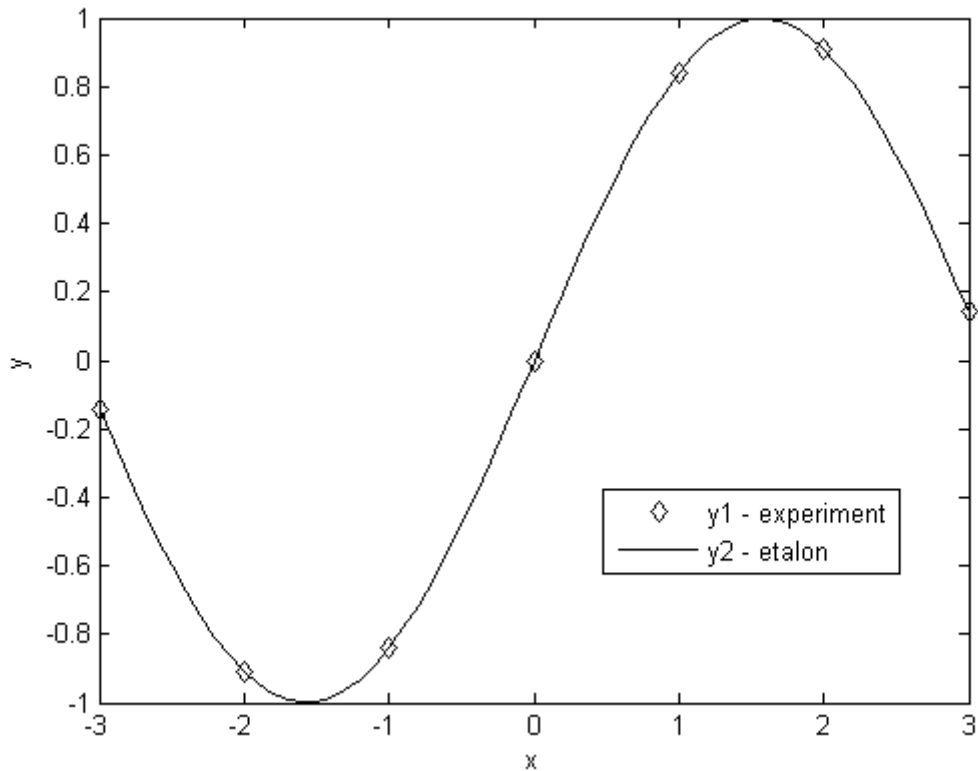


Рис. 1. Обробка результатів експерименту

2. В графічному вікні (див. п. 1) за експериментальними даними y_1 побудувати інтерполяційні поліноми степеня: 1, 2, 4 та 6. Зробити висновки.

3. В графічному вікні (див. п. 1) за експериментальними даними y_1 побудувати апроксимаційні криві, що відповідають різним методам ковзної інтерполяції (не менше 4-х методів). Зробити висновки.

4. В графічному вікні (див. п. 1) за експериментальними даними y_1 побудувати згладжені графіки, що відповідають фактору згладжування: 0.1, 0.5, 0.9 та 1. Зробити висновки.

Контрольні запитання

1. Для яких ще систем координат реалізовані графічні функції в системі MatLab? Наведіть приклади.

2. Як задати діапазони осей графічного вікна?

3. Що таке ковзний поліном та сплайн?

4. В чому полягає метод інтерполяції 'nearest'?

5. Що таке екстраполяція та що таке згладжування даних?

Робота № 2

СИМВОЛЬНІ РОЗРАХУНКИ

Мета роботи: набути практичних навичок у створенні функцій користувача MatLab та використанні засобів бібліотеки SMT для символьних розрахунків.

Теоретичні відомості

Система MatLab передусім призначена для числових розрахунків над матрицями дійсного типу подвійної точності, що дозволяє отримувати в більшості випадків лише “приблизні” (в математичному сенсі) частинні результати. Для розв’язання математичних задач в загальній формі без втрати точності використовують символьні розрахунки. До такого класу задач належать: аналітичне розв’язання систем алгебраїчних та диференціальних рівнянь, інтегральні перетворення, знаходження меж функцій, розрахунок похідних та невизначених інтегралів, виконання підстановок та спрощення математичних виразів тощо. Для здійснення символьних (або аналітичних) розрахунків в системі MatLab використовується додаткова бібліотека Symbolic Math Toolbox [2, 11], що базується на ядрі відомої системи символьної математики Maple.

Бібліотека SMT надає доступ до всіх базових функції ядра Maple через відповідну інтерфейсну конструкцію, а також дозволяє виконувати безпосередньо в MatLab найбільш важливі та широкоживані символьні операції. Зокрема, для розвинення у ряд Тейлора використовується функція `taylor`. Повний перелік доступних функцій символьних розрахунків можна визначити за допомогою команди `help symbolic`.

Змінні в системі MatLab за замовчуванням не визначені і вважаються числовими. Вони ініціалізуються в момент першого до них звернення. Змінна отримує символьний тип, якщо їй привласнити символьний вираз (тоб-то рядок символів, взятий у апострофи).

При реалізації символьних розрахунків, змінні, що беруть у них участь, повинні мати символьний тип. У найпростішому випадку змінні можна оголосити, як символьні, безпосередньо у виразі, взявши їх імена у апострофи. Для попереднього оголошення символьних змінних використовується функція `sym` або команда `syms`.

Для числового розв'язання математичного виразу, поданого в символьній формі, використовується базова функція `eval`.

MatLab, як і більшість інших мов програмування високого рівня, дозволяє створювати власні функції користувача за допомогою ключового слова `function`, та використовувати традиційні керуючі конструкції на кшталт `if-else-end`, `while-end` або `switch-case-otherwise-end` (ключове слово `end` в мові MatLab завершує будь-яку керуючу конструкцію) [1, 9].

При використанні регулярних циклів (конструкція `for-end`) для обробки масивів даних, часто корисною є функція `length`, що визначає кількість елементів вектора (одновимірного масиву) і може застосовуватись як кінцеве значення керуючої змінної циклу.

Завдання на роботу

1. Побудувати в графічному вікні графік функції, наданої викладачем, наприклад, див. рис. 2.

2. Оголосити змінну аргумента функції, як символьну та здійснити розвинення наданої функції у ряд Маклорена, в якому кількість членів дорівнює 10.

3. Створити функцію користувача, що розраховує надану функцію за допомогою здійсненого розвинення із заданою кількістю членів ряду.

4. Побудувати в графічному вікні (див. п. 1) разом з наданою функцією її апроксимації із кількістю членів ряду розвинення: 2, 5, 6 та 9. Зробити висновки.

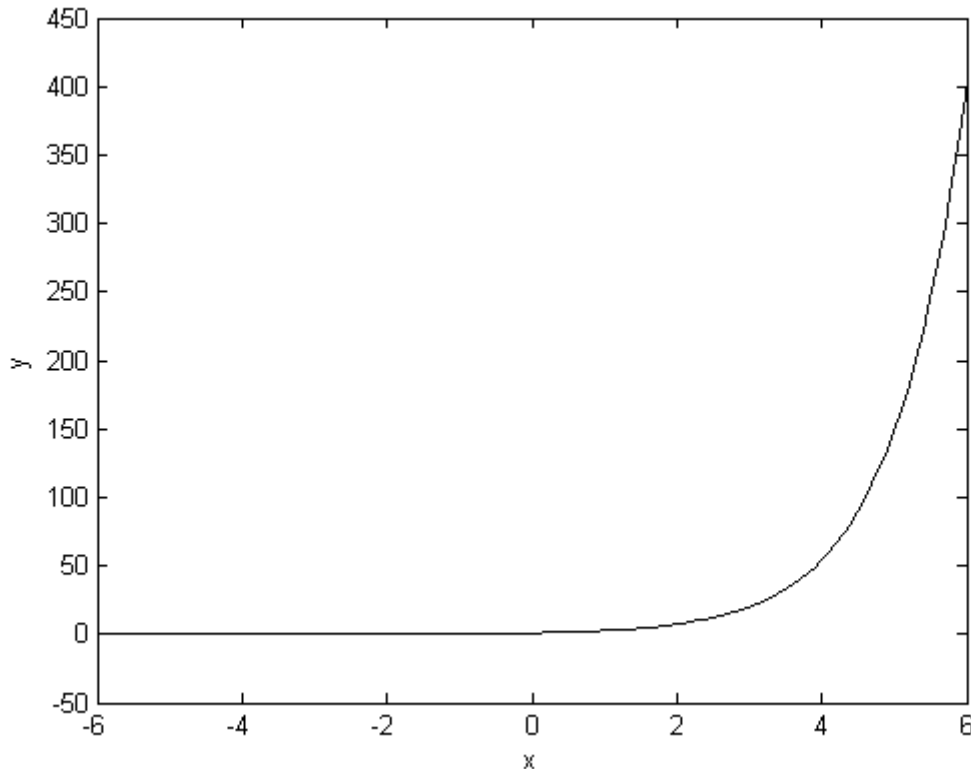


Рис. 2. Графік аналітичної функції $y = \exp(x)$

Контрольні запитання

1. Які ще задачі символічної математики Вам відомі? Наведіть приклади.
2. Що таке ряд Маклорена.
3. Як організувати цикл обробки масиву на основі конструкції `while-end`?
4. Як реалізовано механізм передаваних та повернутих параметрів в функціях MatLab?
5. Як в MatLab розрахувати математичний вираз, заданий символічно?

Робота № 3

РОЗВ'ЯЗАННЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Мета роботи: набути практичних навичок у використанні засобів MatLab для розв'язання звичайних диференціальних.

Теоретичні відомості

Диференціальні рівняння – це один з головних інструментів моделювання динамічних об'єктів та процесів. В системі MatLab реалізовано цілий ряд обчислювачів під загальним ім'ям `ode` (ordinary differential equations) [13, 14]. Функції цього ряду призначено для розв'язання звичайних диференціальних рівнянь, поданих у формі Коші:

$$\frac{dy}{d\tau} = \text{fun}(\tau, y), \quad y = y(\tau), \quad y(\tau_0) = y_0. \quad (3.1)$$

Обчислювачі різними методами визначають шукану функцію y , яка задовольняє заданому рівнянню (3.1) та початковій умові y_0 . В якості незалежної змінної зазвичай виступає час τ , с. Зокрема, найбільш вживаним є метод Рунге-Кутти 4-го та 5-го порядку точності, який можна застосувати за допомогою функції `ode45`.

Слід відзначити, що ряд функцій `ode` потребує задання функції правої частини через покажчик на неї, або шляхом вказання її імені у символній формі, наприклад, `'fun'`. При цьому сама $\text{fun}(\tau, y)$, повинна бути попередньо визначена, як функція користувача від обох аргументів, навіть якщо від y вона не залежить.

Розглянемо принципи числового інтегрування звичайних диференціальних рівнянь на прикладі метода Ейлера, який має 1-й порядок точності і в системі MatLab не реалізований:

$$y_{i+1} = y_i + (\tau_{i+1} - \tau_i) \cdot \text{fun}(\tau_i, y_i), \quad \text{при } i = 0, 1, 2, \dots, N. \quad (3.2)$$

Для того, щоб в алгоритмі можна було використовувати довільну праву частину $\text{fun}(\tau, y)$, при його програмуванні слід скористатися

базовою функцією `feval`, яка розраховує функцію, задану опосередковано, із вказаним списком параметрів.

Завдання на роботу

Розв'язати диференціальне рівняння 1-го порядку, надане викладачем, при вказаних початкових умовах, наприклад,

$$\text{рівняння } \frac{dy}{d\tau} = 1 - e^{-\tau} \text{ на інтервалі } 0 \leq \tau \leq 1 \text{ при } y_0 = 1.$$

Для цього:

1. За можливості, знайти загальний розв'язок наданого рівняння в аналітичній формі (для цього можна скористатися відповідним засобом символьних розрахунків бібліотеки SMT, див. роботу № 2).

2. Задати функцію правої частини, як функцію користувача `fun`.

3. Створити функцію користувача `odee`, що розв'язує задачу Коші за алгоритмом (3.2) при заданому N .

4. Розв'язати надане рівняння методом Рунге-Кутти та методом Ейлера для кількості підінтервалів N : 5, 10, 30 та 60.

5. Порівняти результати аналітичного та числових розв'язків шляхом побудови знайдених функцій в одному графічному вікні. Зробити висновки.

Контрольні запитання

1. Що таке покажчик? Як в мові програмування MatLab створити покажчик на функцію (`handle`).

2. Сформулюйте задачу Коші. Що таке початкові умови.

3. Як розв'язати задачу Коші в символьній формі?

4. Які числові методи розв'язання задачі Коші Вам відомі?

5. Що таке порядок точності методу інтегрування?

Робота № 4

РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ В ЧАСТИННИХ ПОХІДНИХ

Мета роботи: набути практичних навичок у використанні засобів бібліотеки PDET для розв'язання диференціальних рівнянь в частинних похідних.

Теоретичні відомості

Для моделювання розподілених динамічних об'єктів використовуються диференціальні рівняння в частинних похідних. В системі MatLab немає базових засобів для розв'язання задач цього класу, але існує додаткова бібліотека Partial Differential Equation Toolbox, яка дозволяє розв'язувати та досліджувати стаціонарні та динамічні двовимірні диференціальні рівняння методом скінченних елементів. Функції та інструменти PDET призначено для розв'язання pde як у загальній формі, так і у термінах прикладних задач, таких як: задачі механіки суцільного середовища, моделювання електричних та магнітних полів, задачі тепломасопереносу і дифузії тощо [9, 15].

Слід відзначити, що бібліотеку PDET розроблено компанією Comsol Inc, відомим виробником програмного продукту для скінченноелементного аналізу Comsol Mutliphysics, який до 2-ї версії називався FemLab та працював лише у середовищі MatLab.

Деякі можливості бібліотеки PDET розглянемо на прикладі пласкої задачі теплопровідності (heat transfer), яка описується системою рівнянь:

$$\rho c(T) \frac{\partial T}{\partial \tau} - \operatorname{div}(\lambda(T) \cdot \operatorname{grad} T) = Q(T, \tau), \quad (4.1)$$

де T – шукана функція (температура, K), що визначена на просторових координатах x, y, m ; c – питома теплоємність, Дж/(кг·K); ρ – густина, кг/м³;

λ – теплопровідність, Вт/(м·К); Q – об’ємна густина внутрішнього джерела теплоти, Вт/м³.

Рівняння виду (3.2) називається параболічним і описує нестационарну задачу. Для моделювання стаціонарної задачі використовується еліптична форма рівняння, що не містить динамічну складову.

Початкові умовами задачі:

$$T(x, y, \tau_0) = T_0, \quad (4.2)$$

де T_0 – початкова температура.

Граничні умови 1-го роду (задача Діріхле) також називають головними або суттєвими:

$$T \Big|_{\Gamma_1} = \theta(\tau), \quad (4.3)$$

де $\theta(\tau)$ – закон зміни температури на границі, К.

Граничні умови 2-го роду (задача Неймана) або природні граничні умови:

$$-\lambda \frac{\partial T}{\partial \mathbf{n}} \Big|_{\Gamma_2} = q(\tau), \quad (4.4)$$

де $q(\tau)$ – закон зміни густини теплового потоку на границі, Вт/м²; \mathbf{n} – зовнішня нормаль до границі; $q(\tau) = 0$ – відповідає умові симетрії або описує ідеальну теплоізоляцію на границі.

Граничні умови 3-го роду (задача Робена) або змішані граничні умови:

$$-\lambda \frac{\partial T}{\partial \mathbf{n}} \Big|_{\Gamma_3} = \alpha(T - T_e), \quad (4.5)$$

де α – коефіцієнт тепловіддачі, Вт/(м²·К); T_e – температура зовнішнього середовища. В теорії теплопровідності ці умови задають закон конвективного теплообміну на границі (закон Ньютона-Ріхмана).

Граничні умови 4-го роду задають рівність температур та теплових потоків для кожної точки дотичних границь:

$$T \Big|_{\Gamma_+} = T \Big|_{\Gamma_-}; \quad -\lambda \frac{\partial T}{\partial \mathbf{n}} \Big|_{\Gamma_+} = -\lambda \frac{\partial T}{\partial \mathbf{n}} \Big|_{\Gamma_-}. \quad (4.6)$$

Для формулювання, розв'язання та візуалізації pde-задач бібліотека PDET надає велику кількість функцій, більшу частину яких інтегровано у інтерактивний інструмент візуального моделювання `pdetool`.

Алгоритм розв'язання задачі за допомогою інструменту `pdetool`:

1. У вікні графічного редактора створити геометричну модель об'єкта моделювання (Draw Mode).
2. Задати граничні умови задачі (Boundary Mode) у відповідності з (4.3) .. (4.6).
3. Задати диференціальні рівняння (PDE Mode) для кожної просторової області моделі (4.1).
4. Розбити просторові області на скінченні елементи (Mesh Mode).
5. Виконати розрахунок (Solve PDE). Для динамічної задачі попередньо необхідно задати початкові умови (4.2) та часовий діапазон спостереження.
6. Відобразити результати моделювання у вигляді графіків обраних типів та бажаної кольорової гами (Plot Solution).
7. За необхідності, повернутися до п. 1 .. 4 і повторити розрахунок.

Завдання на роботу

1. Дослідити в динаміці температурне поле струмоznімальної графітової щітки на мідній основі. Центральний переріз щітки зображено на рис. 3. Вважати, що щітка розігрівається в області контакту з рухомим дротом з постійною температурою 380 °С. На нижній поверхні мідної основи щітки задати умови ідеальної теплоізоляції.

На інших бокових поверхнях щітки задати умови конвективного теплообміну ($\alpha = 6 \text{ Вт}/(\text{м}^2 \cdot \text{К})$) з оточуючим повітрям, яке має температуру 20 °С. Властивості матеріалів просторових областей наведено в табл. 1.

Матеріал	ρ , кг/м ³	c , Дж/(кг·К)	λ , Вт/(м·К)
Мідь	8930	385	390
Графіт	1410	840	93

2. Задати на всіх бокових поверхнях (крім області контакту з дротом) умови ідеальної теплоізоляції та повторити розрахунок.

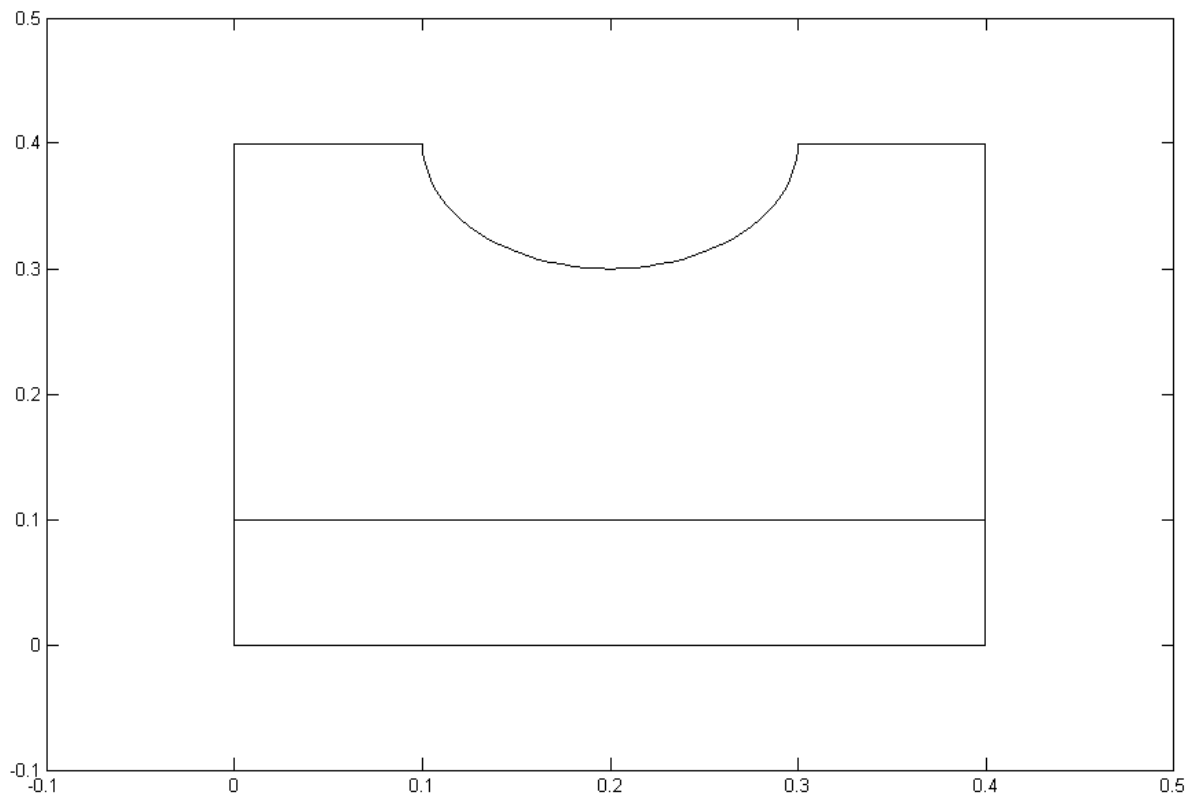


Рис. 3. Геометрична модель задачі

Контрольні запитання

1. Від чого залежить розмірність розподіленої задачі? Наведіть приклади задач різної розмірності.

2. В чому полягає метод скінченних елементів розв'язання диференціальних рівнянь в частинних похідних? Які ще методи розв'язання Вам відомі?

3. Що таке крайові умови? Які види граничних умов Вам відомі?

4. Як задати граничні умови 4-го роду в моделі (інструмент `pdetool`)?

5. Як відобразити поточне значення шуканої функції для будь-якої точки просторових областей моделі (інструмент `pdetool`).

Робота № 5

ДОСЛІДЖЕННЯ LTI-ОБ'ЄКТІВ В ЧАСОВІЙ ОБЛАСТІ

Мета роботи: набути практичних навичок у використанні засобів бібліотеки CST для дослідження лінійних стаціонарних об'єктів в часовій області.

Теоретичні відомості

Бібліотека Control System Toolbox [2, 4] була першим пакетом прикладних програм, що розширює можливості системи MatLab. CST призначена для моделювання і дослідження одновимірних (SISO) та багатовимірних (MIMO), безперервних (continuous-time) та дискретних (discrete-time) лінійних стаціонарних (linear time invariant) об'єктів і є базовою при вивченні класичної теорії автоматичного керування. При цьому модель об'єкта може бути подана у вигляді дробово-раціональної передатної функції (transfer function):

$$W(s) = \frac{b_0 s^{Nb} + \dots + b_{Nb-1} s + b_{Nb}}{a_0 s^{Na} + \dots + a_{Na-1} s + a_{Na}}, \quad (5.1)$$

або факторизованої передатної функції (zero-pole-gain):

$$W(s) = K \frac{(s - c_1)(s - c_2) \dots (s - c_{Nb})}{(s - s_1)(s - s_2) \dots (s - s_{Na})}, \quad (5.2)$$

де s_1, s_2, \dots, s_{Na} – полюси, c_1, c_2, \dots, c_{Nb} – нулі функції, а K – узагальнений коефіцієнт передачі.

Крім того модель об'єкта можна розглядати у просторі станів (state-space), який використовується у сучасній теорії автоматичного керування:

$$\begin{aligned}\frac{d\mathbf{x}(\tau)}{d\tau} &= \mathbf{A}\mathbf{x}(\tau) + \mathbf{B}\mathbf{u}(\tau); \\ \mathbf{y}(\tau) &= \mathbf{C}\mathbf{x}(\tau) + \mathbf{D}\mathbf{u}(\tau),\end{aligned}\tag{5.3}$$

де \mathbf{A} , \mathbf{B} , \mathbf{C} та \mathbf{D} – матричні оператори; \mathbf{x} , \mathbf{u} та \mathbf{y} – вектори сигналів системи: \mathbf{x} – вектор стану; \mathbf{y} – вектор вимірювань; \mathbf{u} – вектор керування.

Подібні форми представлення існують і для дискретних моделей.

В CST реалізовано об'єктний клас змінних – lti-об'єкт, що містить всю необхідну інформацію про модель. Для конкретних задач використовується не сам lti-об'єкт, а його успадковані типи, що відповідають розглянутим формам моделей. Створити lti-об'єкт можна за допомогою функцій: `tf` – у формі (5.1), `zpk` – у формі (5.2) та `ss` – у формі (5.3). Форму lti-об'єкта можна довільно змінювати за допомогою тих же функцій.

Існує можливість (за допомогою відповідних функцій) створювати `tf`- та `zpk`- об'єкти у символічному виді, взявши змінну передатної функції у апострофи, наприклад: $s = \text{zpk}('s')$.

Так само легко можна переводити безперервну модель в дискретну форму та навпаки. Для цього використовуються функції `c2d` та `d2c` відповідно. Крім того існує можливість змінювати період дискретизації (sampling time) дискретної моделі за допомогою функції `d2d`.

Для дослідження lti-об'єктів у часовій області найчастіше використовують функції: `pole` та `zero` для розрахунку полюсів і нулів моделі відповідно, та функцію `pzmap` для побудови діаграми полюсів і нулів, яка має допоміжні функції `sgrid` та `zgrid` для накладення сітки комплексної площини. Крім того часто використовуються функції: `step` – для побудови перехідної характеристики; та `impulse` – для побудови імпульсної перехідної характеристики. Для спостереження реакції моделі на довільний вхідний сигнал використовується функція `lsim`.

Завдання на роботу

1. Створити lti-змінні, що описують безперервні передатні функції, надані викладачем, наприклад, такі:

$$W_1 = \frac{3}{3p^2 + 4p + 1}; W_2 = \frac{4(p + 1)}{(p + 4)(p + 2)}.$$

2. Перетворити створені lti-об'єкти у всі доступні форми.
3. Сформувані дискретні аналоги створених моделей із періодом дискретизації 2.
4. Зменшити період дискретизації отриманих моделей до 1.
5. Визначити полюси і нулі для всіх створених lti-об'єктів.
5. Побудувати діаграми полюси і нулі, а також перехідні та імпульсні перехідні характеристики для всіх створених lti-об'єктів.

Контрольні запитання

1. Що таке лінійний стаціонарний об'єкт?
2. Що таке передатна функція? Наведіть опис моделей (5.1) та (5.2) у дискретній формі.
3. Що таке полюси та нулі об'єкта? Сформулюйте умову стійкості лінійного стаціонарного об'єкта.
4. Що таке перехідна та імпульсна перехідна характеристика?
5. Що таке мінімальна реалізація моделі динамічного об'єкта?

Робота № 6

ДОСЛІДЖЕННЯ LTI-ОБ'ЄКТІВ В ЧАСТОТНІЙ ОБЛАСТІ

Мета роботи: набути практичних навичок у використанні базових засобів MatLab та засобів бібліотеки CST для дослідження лінійних стаціонарних об'єктів в частотній області.

Теоретичні відомості

В бібліотеці CST реалізована непараметрична форма lti-моделі, яка становить собою послідовність комплексних частотних відгуків об'єкта на задану кількість фіксованих частот. Для створення таких моделей або перетворення на непараметричну форму інших типів lti-моделей використовується функція `frd`, яка потребує задання вектора частот. Для розгляду властивостей об'єкта в широкому частотному діапазоні, як правило, використовують логарифмічно-розподілений вектор частот (базова функція `logspace`) [2, 4].

Frd-модель не може бути перетворена на іншу форму, а її аналіз обмежений частотними методами.

В загальному випадку частотна характеристика динамічного об'єкта є комплексною функцією частоти і може бути зображена у вигляді:

$$G(j\omega) = G_{\text{Re}}(\omega) + jG_{\text{Im}}(\omega), \quad (6.1)$$

де $G_{\text{Re}}(\omega)$ та $G_{\text{Im}}(\omega)$ – дійсна та уявна частини функції $G(j\omega)$ відповідно.

Частотну характеристику можна також подати в полярній формі:

$$G(j\omega) = |G(j\omega)| \cdot \exp(j\varphi(\omega)), \quad (6.2)$$

де модуль $|G(j\omega)|$ називається амплітудною характеристикою, а аргумент $\varphi(\omega)$ – фазовою характеристикою об'єкта.

Для отримання доступу до комплексного частотного відгуку frd-моделі можна скористатися функцією `frdata`, яка повертає тривимірний масив, що містить відгуки для кожного з каналів багатовимірної (MIMO) моделі. Якщо модель об'єднана (SISO), комплексний масив містить невикористані розмірності, які можна скоротити базовою функцією `squeeze`.

З отриманими комплексними даними можна працювати за допомогою відповідних базових функцій MatLab. Зокрема, для визначення дійсної та уявної частини відгуку (6.1) можна скористатись функціями `real`

та `imag`, а для визначення модуля та аргумента (6.2) – функціями `abs` та `angle` відповідно.

Бібліотека CST має багато засобів для дослідження lti-об'єктів в частотній області. До найбільш вживаних належать функції: побудови діаграм Боде `bode` (тоб-то амплітудних та фазових логарифмічних частотних характеристик) та побудови годографа Найквіста `nyquist` (амплітудно-фазової частотної характеристики).

Крім того бібліотека CST містить інструмент візуалізації `ltiview`, яки інтегрує всі функції пакета для побудови характеристик (див. роботу № 5).

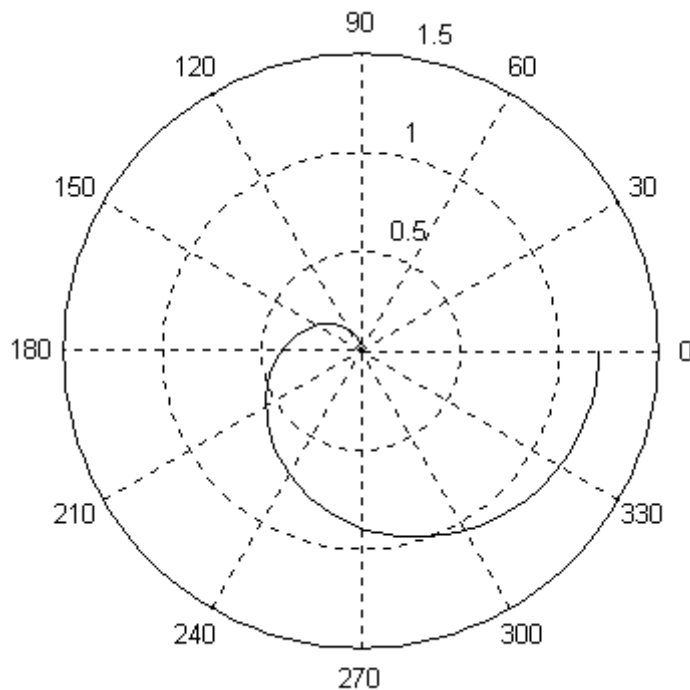


Рис. 4. Годограф АФХ в полярних координатах

Завдання на роботу

1. Для створених в роботі № 5 безперервних lti-моделей побудувати діаграми Боде і зробити висновки щодо власних частот моделюємих динамічних об'єктів.

2. Створити функції користувача, що будують годографи АФХ в декартовій та полярній системах координат (наприклад, див. рис. 4).

3. Для розглянутих безперервних lti-моделей побудувати годографи АФХ за допомогою створених функцій користувача та засобами бібліотеки CST. Порівняти результати.

Контрольні запитання

1. Які ще засоби бібліотеки CST для дослідження lti-об'єктів в частотній області Вам відомі?
2. Що таке частотна характеристика динамічного об'єкта?
3. Як задати значення частот, що відповідають 0 та ∞ у функції logspace.
4. Як побудувати графік в полярній системі координат?
5. Як визначити коефіцієнт підсилення та порядок аперіодичної ланки за її АФХ?

Робота № 7

ВЛАСТИВОСТІ LTI-ОБ'ЄКТІВ

Мета роботи: набути практичних навичок у виконанні операції із lti-змінними та отримати уявлення про принципи роботи з властивостями об'єктних змінних MatLab.

Теоретичні відомості

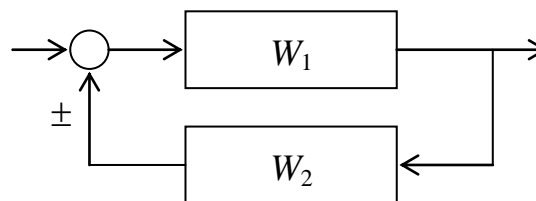


Рис. 5. Система із зворотним зв'язком

Над змінними типу lti-об'єкт можна виконувати арифметичні операції, за допомогою яких здійснюється масштабування та з'єднання динамічних ланок за різними схемами (послідовне, паралельне, зустрічно-паралельне тощо), хоча для цих дій в бібліотеці CST передбачені і

спеціалізовані функції [2, 4]. Наприклад, для організації системи із зворотним зв'язком, що зображена на рис. 5, можна виконати операцію:

$$W = \frac{W_1}{1 \mp W_1 \times W_2},$$

або застосувати функцію **feedback**.

Якщо в операції з динамічними ланками беруть участь lti-змінні різних форм, результат зводиться до форми з найвищим пріоритетом у відповідності з правилом: frd > ss > zpk > tf.

З lti-змінних можна формувати масиви стандартними для MatLab засобами (операція конкатенації), що дозволяє утворювати багатовимірні (MIMO) моделі об'єктів з моделей меншої розмірності, зокрема, одновимірних (SISO) моделей. Багатовимірні моделі також можна створювати безпосередньо при оголошенні lti-змінних за допомогою відповідних функцій (див. роботу № 4), але цей спосіб менш наочний.

Для довільного з'єднання ланок можна скористатися функцією **connect**, яка потребує попереднього створення діагонально-блочної структури ланок функцією **append**, опису зовнішніх входів та виходів системи у вигляді відповідних векторів та задання матриці внутрішніх перехресних зв'язків. Докладніше про цей метод можна дізнатись за допомогою команди **help connect**.

Як і будь-яка об'єктна змінна MatLab, lti-змінна має властивості, доступ до яких здійснюється за допомогою функцій **set** та **get**. Імена властивостей lti-змінних можна скорочувати до кількості символів, необхідної для однозначного їх розпізнання.

Доступ до властивостей lti-змінної також можна здійснювати в традиційний спосіб – прямим зверненням до відповідного поля, ім'я якого вказується через крапку (символ “.”) після імені змінної.

Для зображення транспортного запізнювання динамічного об'єкта Іті-змінні мають декілька властивостей, що описують всі канали багатовимірної моделі з N_u входами та N_y виходами:

$$InputDelay = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_{N_u}], \quad OutputDelay = [\beta_1 \quad \beta_2 \quad \dots \quad \beta_{N_y}],$$

$$ioDelayMatrix = \begin{bmatrix} \alpha_1 + \beta_1 & \alpha_2 + \beta_1 & \dots & \alpha_{N_u} + \beta_1 \\ \alpha_1 + \beta_2 & \alpha_2 + \beta_2 & \dots & \alpha_{N_u} + \beta_2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 + \beta_{N_y} & \alpha_2 + \beta_{N_y} & \dots & \alpha_{N_u} + \beta_{N_y} \end{bmatrix}.$$

Якщо модель має однакові запізнювання у всіх каналах, то таке запізнювання можна задати у скалярному вигляді, привласнивши його будь-якій з вказаних властивостей.

Всі властивості моделі також можна задавати безпосередньо при оголошенні Іті-змінних за допомогою відповідних функцій створення (див. роботу № 5).

Слід відзначити, що зміна властивості 'Ts' (sampling time) вказаними способами не призводить до дійсної зміни періода дискретизації Іті-об'єкта. Цю операцію слід виконувати за допомогою функцій c2d, d2c та d2d (див. роботу № 5), або функції delay2z, яка дозволяє врахувати запізнювання дискретного об'єкта зміною цього характеристичного полінома.

Завдання на роботу

1. Організувати всі види з'єднання безперервних динамічних ланок, моделі яких W_1 та W_2 було оголошено при виконання роботи № 5. Змінити тип змінної вказаних передатних функцій.

2. Сформувати моделі багатовимірних систем: з одним входом і двома виходами та двома входами і одним виходом, канали яких описуються моделями W_1 та W_2 . Задати імена входам та виходам

створених багатовимірних систем. Побудувати перехідні характеристики систем.

3. Задати вказані викладачем транспортні запізнювання моделям W_1 та W_2 та побудувати їх перехідні характеристики. Повторити п. 2 та порівняти результати.

4. Задати вказані викладачем транспортні запізнювання для дискретних аналогів моделей W_1 та W_2 та побудувати їх перехідні характеристики. Змінити типи змінних вказаних передатних функцій та порівняти результати.

5. Для дискретної моделі W_2 виконати зміну характеристичного полінома за рахунок запізнювання, побудувати її перехідну характеристику та порівняти результати.

Контрольні запитання

1. За допомогою яких функцій CST можна організувати основні типи з'єднань динамічних ланок?

2. Як визначити зведену передатну функцію системи динамічних ланок, з'єднаних довільно?

3. Як визначити перелік властивостей l_{ti} -змінної певної форми?

4. Як подати запис моделі дискретного об'єкта в прямих та зворотних поліномах?

5. Чим відрізняється вхідне запізнювання від вихідного для l_{ti} -зоб'єкта в tf та ss формах?

Робота № 8

ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ

Мета роботи: набути практичних навичок у параметричній ідентифікації динамічних об'єктів засобами бібліотеки SIT.

Теоретичні відомості

Додаткову бібліотеку функцій System Identification Toolbox [3, 12] було розроблено під керівництвом проф. Л. Льюнга, всесвітньо відомого фахівця в області ідентифікації та адаптивного керування. SIT містить велику кількість засобів для параметричної та непараметричної ідентифікації лінійних динамічних об'єктів різними методами, а також засоби дослідження та перевірки адекватності отриманих оціночних моделей.

Приблизний алгоритм параметричної ідентифікації може бути таким:

1) Планування експерименту та формування масивів даних шляхом вимірювань вхідних та вихідних сигналів об'єкта, що підлягає ідентифікації.

2) Можлива попередня обробка отриманих даних (наприклад, фільтрація чи видалення тренду з даних).

3) Визначання структури моделі (з набору моделей-кандидатів) в межах якої буде визначена модель.

4) Розрахунок найкращої моделі обраної структури відповідно до вимірюваних даних та заданого критерію.

5) Дослідження властивостей отриманої моделі та перевірка її адекватності.

Якщо модель задовольняє поставленим умовам, то процес ідентифікації припиняється; інакше – повторюється 3-й етап для випробування іншої структури моделі. Можливо також треба змінити методи оцінювання (етап 4) або повторно сформувати масиви даних (етапи 1 і 2).

В даній роботі оцінку ми будемо шукати в класі дискретних моделей в просторі станів, та поліноміальних моделей, загальну форму яких можна подати наступним чином:

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k - N_k) + \frac{C(q)}{D(q)}e(k); \quad (8.1)$$

де $N_k \geq 1$ – запізнювання в системі; $y(k)$, $u(k)$ – вхідні та вихідні змінні об’єкта; A , B , C , D , F – поліноми, визначені за допомогою оператора зсуву назад q (див. роботу № 7), які описують динаміку об’єкта та фільтра випедкового сигналу у вигляді білого шуму $e(k)$.

На практиці використовується не загальна форма (8.1), а деякі її часткові форми, отримані скасуванням відповідних поліномів. Наприклад, якщо прийняти $C = D = F = 1$, отримуємо так звану ARX форму (AutoRegressive with eXternal input model) – модель авторегресії із зовнішнім входом:

$$A(q)y(k) = B(q)u(k - N_k) + e(k).$$

Більшість функцій SIT інтегровано у інтерактивний інструмент візуального моделювання *ident* (див. рис. 6).

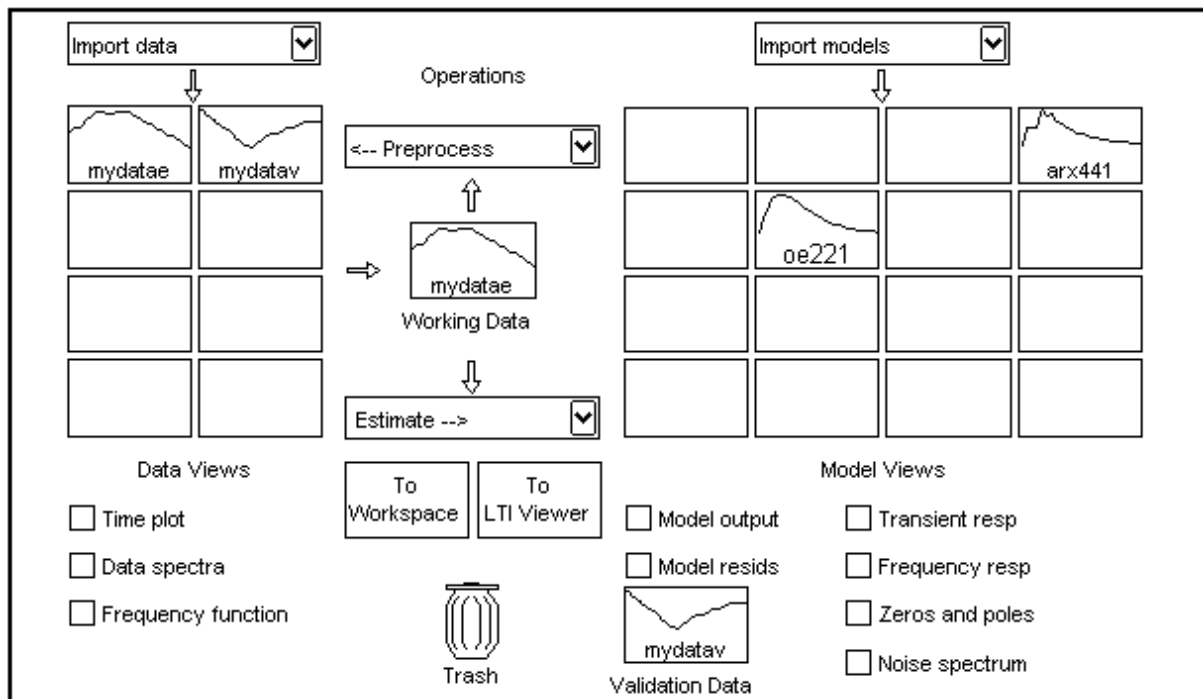


Рис. 6. Головне вікно інструменту *ident*

Введення експериментальних даних, попередньо сформовані у вигляді векторів MatLab, виконується функцією випадног списку *Import data/ Time domain data...* Імпортовані дані (розташовані в лівому

накопичувачі) можна візуалізувати засобами Data Views. Найбільш вживаною функцією візуалізації є Time plot [3].

Для того, щоб над імпортованими даними можна було виконувати дії, їх треба розташувати в області Working Data, де вони стають доступними функціям випадних списків Preprocess та Estimate. Зокрема, корисною є попередня обробка даних, яка дозволяє, наприклад, переформулювати задачу в термінах приростів Preprocess/ Remove means, та вибрати піддіапазони даних для подальшого оцінювання та перевірки адекватності Preprocess/ Select range... Отримані таким чином піддіапазони даних слід розташувати в областях Working Data та Validation Data відповідно.

Параметрична ідентифікація даних здійснюється за допомогою функції Estimate/ Linear parametric models..., яка відкриває вікно, що дозволяє задавати структуру моделей-кандидатів, їх порядки та методи оцінювання. Результатом розрахунку є оціночна модель, яка розташовується в правому накопичувачі (board) і може бути досліджена засобами Model Views.

Основним способом перевірки адекватності отриманої оціночної моделі в SIT є порівняння відгуків моделі та об'єкта на вхідну послідовність даних, що відрізняється від послідовності, яка використовувалась для оцінювання (функція Model output). Тут якість моделі визначається за величиною збігу (fit, %) між виміреним виходом об'єкта та предбаченим виходом моделі у вигляді:

$$\delta_{\text{fit}} = 100 \times \frac{1 - \|\hat{\mathbf{y}} - \mathbf{y}\|}{\|\mathbf{y} - \bar{\mathbf{y}}\|},$$

де \mathbf{y} – вектор виміреної вихідної змінної об'єкта $y(k)$; $\bar{\mathbf{y}}$ – середнє значення вектора \mathbf{y} ; $\hat{\mathbf{y}}$ – вектор вихідної змінної оціночної моделі.

Іншим, реалізованим у SIT, методом перевірки якості моделі є аналіз її остаткової похибки. Якщо модель цілком адекватно описує систему – остаткова похибка $e(k)$ повністю не залежить від входу системи $u(k)$ (тобто є ідеальним білим шумом). Таким чином про адекватність моделі можна судити по характеру автоковаріаційної функції похибки $e(k)$ та взаємної коваріаційної функції між входом моделі $u(k)$ та похибкою для заданої кількості лагів (функція Model resids).

Назадовільні моделі та набори даних можна видалити, помістивши їх в область Trash. Моделі, що визнано задовільними слід експортувати в робочій простір MatLab шляхом переміщення їх в область To Workspace. При цьому слід пам'ятати, що SIT зберігає результати параметричної ідентифікації у th-формі, яка крім коефіцієнтів моделі містить інформацію про оцінку стандартних відхилень параметрів. Для перетворення th-змінних на lti-змінні (з якими можна працювати засобами CST) в бібліотеці SIT передбачені відповідні функції (наприклад, th2tf або th2ss), але найпростіший спосіб – це використання функції поля To LTI Viewer, яка виконує потрібне перетворення та екпортує модель у ltiviewer (інструмент візуалізації CST, див. роботу № 6)

Завдання на роботу

Здійснити ідентифікації динамічного об'єкта за експериментальними даними, наданими викладачем. Для цього:

1. Імпортувати вихідні дані та підготувати їх для ідентифікації.
2. Дослідити засобами Transient resp, Zeros and poles та Model output, щонайменше, 300 моделей об'єкта, отриманих різними методами ідентифікації для всіх структур, що реалізовано у вікні Linear Parametric Models, та обрати 10 найкращих.
3. Дослідити відібрані моделі кореляційними методами (засіб Model resids) та обрати 2 найкращі.

4. Подати отримані моделі у вигляді передатних функцій в безперервному часі (tf-форма) та остаточно обривна модель найпростішої структури, для якої побудувати перехідну характеристику та діаграму полюсів і нулів.

Контрольні запитання

1. Що так ідентифікація динамічного об'єкта?
2. Які ще структури поліноміальної моделі об'єкта Вам відомі?
3. Які методи параметричної ідентифікації застосовуються у SIT?
4. Що таке кореляційна функція випадкового сигналу? Що таке часовий лаг?
5. Наведіть приклади непараметричної ідентифікації.

Робота № 9

СИНТЕЗ СИСТЕМ МЕТОДОМ КОРЕНЕВОГО ГОДОГРАФА

Мета роботи: ознайомитись з принципами настройки замкнених систем з лініними регуляторами та корегуючими ланками методом кореневого годографа засобами бібліотеки CST.

Теоретичні відомості

Кореневий годограф зображує траєкторії полюсів замкненої динамічної системи при варіюванні одного з її параметрів при фіксованому значенні всіх інших параметрів. Найчастіше метод кореневого годографа використовується для настроювання коефіцієнта зворотного зв'язку з метою бажаного розташування полюсів замкненої одновимірної системи керування.

Для роботи з кореневими годографами в бібліотеці CST передбачено функції `rlocus` та `rlocfind` [2, 4], але зручніше для цього скористатися інтерактивним інструментом `rltool`, який в нових версіях MatLab є

частиною розширеного інструменту візуального конструювання лінійних одновимірних систем *sisotool* (див. рис. 7).

Плотер корневих годографів *rltool* дозволяє в інтерактивному режимі традиційними методами (див. роботи № 5 та 6) досліджувати властивості замкненої системи при зміні коефіцієнта зворотного зв'язку.

Крім того, плотер дозволяє конструювати корегуючу ланку *C* (Compensator) шляхом розташування та пересування на комплексній площині додаткових нулів та полюсів, використовуючи відповідні засоби панелі інструментів (див. рис. 7).

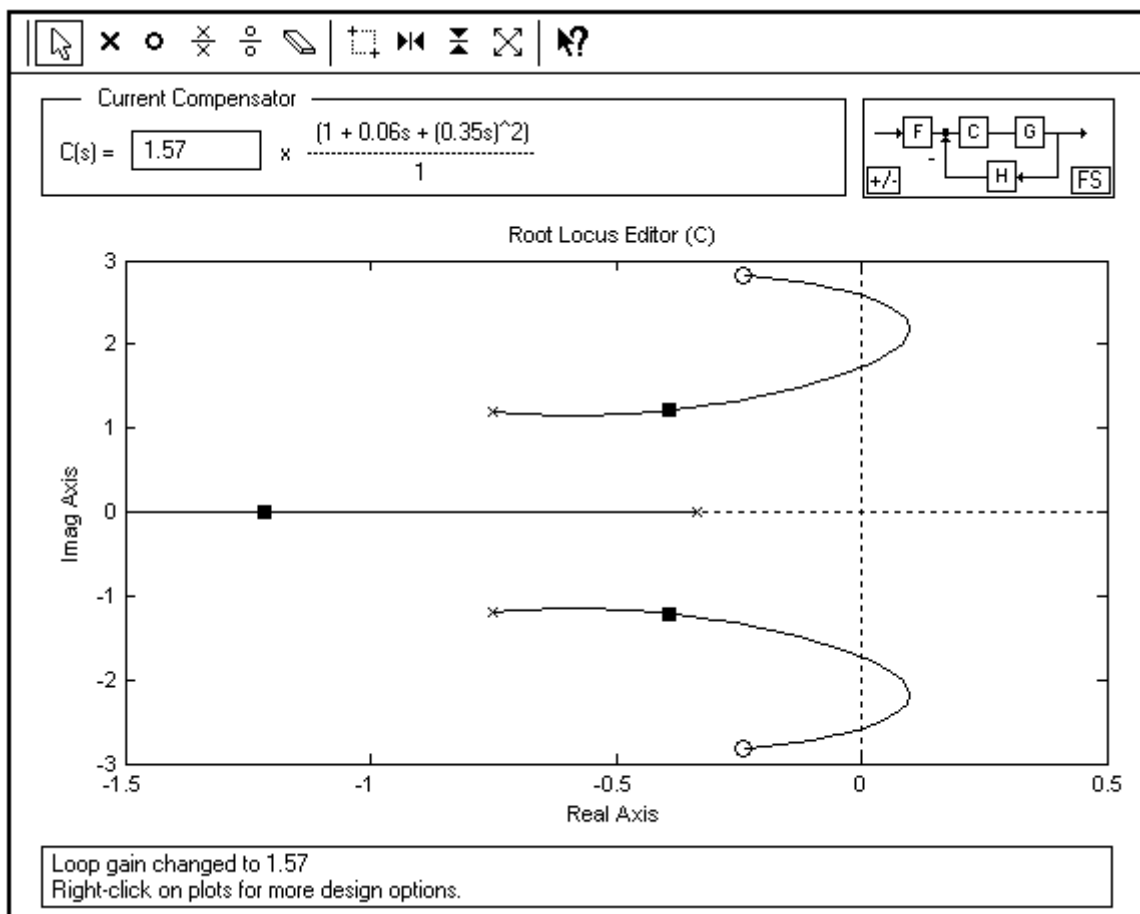


Рис. 7. Головне вікно плотера *rltool*

Загальний опис замкненої системи здійснюється в окремому вікні (див. рис. 8), яке викликається командою *File/ Import....* У цьому вікні можна вказати структуру системи, тип зворотного зв'язку, а також задати динамічні властивості окремим елементам системи: об'єкту керування *G*

(Plant), фільтру вхідного сигналу F (Prefilter) та вимірювачу вихідного сигналу H (Sensor). Динамічні властивості ланок задаються шляхом привласнення відповідному полю попередньо оголошеної l -змінної.

Результат конструювання – модель корегуючої ланки можна зберегти функцією File/ Export....

Слід відзначити, що в системі MatLab 7-ї версії інтерфейс користувача (GUI) плотера було дещо змінено.

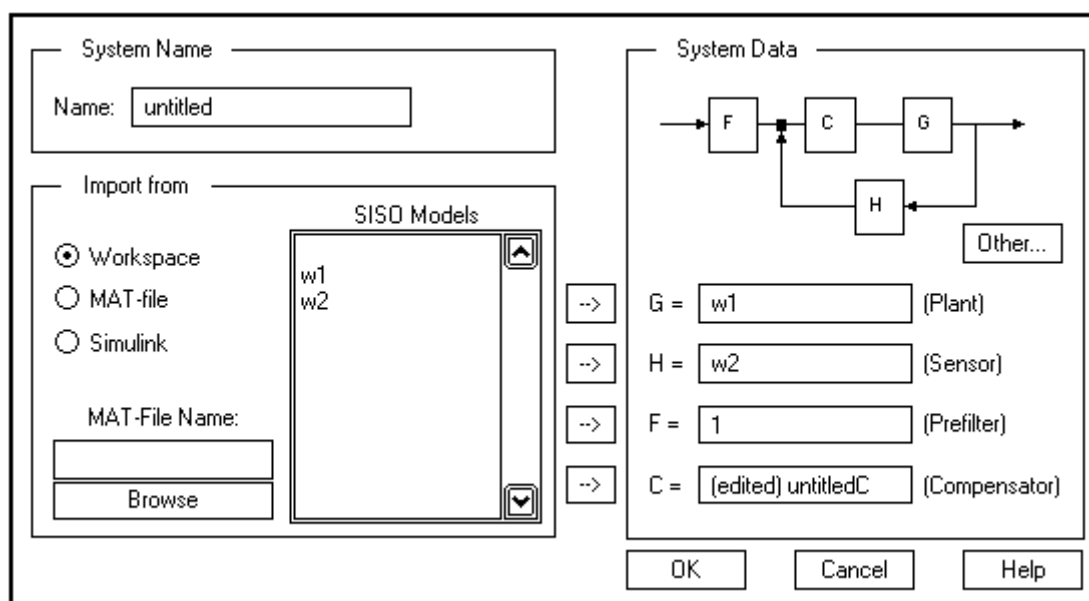


Рис. 8. Вікно формулювання задачі плотера rltool

Завдання на роботу

1. Налаштувати замкнену систему з об'єктом, модель якого отримано в роботі № 8, та П-регулятором на максимальну швидкодію.

2. Ввести в систему динамічну ланку вимірювача з передатною функцією $W_h = \frac{1}{3s+1}$ та дослідити її вплив на властивості системи.

3. Створити корегуючу ланку, що усуває негативний вплив динаміки вимірювача та повторно налаштувати систему на максимальну швидкодію.

4. Результати роботи подати у вигляді зрк-змінної та зробити висновки.

Контрольні запитання

1. Наведіть основні властивості кореневого годографа.
2. Для чого використовуються корегуючі ланки?
3. Як змінити структуру системи, що моделюється інструментом `sisotool`?
4. Як задати моделі елементам системи що моделюється інструментом `sisotool`?
5. Для чого використовується функція `rlocfind`?

Робота № 10

РОЗРАХУНОК СИСТЕМ НА ЗАДАНИЙ ПОКАЗНИК КОЛИВНОСТІ

Мета роботи: набути практичних навичок настройки замкнених систем з лінійними регуляторами на заданий показник коливності базовими засобами MatLab.

Теоретичні відомості

Обмеження на коливність власних процесів в замкненій системі можна сформулювати як вимогу, щоб годограф амплітудно-фазової частотної характеристики (див. роботу № 5) розімкненої системи $W(j\omega)$ з регулятором не перетинав області, обмеженої окружністю з координатами $(-x_m, 0)$ та радіусом r_m (М-колом), де

$$x_i = \frac{M^2}{M^2 - 1}; \quad r_i = \frac{M}{M^2 - 1}.$$

Таким чином, розрахунок системи на обраний показник коливності полягає у визначенні параметрів регулятора, які забезпечують торкання АФХ розімкненої системи з обраним М-колом.

При визначенні параметрів регулятора достатньо розглядати тільки третій квадрант годографа АФХ системи.

Якщо передатну функцію лінійного регулятора подати, як:

$$W_r(s) = k_p + \frac{1}{T_{in}s} + T_{df}s = k_p \frac{T_i T_v s^2 + T_i s + 1}{T_i s}, \quad (10.1)$$

де k_p – коефіцієнт пропорційності; T_{in} – час інтегрування; T_{df} – час диференціювання; $T_i = T_{in} \cdot k_p$ – час ізодрому; $T_v = T_{df} / k_p$ – час випередження, то в загальному випадку умові максимальної фільтрації збурень відповідає максимальне можливе значення k_p для П-регулятора та відношення k_p / T_i для ПІ- та ПІД-регуляторів. Для визначення параметрів ПІД-регулятора співвідношення T_v / T_i повинно бути в межах 0.3 ... 0.5 для об'єктів з самовирівнюванням та 0.15 для об'єктів без самовирівнювання.

Завдання на роботу

1. Створити функцію користувача `rotach`, яка будує на комплексній площині годограф розімкненої системи з ПІ-регулятором та М-коло для показника коливності 1.5, наприклад, див. рис. 9.

2. Для моделі об'єкта, отриманої в роботі № 8, визначити оптимальну настройку регулятора шляхом інтерполяції вибірки даних, сформованої з, щонайменше, 8 настрок ПІ-регулятора, які забезпечують показник коливності $M = 1.5$.

3. Побудувати перехідні характеристики системи за каналами збурення та завдання.

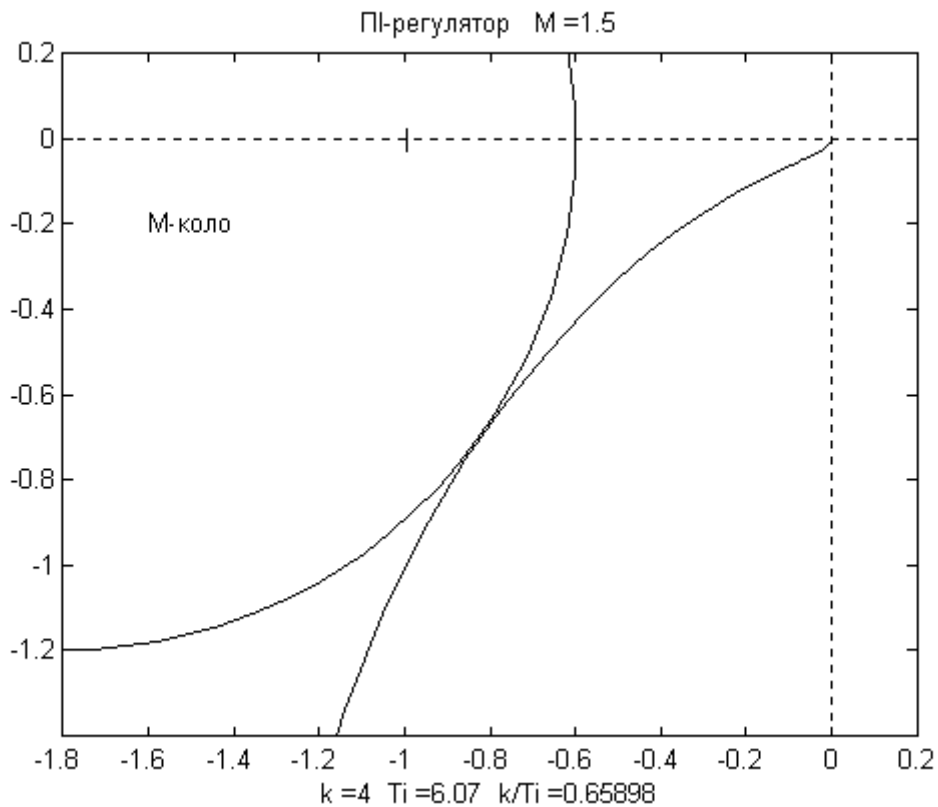


Рис. 9. Розрахунок на заданий показник коливності

Контрольні запитання

1. Як побудувати окружність в декартовій системі координат?
2. Як побудувати осі з початком координат всередині графіка?
3. Що таке коефіцієнт згасання та показник коливності?
4. Які типи лінійних регуляторів Вам відомі і як отримати їх передатні функції з загальної форми (10.1)?
5. Як залежить запас стійкості системи від показника її коливності?

Робота № 11

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ

Мета роботи: набути практичних навичок імітаційного моделювання динамічних об'єктів засобами Simulink.

Теоретичні відомості

У склад системи MatLab входить інструмент математичного моделювання лінійних та нелінійних динамічних систем Simulink [5, 6, 7, 11], який дозволяє за допомогою наборів блоків конструювати моделі систем у візуальному вигляді та здійснювати їх імітаційне моделювання.

Simulink має базові бібліотеки (наприклад, Simulink Blockset та Simulink Extras Blockset), які містять візуальні блоки, що дозволяють моделювати широкий клас об'єктів в дискретному та безперервному часі, а також блоки-генератори та приймачі сигналів, блоки, що здійснюють математичні та логічні операції над сигналами тощо.

Додаткові бібліотеки блоків розробляються для розв'язання специфічних прикладних та наукових задач і можуть бути проінстальовані окремо. Наприклад, бібліотеку Nonlinear Control Design Blockset призначено для настройки параметрів динамічних систем, що забезпечують бажану якість перехідних процесів. Користувач може створювати власні візуальні блоки та об'єднувати їх у бібліотеки.

Крім того, деякі додаткові бібліотеки функцій MatLab містять також і блоки Simulink. Наприклад, бібліотека CST містить візуальний блок LTI System, що дозволяє безпосередньо використовувати в діаграмах Simulink lti-змінні, оголошені у командному вікні MatLab.

Завдання на роботу

1. Створити імітаційну діаграму, в якій модель, отриману в роботі № 8, подано, щонайменше, 3-а різними способами. Наприклад, на рис. 10 зображено модель об'єкта у формі (5.3).

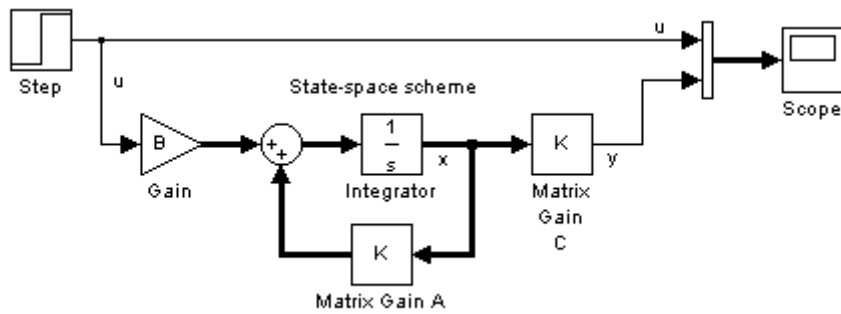


Рис. 10. Імітаційна модель об'єкта в просторі станів

2. Здійснити спостереження перехідного процесу об'єкта, як реакції на збурення (з амплітудою 1): ступінчасте, гармонійне та випадкове.
3. Експортувати результати в командне вікно та побудувати перехідні процеси базовими засобами MatLab.

Контрольні запитання

1. Що таке імітаційне моделювання?
2. Які стандартні блоки для моделювання динамічних об'єктів Вам відомі?
3. Які стандартні блоки: генератори (Sources), приймачі (Sinks) та маршрутизатори (Routing) сигналів Вам відомі?
4. Які дії з візуальними блоками, розташованими в області діаграми, можна виконувати?
5. Як здійснюється обмін даними між моделлю Simulink та командним вікном MatLab?

Робота № 12

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ЗАМКНЕНИХ СИСТЕМ

Мета роботи: набути практичних навичок дослідження замкнених динамічних систем на імітаційних моделях засобами Simulink.

Теоретичні відомості

Лінійний регулятор (див. роботу № 10) можна розглядати як звичайну динамічну ланку, модель якої можна задати традиційними засобами (див. роботу № 11). Разом з тим стандартні бібліотеки Simulink містить блоки, що дозволяють працювати з ПД-регулятором в термінах його складових (наприклад, див. рис. 10).

При зображенні в імітаційних діаграмах об'єктів із запізнюванням слід пам'ятати, що властивість *lti*-змінної *ioDelayMatrix* Simulink ігнорує, тому запізнювання слід задавати властивістю *InputDelay* або *OutputDelay*. Крім того, для зображення транспортного запізнювання можна скористатись одним з відповідних блоків стандартних бібліотек блоків. Зокрема, якщо імітаційну модель планується лінеаризувати, транспортне запізнювання слід задавати блоком *Switched transport delay*, який дозволяє виконувати апроксимацію Паде для цієї динамічної ланки [2, 11].

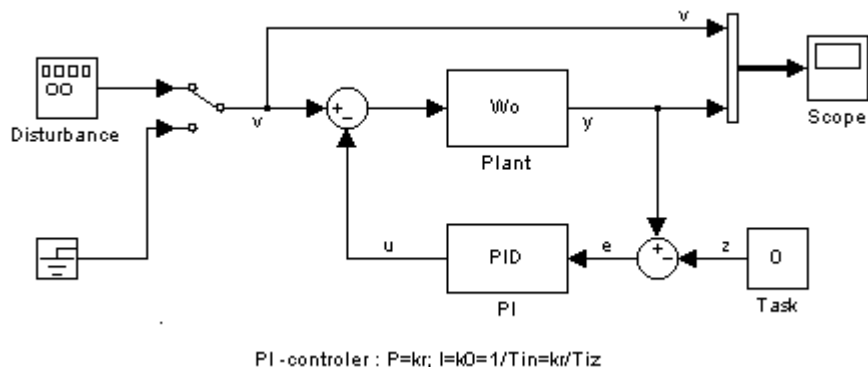


Рис. 11. Імітаційна модель замкненої системи з ПІ-регулятором

Завдання на роботу

1. Створити імітаційну модель замкненої системи з об'єктом, отриманим в роботі № 8 та ПІ-регулятором, отриманим в роботі № 10 (наприклад, див. рис. 11).

2. Дослідити реакцію системи на сигнал завдання та різних видів збурення (див. роботу № 11).

3. Внести в систему ланку транспортного запізнювання та дослідити вплив величини запізнювання на якість регулювання за наявності та відсутності сигналу збурення.

4. Наочні результати подати у вигляді графіків перехідних процесів, виконані засобами командного вікна MatLab.

Контрольні запитання

1. Як організувати відображення результатів моделювання блоком XY Graph (замість Scope)?

2. Як настроїти блок Sum для організації від'ємного зворотного зв'язку?

3. Як ще можна організувати сигнал збурення у вигляді випадкового процесу?

4. Чим відрізняється блок PID Controller від блока PID Controller (2DOF)?

5. Що так апроксимація Паде?

Контрольна робота

Мета роботи: перевірити набуті практичні навички у моделюванні динамічних об'єктів, синтезі замкнених систем та їх дослідженні на імітаційних моделях засобами MatLab.

Завдання на роботу

1. Виконати згладжування експериментальних даних, наданих викладачем.

2. Здійснити ідентифікацію безперервного лінійного стаціонарного динамічного об'єкта за підготованими експериментальними даними.

3. Розрахувати замкнену систему з отриманим об'єктом та регулятором, тип якого та метод синтезу вказано викладачем.

4. Дослідити систему методами, вказаними викладачем.

5. Створити імітаційну модель системи та дослідити вплив транспортного запізнювання та збурень, вказаного викладачем виду, на якість роботи регулятора.

Всі завдання роботи виконуються студентом індивідуально в комп'ютерному класі. Графічні результати роботи повинні бути оформленими у відповідності з рекомендаціями ЄСКД (Р 50-77-88).

Список рекомендованої літератури

1. Потемкин В.Г. Введение в MatLab / В.Г. Потемкин. – М. : Диалог-МИФИ, 2000. – 247 с.

2. Медведев В.С. Control System Toolbox. MatLab 5 для студентов / В.С. Медведев, В.Г. Потемкин. – М.: Диалг-МИФИ, 1999. – 287 с.

3. System Identification Toolbox For Use with MatLab : User's guide [Electron resource]. – Natick, MA : The MathWorks, Inc, 1997. – 274 p.

4. Control System Toolbox For Use with MatLab : User's guide [Electron resource]. – Natick, MA: The MathWorks, Inc, 1999. – 649 p.
5. Simulink. Dynamic System Simulation for MatLab : Using Simulink [Electron resource]. – Natick, MA : The MathWorks, Inc, 1999. – 605 p.
6. Дэбни Дж. Simulink 4. Секреты мастерства / Дж. Б. Дэбни, Т.Л. Харман. – М. : Бином, 2003. – 403 с.
7. Гультияев А.К. Визуальное моделирование в среде MatLab : Учебный курс / А.К. Гультияев. – СПб. : Питер, 2000. – 432 с.
8. Мартынов Н.Н. Введение в MatLab 6. / Н.Н. Мартынов. – М. : Кудиц-образ, 2002. – 352 с.
9. Ануфриев И.Е. MatLab 7 / И.Е. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова. – СПб. : БХВ-Петербург, 2005. – 1104 с.
10. Дьяконов В.П. Simulink 4. Специальный справочник / В.П. Дьяконов. – СПб. : Питер, 2002. – 528 с.
11. Дьяконов В.П. Математические пакеты расширения MatLab. Специальный справочник / В. Дьяконов, В. Круглов. – СПб. : Питер, 2001. – 480 с.
12. Дьяконов В.П. MatLab. Анализ, идентификация и моделирование систем. Специальный справочник / В. Дьяконов, В. Круглов. – СПб. : Питер, 2002. – 448 с.
13. Кетов Ю.Л. MatLab 7: программирование численных методов / Ю.Л. Кетов, А.Ю. Кетов, М.М. Шульц. – СПб. : БХВ-Петербург, 2005. – 752 с.
14. Мэтьюз Дж. Численные методы. Использование MatLab / Джон Г. Мэтьюз, Кертис Д. Финк. – М. : “Вильямс”, 2001. – 720 с.
15. Мартынов Н.Н. MatLab 5.x. Вычисления, визуализация, программирование / Н.Н. Мартынов, А.П. Иванов. – М. : Кудиц-образ, 2000. – 336 с.

Додаток А
ПРИКЛАД РОЗВ'ЯЗАННЯ ЗВИЧАЙНОГО
ДИФЕРЕНЦІАЛЬНОГО РІВНЯННЯ

Розв'язати диференціальне рівняння:

$$\frac{dy}{dt} = 1 - e^{-t} \text{ на інтервалі } 0 \leq t \leq 1 \text{ при початковій умові } y_0 = 1.$$

Розв'язання в аналітичній формі

загальний розв'язок:

```
>> dsolve('Dy=1-exp(-t)')           % D = d/dt
ans = C2 + t + 1/exp(t)
```

частинний розв'язок для $y_0 = 1$:

```
>> dsolve('Dy=1-exp(-t)', 'y(0)=1')
ans = t + 1/exp(t)
```

Розв'язання числовими методами

метод Ейлера:

```
function [te,ye]=odee(f,x,y0,n)
% Розв'язання диф. рівнянь методом Ейлера
% f - функція правої частини
% x - інтервал інтегрування
% y0 - початкова умова
% n - кількість підінтервалів інтегрування
te(1)=x(1);
h=(x(2)-x(1))/n;
ye(1)=y0;
for i=1:n
    te(i+1)=te(i)+h;
    ye(i+1)=ye(i)+h*feval(f,te(i),ye(i));
end
```

функція правої частини:

```
function dy=fun(t,y)
% Функція правої частини диф.рівняння
dy=1-exp(-t);
```

Розв'язок задачі

```
>> x1=0:0.1:1;
>> y1=x+exp(-x);
>> [x2,y2]=ode45('fun',[0 1],1);
>> [x3,y3]=odee('fun',[0 1],1,5);
>> [x4,y4]=odee('fun',[0 1],1,10);
>> plot(x1,y1,x2,y2,x3,y3,x4,y4)
>> legend('Analytic','Runge-Kutta','Euler 5','Euler 10',0)
>> xlabel('t'); ylabel('y')
```

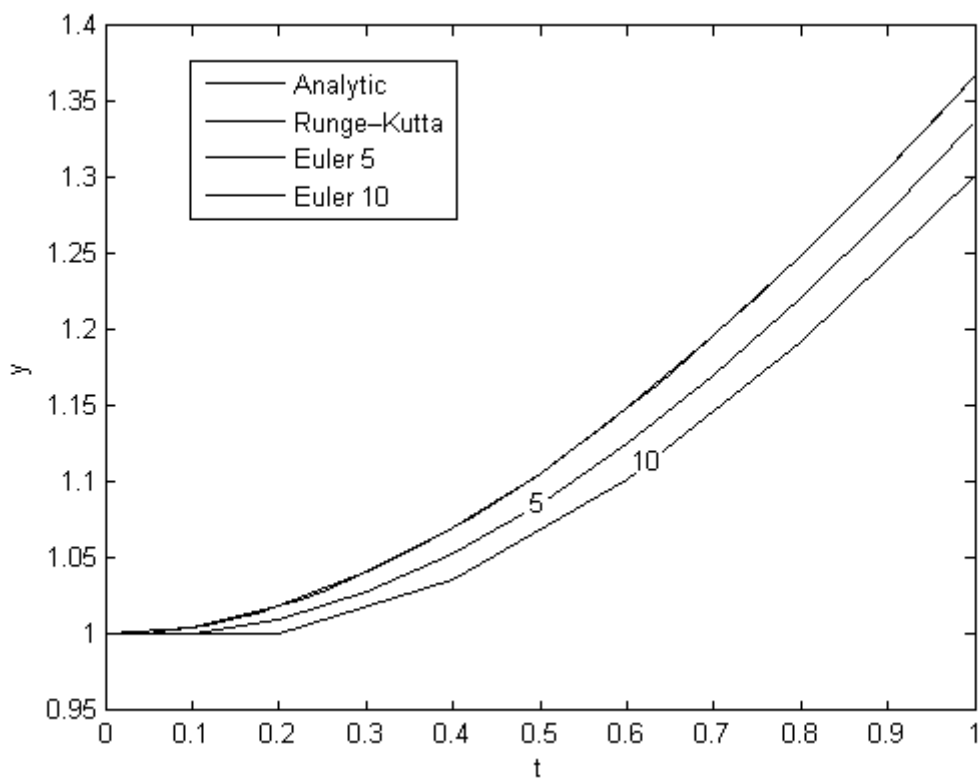


Рис. 12. Розв'язок на інтервалі $t = [0 1]$

Додаток Б
РОЗРАХУНОК СИСТЕМИ З ПІ-РЕГУЛЯТОРОМ
НА ЗАДАНИЙ ПОКАЗНИК КОЛИВНОСТІ

```
function Wrg=rotach(Wo,k,Ti,M)
% Побудова М-кола та АФХ розімкненої системи з регулятором
% Wrg=rotach(Wo,k,Ti,M)
% Wo - передатна функція SISO об'єкта;
% k - коефіцієнт пропорційності регулятора;
% Ti - час ізодрому;
% М - показник коливності системи (М=1.3...1.6).
% Wrg - повертається передатна функція регулятора

% ПІ-регулятор
Wrg=k*tf([Ti 1],[Ti 0]);

% Формування розімкненої системи
W=minreal(Wo*Wrg);

% Визначення вектора частотного відгуку системи W
fr=logspace(-2,1,80);
resp=squeeze(frd(frd(W,fr)));
% Або resp=squeeze(freqresp(W,fr));

% Побудова годографа
plot(resp); hold on

% Побудова М-кола
mx=M^2/(M^2-1); % Абсциса
mr=M/(M^2-1); % Радіус
circus(-mx,0,mr)

% Оформлення графіка
axis equal % Однаковий масштаб осей
axis([-mx 0.2 -mr-0.2 0.2])
```

```

% Побудова координатних осей
line([-mx;0.2],[0;0],'LineStyle',':','Color','black');
line([0;0],[-mr-0.2;0.2],'LineStyle',':','Color','black');
hold off

function circus(x,y,r)          % Приватна функція
% Побудова окружності радіуса r з центром в точці x,y
t=pi*(0:.02:2);
plot(x+r*cos(t),y+r*sin(t),'r') % Червоний колір

```

Виконання:

```
» rotach(Wo,3,8,0,1.5);
```

Transfer function:

39.13 s + 5.207

7.514 s

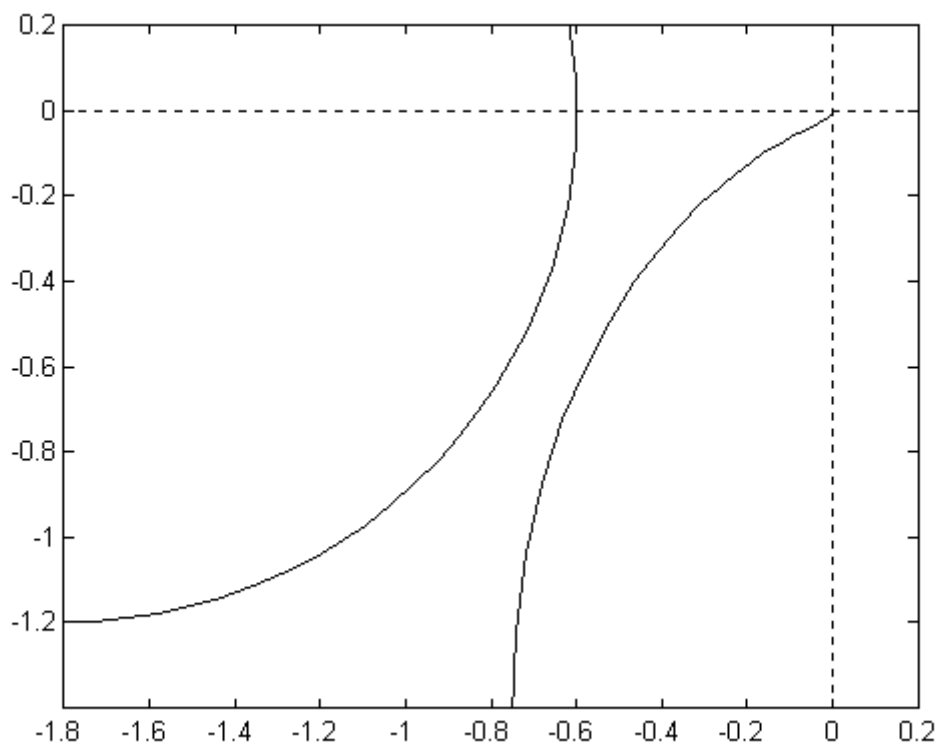


Рис. 13. Результат побудови