

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УКРАЇНСЬКИЙ ДЕРЖАВНИЙ ХІМІКО-ТЕХНОЛОГІЧНИЙ  
УНІВЕРСИТЕТ»**

**МЕТОДИЧНІ ВКАЗІВКИ  
ДО ЛАБОРАТОРНИХ ЗАНЯТЬ З ДИСЦИПЛІНИ  
“СУЧАСНІ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ”  
ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 8.05020201–АВТОМАТИЗАЦІЯ ТА  
КОМП’ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ.**

**Дніпропетровськ УДХТУ 2016**

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УКРАЇНСЬКИЙ ДЕРЖАВНИЙ ХІМІКО-ТЕХНОЛОГІЧНИЙ  
УНІВЕРСИТЕТ»

МЕТОДИЧНІ ВКАЗІВКИ  
ДО ЛАБОРАТОРНИХ ЗАНЯТЬ З ДИСЦИПЛІНИ  
“СУЧАСНІ ПРОБЛЕМИ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ”  
ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 8.05020201 – АВТОМАТИЗАЦІЯ ТА  
КОМП’ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ.

Затверджено на засіданні кафедри  
комп’ютерно-інтегрованих  
технологій і метрології.  
Протокол № 6 від 21 грудня 2015 р.

Дніпропетровськ УДХТУ 2016

Методичні вказівки до лабораторних занять з дисципліни “Сучасні проблеми автоматизованого управління” для студентів спеціальності 8.05020201 – Автоматизація та комп’ютерно-інтегровані технології. Частина 1 / Укл.: Г.І.Манко. – Дніпропетровськ: УДХТУ, 2016. – 44 с.

Укладач Г.І. Манко, канд. техн. наук

Відповідальний за випуск О.П. Мисов, канд. техн. наук

Навчальне видання

Методичні вказівки

до лабораторних занять

з дисципліни “ Сучасні проблеми автоматизованого управління”  
для студентів спеціальності 8.05020201– Автоматизація та комп’ютерно-  
інтегровані технології

Укладач: МАНКО Геннадій Іванович

Редактор Л.М. Тонкошкур

Коректор Л.Я. Гоцуцова

Підп. до друку \_\_\_\_\_ . Формат 60x84 1/16. Папір ксерокс. Друк різнограф.  
Умовн.-друк. арк. \_\_\_\_\_ Облік. – вид. арк. \_\_\_\_\_ Тираж \_\_\_\_\_ прим. Зам. № \_\_\_\_\_  
Свідотство ДК №303 від 27.12.2000.

---

УДХТУ, 49005, Дніпропетровськ-5, пр-т Гагаріна, 8

---

Видавничо-поліграфічний комплекс ІнКомЦентру

# Лабораторна робота № 1

## СТРУКТУРНІ МОДЕЛІ ДИНАМІЧНИХ ПРОЦЕСІВ

**Мета роботи:** вивчення методів побудови структурних моделей динамічних процесів.

Прикладні методи дослідження систем автоматичного управління базуються, як правило, на використанні структурних моделей динамічних процесів, що протікають в них. Структурна схема є графічним зображенням математичної моделі системи у вигляді з'єднань ланок і дає наочне уявлення про зв'язки між ланками, проходженні і перетворенні сигналів в системі. На побудові структурних схем заснований метод імітаційного моделювання динамічних процесів в системах автоматичного управління.

Перевагами імітаційного моделювання в середовищі SIMULINK є простота і наочність. Від користувача не потрібно знання яких-небудь мов програмування.

Розробка моделей засобами SIMULINK (надалі S-моделей) заснована на використанні технології Drag-and-Drop. Для побудови S-моделі використовуються модулі (чи блоки), що зберігаються у бібліотеках SIMULINK.

Блоки, що включаються в створювану модель, можуть бути пов'язані один з одним як за інформацією, так і по управлінню. Тип зв'язку залежить від типу блоку і логіки роботи моделі. Дані, якими обмінюються блоки, можуть бути скалярними величинами, векторами або матрицями довільної розмірності.

Будь-яка S-модель може мати ієрархічну структуру, тобто складатися з моделей нижчого рівня, причому число рівнів ієрархії практично не обмежене.

Разом з іншими параметрами моделювання користувач може задавати спосіб зміни модельного часу (з постійним або змінним кроком), а також умови закінчення моделювання.

В ході моделювання є можливість спостерігати за процесами, що відбуваються в системі. Для цього використовуються спеціальні "оглядові вікна", що входять до складу бібліотеки SIMULINK. Характеристики системи, що цікавлять користувача, можуть бути представлені як в числовій, так і в графічній формі. Крім того, існує можливість включення до складу моделі засобів анімації.

Ще одна важлива перевага SIMULINK полягає в тому, що він є відкритою системою: склад бібліотеки може бути поповнений користувачем за рахунок розробки власних блоків.

### Запуск SIMULINK

Спочатку необхідно запустити MATLAB будь-яким способом, передбаченим у Windows. Після запуску з'являється основне вікно системи MATLAB.

У панелі інструментів натиснути кнопку . В результаті відкривається вікно інтегрованого браузера бібліотек SIMULINK, показане на рис 1.1.

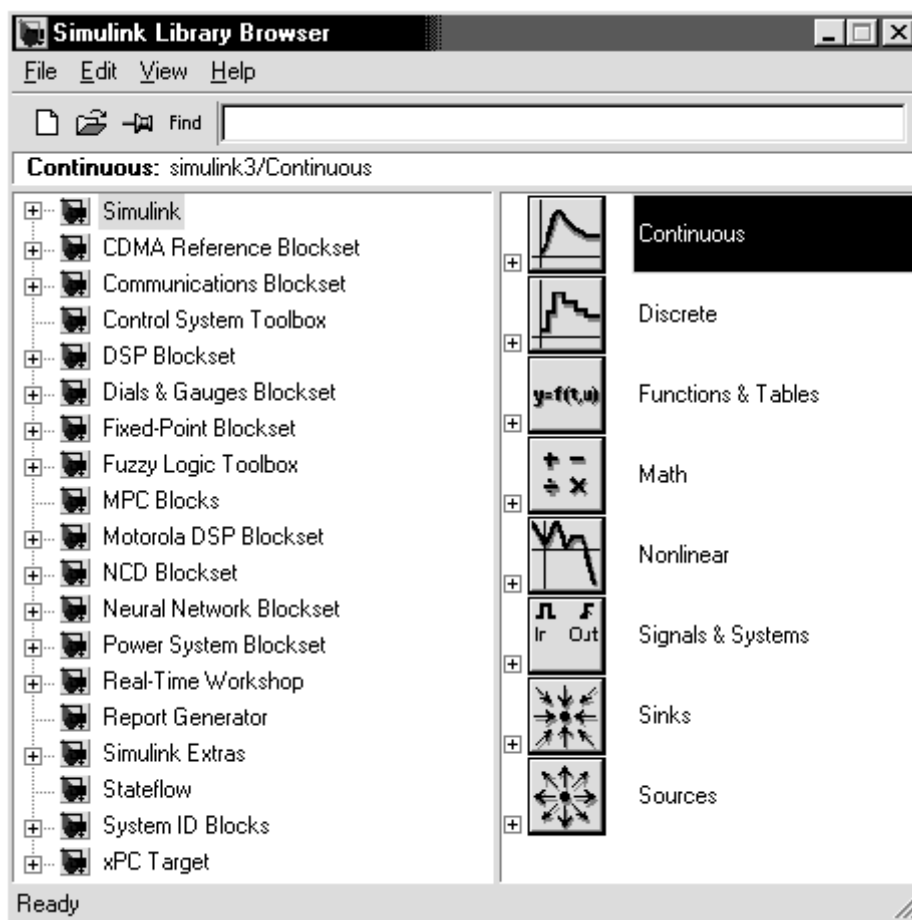




Рис. 1.1 –Вікно браузера бібліотек SIMULINK

У вікні браузера бібліотек міститься дерево компонентів бібліотек SIMULINK. Для перегляду розділу бібліотеки треба виділити його мишею - в правій частині вікна з'явиться набір піктограм компонентів активного розділу бібліотеки.

За допомогою меню браузеру або кнопок  і  його панелі інструментів можна відкрити вікно для створення нової панелі або відкрити існуючу модель.

### Створення S-моделей

Процес побудови S-моделі включає компонування моделі і завдання необхідних параметрів. Компонування полягає у виборі з бібліотек SIMULINK необхідних блоків, їх розміщення у вікні, що відкрилося, і з'єднання між собою. Далі для кожного блоку встановлюються відповідні параметри, що відповідають вимогам модельованої системи. Для їх зміни потрібно двічі клацнути по блоку і змінити потрібні значення в діалоговому вікні.

Блоки мають назви, які можна змінювати, клацнувши по ньому лівою клавішею миші і відредагувавши текст.

Блоки можна повертати. Для того, щоб повернути блок на 90 градусів, потрібно виділити його і натиснути клавіші Ctrl+R. Комбінація Ctrl+I дозволяє виконати дзеркальне відображення входів і виходів.

Блоки з'єднуються лініями зв'язку, по яких поширюються сигнали. Для того, щоб з'єднати блоки, потрібно клацнути лівою клавішею миші по джерелу

сигналу і потім, при натиснутій клавіші Ctrl, по блоку-приймачу. Можна також протягнути мишкою лінію зв'язку між потрібними виходом і входом.


При створенні відгалуження від існуючої сполучної лінії, тобто для з'єднання вхідного порту якого-небудь блоку з існуючою лінією, необхідно при виконанні з'єднання скористатися клавішею Ctrl. Також для створення точки розгалуження в сполучній лінії можна підвести курсор до передбачуваного вузла і, натиснувши праву клавішу миші, протягнути лінію.

Модель можна скопіювати у буфер обміну у вигляді растрового малюнка. Для цього у вікні моделі потрібно вибрати у верхньому меню пункт **Edit–Copy model to clipboard**. Заздалегідь краще зменшити розміри вікна до мінімальних, щоб не було білих полів.

Блок-діаграми SIMULINK можуть бути об'єднані в складені блоки, що дозволяє використати ієрархічне представлення структури моделі. Створення складеного блоку можливе двома шляхами:

а) перенести з бібліотеки блоків блок *Subsystem* (підсистема) і, розкривши його вікно, сформуванати його структуру в цьому окремому вікні;

б) збудувати блок-діаграму в основному вікні створюваної моделі, виділити її частину (призначену для об'єднання в підсистему) і, перейшовши в меню, вибрати пункт **Edit–Create Subsystem**. В результаті в основному вікні автоматично утворюється складений блок, в якому знаходитиметься та частина блок-діаграми, що була виділена.

Для того, щоб запустити моделювання, потрібно клацнути ліву клавішу миші по кнопці  на панелі інструментів вікна моделі. Ця ж кнопка дозволяє зупинити моделювання при необхідності. Запуск моделі на розрахунок можна здійснити також через пункт меню **Simulation–Start**.

Заздалегідь перед запуском розробленої моделі необхідно:

а) встановити параметри роботи моделі (інтервал часу роботи моделі, метод інтеграції, погрішність та ін.). Установка цих параметрів здійснюється через пункт меню **Simulation–Parameters**. Найважливіші параметри - цей час моделювання (*Stop time*) і метод чисельної інтеграції рівнянь (*Solver options*);

б) зберегти розроблену модель (опція **File–Save**).

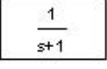

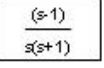


## Бібліотеки блоків SIMULINK

Доступні блоки SIMULINK відображаються у вікні *Simulink Library Browser*. Усі блоки по функціональному призначенню розподілені по бібліотеках. Серед них:






- *Continuous* – бібліотека неперервних елементів (інтегратор, диференціатор, лінійна система звичайних диференціальних рівнянь та ін.);
- *Discontinuities* – нелінійні елементи (реле, зона нечутливості, люфт та ін.);
- *Discrete* – бібліотека дискретних елементів (інтегратор з дискретним часом, дискретний фільтр та ін.);
- *Math operations* – математичні функції;
- *Logic and Bit operations* – логічні і бітові операції;

- *Sinks* – засоби відображення (тимчасова діаграма, виведення результатів у файл, зупинка виконання моделі і так далі);
- *Sources* – джерела сигналів (генератор імпульсних/синусоїдальних сигналів, генератор випадкових чисел, генератор пілкоподібних сигналів, годинник і так далі).




### Безперервні лінійні системи (Continuous)








	<b>Transfer Fcn</b> – передатна функція, в параметрах задаються чисельник ( <i>Numerator</i> ) і знаменник ( <i>Denominator</i> ) у вигляді поліномів
	<b>State Space</b> – модель в просторі станів, в параметрах задається четвірка матриць, що визначають модель, і початкові умови для вектору стану ( <i>Initial conditions</i> ).
	<b>Zero-Pole</b> – модель у формі "нулі-полюси", в параметрах задаються масиви нулів ( <i>Zeros</i> ), полюсів ( <i>Poles</i> ), а також коефіцієнт посилення ( <i>Gain</i> ).
	<b>Integrator</b> – інтегратор з можливістю установки початкових умов ( <i>Initial condition</i> ), а також границь насичення ( <i>Lower saturation limit</i> і <i>Upper saturation limit</i> ). Коли сигнал виходу виходить за границі, визначувані цими межами, інтеграція припиняється.
	<b>Transport Delay</b> – блок фіксованої затримки.

### Нелінійні елементи (Discontinuities)



	<b>Relay</b> – блок реалізує релейну характеристику згістерезисом.
	<b>Saturation</b> – блок виконує обмеження величини сигналу.
	<b>Backlash</b> – блок моделює нелінійність типу «люфт».
	<b>Rate Limiter</b> – блок забезпечує обмеження швидкості зміни сигналу (першої похідної).
	<b>Dead Zone</b> – блок реалізує нелінійну залежність типу "зона нечутливості (мертва зона)"

### Основні джерела сигналів (Sources)







	<b>Constant</b> – сигнал постійної величини.
	<b>Step</b> – ступінчастий сигнал, міняється час стрибка ( <i>Step Time</i> ), початкове ( <i>Initial Value</i> ) і кінцеве значення ( <i>Final Value</i> ).
	<b>Ramp</b> – лінійно зростаючий сигнал із заданим нахилом ( <i>Slope</i> ). Можна задати також час початку зміни сигналу ( <i>Start Time</i> ) і початкове значення ( <i>Initial Value</i> ).

	<b>Pulse Generator</b> – генератор прямокутних імпульсів, задаються амплітуда ( <i>Amplitude</i> ), період ( <i>Period</i> ), ширина ( <i>Pulse Width</i> у відсотках від періоду), фаза ( <i>Phase Delay</i> ).
	<b>Repeating Sequence</b> – послідовність імпульсів, їх форма задається у вигляді пар чисел (час; величина сигналу).
	<b>Sine Wave</b> – синусоїдальний сигнал, задається амплітуда ( <i>Amplitude</i> ), частота ( <i>Frequency</i> ), фаза ( <i>Phase</i> ) і середнє значення ( <i>Bias</i> )
	<b>Signal Builder</b> – будівник сигналів, що дозволяє задавати форму сигналу, перетягуючи мишею опорні точки
	<b>Random Number</b> – випадкові числа з нормальним (гаусовим) розподілом. Можна задати середнє значення ( <i>Mean Value</i> ), дисперсію ( <i>Variance</i> ), період зміни сигналу ( <i>Sample Time</i> )
	<b>Uniform Random Number</b> – випадкові числа з рівномірним розподілом в заданому інтервалі від <i>Minimum</i> до <i>Maximum</i>
	<b>Band Limited White Noise</b> – випадковий сигнал, обмежений по смузі білий шум (що має рівномірний спектр, до деякої частоти). Задається інтенсивність ( <i>Noise Power</i> ) і інтервал дискретизації ( <i>Sample Time</i> ), впродовж якого утримується постійне значення сигналу

### Основні облаштування виводу (*Sinks*)

	<b>Display</b> – цифровий дисплей, показує зміни вхідного сигналу в цифровому виді
	<b>Scope</b> – осцилограф, показує зміни сигналу у вигляді графіку, дозволяє передавати дані в робочу область MATLAB для подальшої обробки і оформлення



### Інші часто використовувані блоки


	<b>Gain</b> – підсилювач, задається коефіцієнт посилення ( <i>Gain</i> )
	<b>Sum</b> – суматор, використовується для складання і віднімання входів. Параметр <i>List of signs</i> задає кількість входів, їх знаки ("+" для складання і "-" для віднімання), проміжки між входами (позначаються знаком )
	<b>Trigonometric Function</b> – тригонометрична функція
	<b>Signal Routing Manual Switch</b> – ручний перемикач, дозволяє подвійним клацанням перемикати вихід на один з двох вхідних сигналів
	<b>Mux</b> – мультиплексор, об'єднує декілька сигналів в один "джгут" (векторний сигнал), в параметрах задається число входів ( <i>Number of Inputs</i> )
	<b>Demux</b> – демультимплексор, дозволяє "розбити" векторний сигнал на декілька скалярних, в параметрах задається число виходів ( <i>Number of Outputs</i> )



## Блок Score

У вікні блоку **Score** виводиться графік зміни вхідного сигналу блоку. Якщо треба вивести графіки кількох сигналів, використовується мультиплексор **Mux**.

За умовчанням на осі ординат використовується діапазон від  $-5$  до  $5$ . Якщо цей варіант не підходить, можна вибрати масштаб автоматично (так, щоб було видно увесь графік) за допомогою кнопки . Сусідня кнопка  зберігає ці налаштування для наступних запусків.

Кнопка  відкриває вікно налаштувань, причому найбільш важливі дані містяться на вкладці *Data history*. Якщо не скинути прапорець *Limit data points*, у пам'яті зберігатиметься тільки задане число точок графіка, тобто, при великому часі моделювання початок графіка буде втрачений.

Відмітивши на цій же сторінці прапорець *Save data to workspace*, можна відразу передати результати моделювання в робочу область Matlab для того, щоб їх можна було далі обробляти, виводити на графіки і зберігати у файлі. Поле *Variable name* задає ім'я змінної в робочій області, в якій зберігаються дані. У простому випадку вибирається формат *Array* (у списку *Format*). Це означає, що дані зберігатимуться в масиві з декількох стовпців (перший стовпець – час, другий – перший сигнал, третій – другий сигнал і так далі, у порядку входів мультиплексора).

### Установка параметрів моделювання

Установка параметрів моделювання виконується командою *Simulation Parameters* в меню *Simulation* вікна Simulink. При цьому відкривається вікно установки параметрів моделювання. Це вікно має ряд вкладок з досить великим числом параметрів. Розглянемо вкладку, відкриту за умовчанням, *Solver* (Вирішувач). Ця вкладка дозволяє встановити параметри системи моделювання Simulink.

До найважливіших параметрів вирішувача належить час моделювання - *Simulation time*. Він задається початковим часом *Start time* (зазвичай  $0$ ) і кінцевим часом *Stop time* (за умовчанням  $10$  с). Рівність *Stop time* нескінченності (**inf**) означає, що моделювання відбуватиметься нескінченно довго, поки не перервати його. Проте в цьому випадку важко отримати помітні осцилограми роботи пристрою, тому рекомендується задавати кінцеві значення *Stop time*.

Час моделювання – величина умовна. Відповідності між часом моделювання в секундах чи інших одиницях часу і заданим значенням немає. Реальний час моделювання залежить від швидкодії комп'ютера, на якому виконується моделювання.

Первинне значення мають дві опції вирішувача в полі *Solver options*: тип рішення і метод рішення. Можливі два типи рішення :

- *Variable-step solvers* – рішення зі змінним кроком;
- *Fixed-step solvers* – рішення з фіксованим кроком.

Як правило, кращі результати дає рішення зі змінним кроком (зазвичай за часом, але не завжди). В цьому випадку крок автоматично зменшується, якщо

швидкість зміни результатів в процесі рішення зростає. І навпроти, якщо результати міняються слабо, крок рішення автоматично збільшується. Це виключає (знову-таки, як правило) розбіжність рішення, яке нерідко трапляється при фіксованому кроці.

Метод з фіксованим кроком варто застосовувати тільки тоді, коли фіксований крок обумовлений специфікою рішення задачі, наприклад, якщо її мета полягає в отриманні таблиці результатів з фіксованим кроком. Цей метод дає непогані результати, якщо поведінка системи описується майже монотонними функціями.

Друга з вказаних опцій – вибір методу моделювання. Для вирішення диференціальних рівнянь можна вибрати наступні методи: *discrete* (дискретний), *ode45*, *ode23* (три варіанти, включаючи метод Розенброка), *rk45* (метод Дорманда-Принса), *ode113* (метод Адамса) і *ode15s*. Методи, в найменуванні яких є слово *stiff*, служать для вирішення жорстких систем диференціальних рівнянь.

Наступні три параметри зазвичай задаються автоматично (значення опції *Auto*) :

- *Max step size* - максимальний крок інтеграції системи однорідних диференціальних рівнянь;

- *Min step size* - мінімальний крок інтеграції;

- *Initial step size* - початковий крок інтеграції.

Важливим параметром є і точність інтеграції :

- *Relative tolerance* - відносна погрішність інтеграції;



- *Absolute tolerance* - абсолютна погрішність інтеграції.


За умовчанням вони мають значення  $10^{-3}$  і  $10^{-6}$  відповідно. Зменшення погрішності призводить до збільшення часу обчислення; збільшення погрішності може привести до фрагментації графіків результатів моделювання.

### Порядок виконання лабораторної роботи

1. На диску D комп'ютера створити свою власну теку з іменем, написаним латинськими символами.

2. Запустити браузер SIMULINK. Натиснути кнопку "Створити" панелі інструментів. Відкрити бібліотеку джерел вхідних дій *Sources*. Перетягнути

мишею у вікно нової моделі джерела синусоїдального  і одиничного ступінчастого  сигналів.

З розділу бібліотеки *Sinks* витягнути осцилоскоп .

Для одночасного спостереження декількох сигналів на одному осцилоскопі потрібний мультиплексор ,

що знаходиться в розділі бібліотеки *Signals & Systems*.

3. Зібрати схему, як показано на рис 1.2 а), сполучаючи мишею входи і виходи елементів, при необхідності змінюючи їх розміри за допомогою "ручок", що з'являються на кутах елемента при його виділенні.

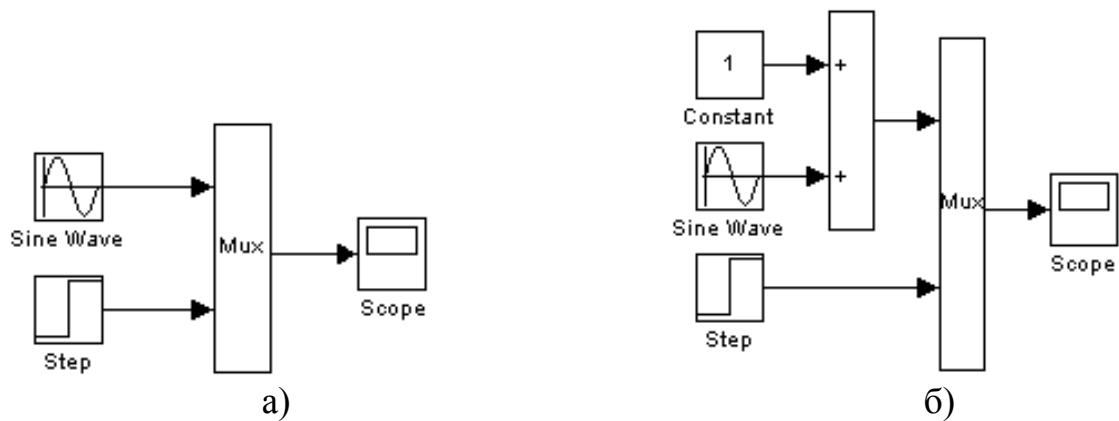



Рис. 1.2 – Схема дослідження генераторів сигналів

4. Для налаштування параметрів елементу схеми двічі клацнути його мишею і у вікні, що відкрилося, зробити необхідні зміни.

5. Запустити моделювання, натиснувши кнопку  панелі інструментів. Для перегляду результатів відкрити вікно осцилоскопа подвійним клацанням.

6. Ввести в синусоїдальний сигнал постійну складову, використовуючи джерело постійного сигналу **Constant** з бібліотеки *Sources* і суматор **Sum** з бібліотеки *Math*, як показано на рис. 1.2 б). Запустити моделювання і подивитися результат у вікні осцилоскопа.

7. Для дослідження інтеграції і диференціювання створити S-модель, показану на рис. 1.3, використовуючи елементи **Discrete Pulse Generator** з бібліотеки *Sources*, **Derivative** і **Integrator** з бібліотеки *Continuous*. Зберегти модель у своїй теці під ім'ям lab1.mdl.

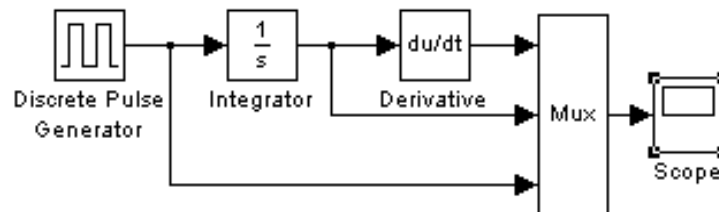


Рис. 1.3 –S-модель для дослідження роботи елементів SIMULINK

8. Запустити моделювання і пояснити отриманий результат.

9. Змінити час моделювання і точність інтеграції. Виконати пункт 8 з новими параметрами.

10. Звіт з лабораторної роботи повинен містити мету роботи, схеми Simulink, осцилограми, що спостерігаються на кожному етапі роботи, а також висновки про проведені дослідження.

### Контрольні питання

1. Що називається структурною схемою і які достоїнства структурних моделей?
2. Що таке Simulink Library Browser?
3. Як зробити, щоб один і той же сигнал надходив на декілька блоків?
4. Як продублювати блок у вікні моделі?
5. Як скопіювати зображення моделі в документ Microsoft Word?
6. Як змінити знаки арифметичних дій в суматорі?
7. Як одночасно спостерігати на осцилографі кілька сигналів?
8. Як додати до сигналу змінної напруги постійну складову?

## Лабораторна робота № 2

### ДОСЛІДЖЕННЯ ПЕРЕХІДНИХ ХАРАКТЕРИСТИК КОЛИВАЛЬНОЇ ЛАНКИ

**Мета роботи:** отримання і аналіз перехідних характеристик коливальної ланки при різних значеннях його параметрів.

Перехідною характеристикою (перехідним процесом)  $h(t)$  називають реакцію якої-небудь системи на одиничну ступінчасту дію (рис. 2.1).

У цій роботі об'єктом досліджень є коливальна ланка. Розглянемо побудову моделі такої ланки на прикладі активного коливального RLC-контуру (рис. 2.2).

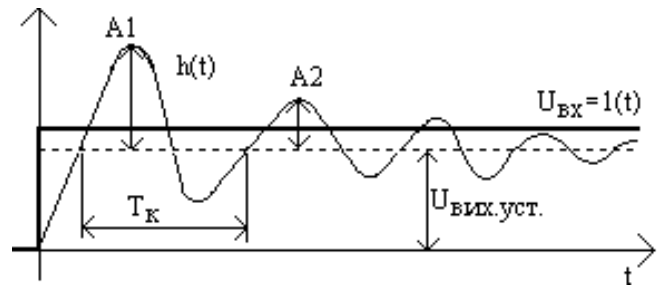


Рис. 2.1 – Перехідна характеристика коливальної ланки

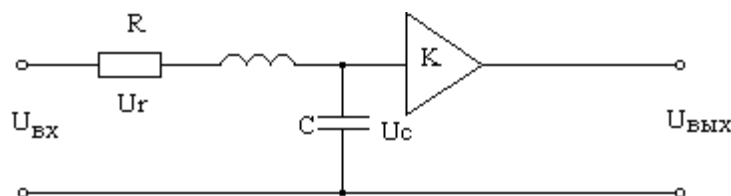


Рис. 2.2 – Активний коливальний контур

Рівняння, що описує електричні процеси в цьому контурі:

$$LC \frac{d^2 U_{ВЫХ}}{dt^2} + RC \frac{dU_{ВЫХ}}{dt} + U_{ВЫХ} = KU_{ВХ} \quad (2.1)$$

При необхідності дослідження процесів в елементах цього контура, а також з метою розкладання перетворення сигналів в контурі на прості обчислювальні операції (підсумовування, множення, інтеграція) будуюмо структурну схему (рис. 2.3).

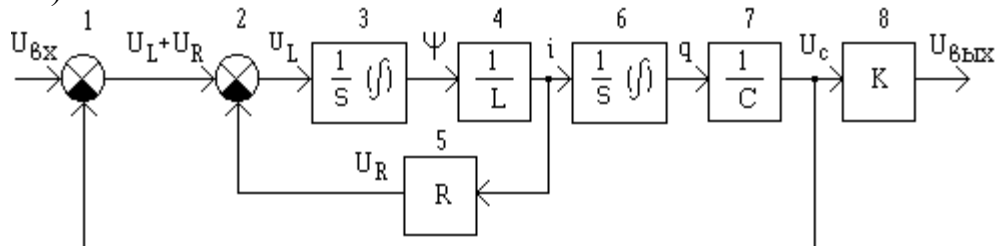


Рис. 2.3 – Структурна модель коливальної ланки

У цій моделі різниця напруги  $U_{ВХ}$  і на конденсаторі  $U_c$  визначає сумарну напругу на індуктивності і опорі  $U_L + U_R$  (суматор 1). Якщо з сумарної напруги відняти величину напруги на опорі  $U_R$ , то залишиться напруга на індуктивності  $U_L$  (суматор 2). Шляхом інтеграції за часом напруги на індуктивності отримуємо потікозчеплення  $\psi$  (інтегратор 3), а діленням останнього на величину індуктивності – струм  $i$  (підсилювач 4). Зазвичай використовується форма двох останніх перетворень:

$$U_L = L \frac{di}{dt}$$

Струм  $i$  створює падіння напруги на опорі  $U_R=iR$  (підсилювач 5). Інтегрування струму за часом дозволяє розрахувати заряд конденсатора  $q$  (інтегратор 6), а ділення заряду на величину ємності – напругу на конденсаторі  $U_C$  (підсилювач 7). Останні два перетворення записуються в такій формі:

$$i = C \frac{dU_C}{dt}$$

Шляхом множення напруги на конденсаторі на коефіцієнт посилення  $K$  знайдемо вихідну напругу  $U_{\text{вих}}$  (підсилювач 8).

Аналогічну структурну модель коливального контура можна вичислити, не розглядаючи математичний опис окремих елементів, а використовуючи повне рівняння контура (2.1). Для цього в рівнянні треба виділити старшу похідну вихідної величини:

$$\frac{d^2 U_{\text{вих}}}{dt^2} = -\frac{R}{L} \frac{dU_{\text{вих}}}{dt} + \frac{1}{LC} (KU_{\text{вх}} - U_{\text{вих}}) \quad (2.2)$$

Відповідно до отриманого рівняння побудуємо структурну схему, почи-

наючи з величини  $\frac{d^2 U_{\text{вих}}}{dt^2}$  (рис. 2.4). Двічі проінтегрувавши цю величину, роз-

рахуємо  $\frac{dU_{\text{вих}}}{dt}$  і  $U_{\text{вих}}$  (інтегратори 5 і 6). Шляхом множення  $U_{\text{вх}}$  на  $K$  шляхом

множення  $U_{\text{вих}}$  (суматор 2), а потім за допомогою множення результуючого сигналу на  $\frac{1}{LC}$  (підсилювач 3) знайдемо  $\frac{1}{LC} (KU_{\text{вх}} - U_{\text{вих}})$ .

Віднявши від цієї величини сформований за допомогою підсилювача 7 сигнал  $\frac{R}{L} \frac{dU_{\text{вих}}}{dt}$  (суматор 4),

згідно з рівнянням (1.2) отримаємо величину  $\frac{d^2 U_{\text{вих}}}{dt^2}$ .

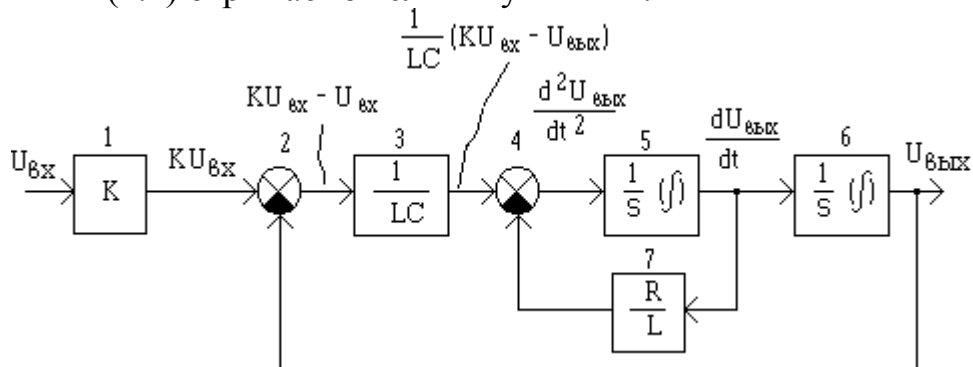


Рис. 2.4 – Перетворена структурна модель коливальної ланки

Використовуючи правила перетворення структурних схем, можна довести еквівалентність моделей, показаних на рис. 2.3 і 2.4. У них застосовуються лише різні внутрішні змінні. Важливо відмітити, що усі подібні структурні моделі будуються на базі інтеграторів, масштабних підсилювачів і суматорів. Ця модель будується на двох послідовно сполучених блоках-інтеграторах, охоплених загальним негативним зворотним зв'язком з коефіцієнтом, рівним 1.


Параметри коливальної ланки обчислюють по наступних формулах:

$$K = \frac{U_{\text{вих.уст}}}{U_{\text{вх}}} \text{—коєфіцієнт посилення; } \omega = \frac{2\pi}{T_K} \text{—частота власних коливань;}$$

$$\alpha = \frac{1}{T_K} \ln\left(\frac{A_1}{A_2}\right) \text{—коєфіцієнт загасання; } \omega_0 = \sqrt{\omega^2 - \alpha^2} \text{—резонансна частота;}$$

$$T = \frac{1}{\omega_0} \text{—стала часу коливальної ланки; } \xi = \frac{\alpha}{\omega_0} \text{—міра демпфування.}$$

### Порядок виконання лабораторної роботи

1. Для запуску пакету SIMULINK клацніть по кнопці  в командному вікні MATLAB або введіть команду *simulink* в командному рядку.

2. Створіть нову модель за допомогою верхнього меню вікна Simulink Library Browser, що відкрилося. Для цього в меню **File** виберіть пункт **New**, а потім **Model**, або скористайтеся поєднанням клавіш Ctrl+N.

3. Перетягніть блок **Gain** (підсилювач) з вікна Simulink Library Browser (група Math Operations) у вікно моделі і продублюйте його двічі. Відредагуйте опис блоків (1/L, 1/C, R). Перейдіть у вікно редагування властивостей (двічі натиснувши на ліву клавішу миші на зображенні блоку) кожного блоку і занесіть у поле *Gain* відповідне значення коєфіцієнту  $K_1=1/L$ ;  $K_2=1/C$ ;  $K_3=R$  згідно варіантів, наведених у табл. 2.1.

Таблиця 2.1

Варіант	0	1	2	3	4	5	6	7	8	9
<i>L</i> , Гн	10	1	100	1	0,1	10	100	10	100	0,1
<i>C</i> , мкФ	10	100	1000	100	200	1000	10	100	1000	1
<i>R</i> , Ом	50	5	15	50	5	10	150	15	1,5	10

4. Перетягніть блок **Integrator** (інтегратор) з вікна Simulink Library Browser (група *Continuous*) у вікно моделі і продублюйте його.

5. Перетягніть блок **Sum** (суматор) з Simulink Library Browser (група *Math Operations*) у вікно моделі і перейдіть у вікно редагування його властивостей (двічі натиснувши на ліву клавішу миші на зображенні блоку). Для властивості Icon shape встановіть *rectangular* (прямокутний вид суматора), а для властивості List of signs задайте *+-*, що говорять про те, що буде використано три входи, два, що віднімають, та один, що підсумовує.

6. Перетягніть у вікно моделі блок **Scope** (осцилограф) з групи *Sinks* і встановіть його в правій частині вікна.

7. Перетягніть у вікно моделі блок **Step** з групи *Sources* і встановіть його ліворуч від суматора.

8. Виконайте з'єднання блоків відповідно до схеми, представленої на рис. 2.4.

9. Збережіть модель у своїй теці під ім'ям lab2.mdl.

10. Час моделювання підбирайте так, щоб осцилограма була схожа на рис. 2.1.

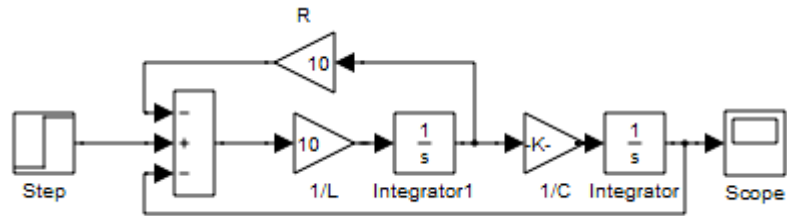


Рис. 2.4 – S-модель коливальної ланки

11. Отримайте перехідні характеристики для трьох значень коефіцієнта  $K3$ . Результати вимірів занести в табл. 2.1.

Таблиця 2.1

Перехідна характеристика	$U_{вх}$	$U_{вих.уст}$	$A1$	$A2$	$T_K$	$K$	$\omega$	$\alpha$	$\omega_0$	$T$	$\xi$
$R_{min}$											
$R$											
$R_{max}$											

Тут  $U_{вх}$  – величина сходинки вхідного сигналу,  $U_{вих.уст}$  значення вихідного сигналу, що встановилося,  $A1$  і  $A2$  – амплітуди першого і другого коливань,  $T_K$  – період коливань, вимірювані безпосередньо за перехідною характеристикою (див. рис. 2.1.).  $R_{min}$ ,  $R$ ,  $R_{max}$ , – значення величини активного опору коливального контура  $R$  (в порядку зростання значень).

Значення  $R_{min}$  рекомендується узяти в  $2 \div 3$  рази менше, ніж  $R$ , а  $R_{max}$  - в  $2 \div 3$  рази більше. В ході досліджень  $U_{вх}$ ,  $1/L(K1)$  і  $1/C(K2)$  повинні залишатися незмінними.

12. Замалювати перехідні характеристики для трьох значень опору.

13. Дослідіть зміни перехідної характеристики при змінах  $L$  і  $C$  і зробіть висновки.

14. Звіт повинен містити мету роботи, рисунок початкової структурної моделі, схеми SIMULINK, таблицю з результатами, розрахункові формули і графіки перехідних характеристик, а також висновки про проведені дослідження.

### Контрольні питання

1. Що називається перехідною характеристикою?
2. Що називається імпульсною перехідною характеристикою?
3. Що таке "одинична ступінчаста дія"?
4. Поясніть зміну перехідної характеристики при зміні  $R$ .
5. Які відмінності між  $\omega$  і  $\omega_0$ ?

## Лабораторна робота № 3 ДОСЛІДЖЕННЯ СТІЙКОСТІ САУ

**Мета роботи :** експериментальне підтвердження теоретичного обґрунтування стійкості динамічних систем.

У загальному випадку стійкістю системи називається її здатність повертатися в початковий стан після припинення зовнішньої збурюючої дії, яка вивела цю систему з початкового стану.

Відомо, що перехід системи з одного стійкого стану в інший під впливом зовнішньої дії описується рівняннями динаміки. Наочне представлення процесу зміни станів системи можна отримати, спостерігаючи перехідну характеристику на екрані осцилографа (див. лабораторну роботу № 2). Для стійких систем перехід з одного стану в інший може носити монотонний (1), аперіодичний (2) або загасаючий коливальний характер (3) (рис. 3.1). У нестійких системах перехідний процес зазвичай набуває характеру незгасаючих коливань.

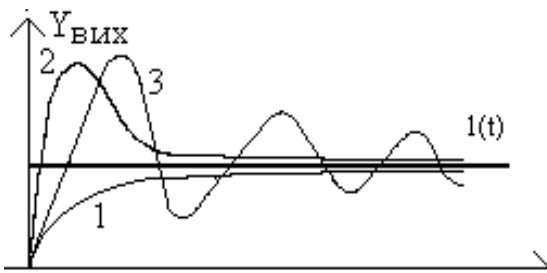


Рис. 3.1 –Різні види перехідних процесів

Стійкість систем зазвичай визначають за критеріями стійкості, зокрема за критерієм Гурвиця. Це алгебраїчний критерій, при використанні якого про стійкість систем судять по коефіцієнтах характеристичних рівнянь.

Нехай  $W(s)=Y(s)/U(s)$  – передавальна функція системи, а  $U(s)=0$  — характеристичне рівняння системи. Запишемо характеристичний поліном  $U(s)$  у вигляді

$$U(s) = a_0 s^n + a_1 s^{n-1} + \dots + a_n. \quad (3.1)$$

Із коефіцієнтів характеристичного рівняння (3.1) будується матриця розмірності  $n \times n$  згідно наступному алгоритму:

- по головній діагоналі зліва направо виставляються всі коефіцієнти характеристичного рівняння від  $a_1$  до  $a_n$ ;
- від кожного елемента діагоналі вгору і вниз добудовуються стовпці матриці так, щоб індекси зменшувалися згори донизу;
- на місце коефіцієнтів з індексами менше нуля або більше  $n$  ставляться нулі.

Відповідно до **критерію Гурвиця:**

– для того, щоб динамічна система була стійка, необхідно і достатньо, щоб усі  $n$  діагональних мінорів матриці були додатні. Ці мінори називаються визначниками Гурвиця.

Для рівнянь 1-го і 2-го степеня умова стійкості визначається позитивністю усіх коефіцієнтів. Для рівнянь 3-го степеня, окрім позитивності усіх коефіцієнтів, необхідно, щоб визначник Гурвиця 2-го порядку був більше нуля. Тобто умови стійкості системи третього порядку мають вигляд:



$$\Delta_1 = a_{n-1} > 0; \quad \Delta_2 = \begin{vmatrix} a_{n-1} & a_{n-3} \\ a_n & a_{n-2} \end{vmatrix} > 0$$

Прикладом такої системи можуть служити послідовно сполучені інерційна і коливальна ланки, охоплені негативним зворотним зв'язком (рис. 3.2).

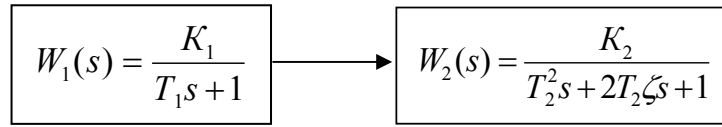


Рис. 3.2 – Структура системи третього порядку

Тут  $K_1, K_2$  – коефіцієнти передачі;  $T_1, T_2$  – сталі часу;  $\zeta$  – коефіцієнт демпфування.

Для отримання перехідної характеристики необхідно зібрати схему, показану на рис. 3.3.

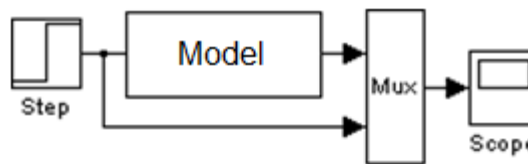


Рис. 3.3 – Структурна схема отримання перехідної характеристики

Схема моделі наведена на рис. 3.4.

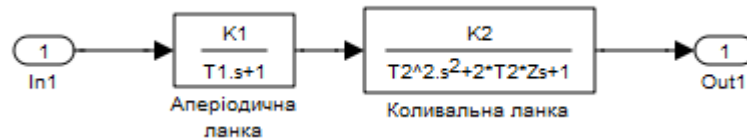


Рис. 3.4 – S-модель системи третього порядку

Параметри моделі беруться з табл. 3.1 і вводяться у командному вікні MATLAB за наступним зразком:

```
>> K1=1; T1=0.1; K2=2; T2=0.4; Z=0.5;
```

Тут змінна  $Z$  являє собою коефіцієнт демпфування  $\zeta$ .

Таблиця 3.1

Варіант	$K_1$	$T_1$	$K_2$	$T_2$	$Z$	Варіант	$K_1$	$T_1$	$K_2$	$T_2$	$Z$
1	0.1	0,5	10	0,2	0,1	13	0,333	0,2	3	0,5	1,2
2	0.2	0,4	5	0,1	0,2	14	0,25	0,1	4	0,4	0,5
3	0.333	0,3	3	0,3	-0,1	15	0,2	0,3	5	0,3	-0,6
4	0.25	0,2	4	0,1	0,3	16	0,167	0,1	6	0,2	0
5	0.2	0,1	5	0,2	-0,4	17	0,143	0,2	7	0,1	1,3
6	0,167	0,2	6	0,4	0,8	18	0,125	0,4	8	0,2	0,9
7	0,143	0,3	7	0,5	1,3	19	0,5	0,5	2	0,3	-0,2
8	0,125	0,4	8	0,1	-1	20	0,2	0,1	5	0,4	0,5
9	0,5	0,3	2	0,2	0,9	21	0,1	0,2	10	0,3	0,4
10	0,2	0,6	5	0,3	0,5	22	0,2	0,3	5	0,6	0,1
11	0,1	0,2	10	0,4	0,6	23	0,25	0,4	4	0,2	0,3
12	0,2	0,4	5	0,5	0,7	24	0,333	0,5	3	0,4	0,2

### Порядок виконання лабораторної роботи

1. Створити S-модель системи третього порядку, як показано на рис. 3.4. Перетягнути блок **Transfer Fcn** (передавальна функція) з вікна Simulink Library Browser (група *Continuous*) у вікно моделі. Блок **Transfer Fcn** дублювати. Блоки **In1** і **Out1** беруться з бібліотек *Sources* і *Sinks* відповідно.

2. Введіть чисельники (*Numerator*) і знаменники (*Denominator*) передатних функцій за наступною схемою:

Аперіодична ланка	Коливальна ланка
<i>Numerator</i> [K1]	<i>Numerator</i> [K2]
<i>Denominator</i> [T1 1]	<i>Denominator</i> [ T2*T2 2*T2*Z 1]

3. Створити з схеми за п. 1 підсистему (**Subsystem**). Для цього виділити усі елементи схеми, натиснути на виділеній області праву клавішу миші і у вікні, що з'явилося, виберіть пункт *Create Subsystem*.

3. Відредагувати назву блоку, що утворився, назвавши його **Model**.

4. Перетягнути у вікно моделі блок **Step** з групи *Sources* і встановити його ліворуч від блока моделі.

5. Перетягніть у вікно моделі блок **Mux** з групи *Signal Routing* і встановити його праворуч від блока моделі.

6. Перетягнути у вікно моделі блок **Scope** (осцилограф) з групи *Sinks* і встановити його в правій частині вікна.

7. Виконайте з'єднання блоків відповідно до схеми, представленої на рис. 3.3.

8. Зберегти модель у своїй теці під ім'ям lab3.mdl

9. Зняти перехідні характеристики при трьох різних значеннях параметра моделі Z. Оцінити перерегулювання  $\sigma$  — це максимальне відхилення перехідної характеристики від значення, що встановилося:

$$\sigma = \frac{\Delta X_{вих}}{X_{вих}} 100. \quad (3.2)$$

У випадку коливального характеру перехідного процесу визначити період коливань  $T_k$  відповідності з рис. 3.1. Занести результати в табл. 3.1

Таблиця 3.1

№ п/п	Z	$\sigma$	$T_k$	Характер перехідного процесу	$M_1$	$M_2$	Стійкість системи
1							
2							
3							

10. Вивести вираження передатної функції системи. Провести аналіз отриманих результатів і визначити умови стійкості системи за критерієм Гурвіця, використовуючи засоби MATLAB. Для цього набрати в текстовому редакторі наступний скрипт.

K1=1; T1=0.1; K2=2; T2=0.4; Z=0.5;

W1=tf([K1],[T1 1]) % аперіодична ланка

W2=tf([K2],[T2\*T2 2\*T2\*Z 1]) % коливальна ланка

W=series(W1,W2) % з'єднання ланок

[num,den]=tfdata(W); % den - знаменник ПФ

lm=den{1, 1} % характеристичне рівняння системи

```

[L, N] =size(lm); % визначення розмірності
g=zeros(N, N);% створення матриці з нульовими значеннями
% заповнення непарних рядків матриці Гурвиця
s=0;
for i=1:2:N
j=1; j=j+s; r=0;
for r=2:2:N
g(i, j)=lm(r); j=j+1;
end
s=s+1;
end
% заповнення парних рядків матриці Гурвиця
s=0;
for i=2:2:N
j=1; j=j+s; r=0;
for r=1:2:N
g(i, j)=lm(r); j=j+1;
end
s=s+1;
end
g=g(1:N-1, 1:N-1)

minor=1; % ознака додатності мінорів
for i=1:N-1
dd(i) = det(g(1:i, 1:i)); % обчислення мінорів
if dd(i)<0
minor=0; % ознака від'ємності мінорів
end
end

fprintf(1, ' Minor = %g ', dd)
% виведення результатів
if minor==0
fprintf(1, '\n Nonstable system\n');
else
fprintf(1, '\nStable system\n');
end

```

11. Занести в табл. 3.2 значення мінорів першого ( $M_1$ ) і другого ( $M_2$ ) рядків.

12. Занести в табл. 3.2 ознаку стійкості системи. Для стійкої системи в командному вікні виводиться текст «Stable system», для нестійкої – «Nonstable system».

13. Порівняти експериментальні дані з розрахунковими.

14. Звіт повинен містити мету роботи, схеми SIMULINK, скопійований з командного вікна MATLAB хід виконання скрипту за п. 10, таблицю з результа-

тами, графіки перехідних характеристик, а також висновки про проведені дослідження.

### **Контрольні питання**

1. Дайте визначення стійкості системи.
2. Сформулювати критерій стійкості Гурвиця.
3. Як визначити за коренями характеристичного рівняння, стійка система чи ні?
4. Чи може бути стійкою система регулювання, що містить нестійкий об'єкт керування?
5. Чи дозволяє вид перехідної характеристики судити про стійкість системи?
6. Які бувають характери перехідного процесу?
7. Як розрахувати параметри перехідного процесу?

## Лабораторна робота № 4 ДОСЛІДЖЕННЯ ПІД-РЕГУЛЯТОРІВ

**Мета роботи:** оцінити вплив налаштувальних параметрів ПІД-регулятора на якісні показники процесу регулювання.

Передавальна функція (ПФ) промислових об'єктів у багатьох випадках з достатньою точністю може бути представлена моделлю другого порядку з запізненням:

$$W_{oy}(s) = \frac{K_{об} e^{-\tau s}}{(T_1 s + 1)(T_2 s + 1)}, \quad (4.1)$$

де  $T_1, T_2$  – постійні часу об'єкта управління;  $K_{об}$  - коефіцієнт посилення об'єкта управління;  $\tau$  - час чистого запізнювання.

Модель системи з регулятором в загальному вигляді показана на рис. 4.1.

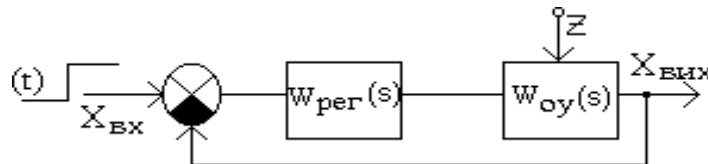


Рис. 4.1 – Структура моделі системи з регулятором

Передавальна функція ПІД-регулятора

$$W_p(s) = K_p + \frac{K_i}{s} + K_d s,$$

де  $K_p, K_i, K_d$  - відповідно, налаштувальні коефіцієнти пропорційної, інтегральної і диференціальної складових.

Основною метою використання регуляторів є зменшення статичної погрішності. Крім того, підбираючи параметри регуляторів, можна поліпшити показники перехідного процесу – час регулювання і перерегулювання.

Основні параметри, що характеризують перехідний процес, показані на рис. 4.2.

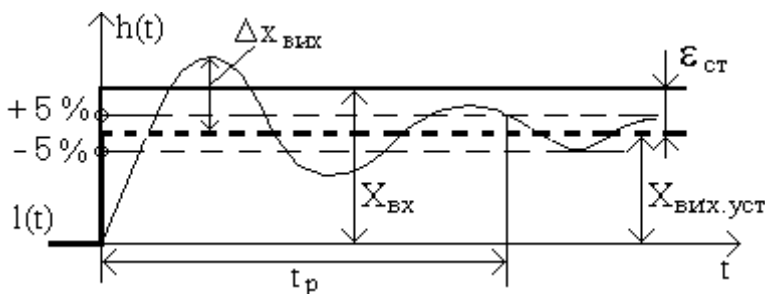


Рис. 4.2 – Параметри перехідного процесу

Статична погрішність  $\epsilon_{ст}$  є різницею між сигналом, що подається на вхід системи, і сигналом, що встановилося на її виході після закінчення перехідного процесу.

Період коливань  $T_k$  – час між однаковими переходами кривої перехідного процесу.

процесу через лінію, що є паралельною осі абсцис і відповідає встановленому значенню регульованої величини. Визначається тільки у випадку коливального перехідного процесу.

Час регулювання  $t_p$  – це час, впродовж якого крива перехідного процесу входить в межі, що визначають точність регулювання. Ці межі встановлюються розробником системи. У цій роботі точність регулювання, тобто максимально допустиме відхилення кривої перехідного процесу, слід задавати не менше  $\pm 5\%$  від значення, що встановилося. Таким чином, можна вважати, що перехідний процес закінчився після того, як крива перестала виходити із заданого діапазону.

Перерегулювання  $\sigma$  – це максимальне відхилення перехідної характеристики від значення, що встановилося:  $\sigma = \frac{\Delta X_{вих}}{X_{вих.уст.}} 100$ .

### Порядок виконання лабораторної роботи

1. Створити S-модель ПІД-регулятора, як показано на рис. 4.3. блоки **In1** і **Out1** беруться з бібліотек *Sources* і *Sinks* відповідно.

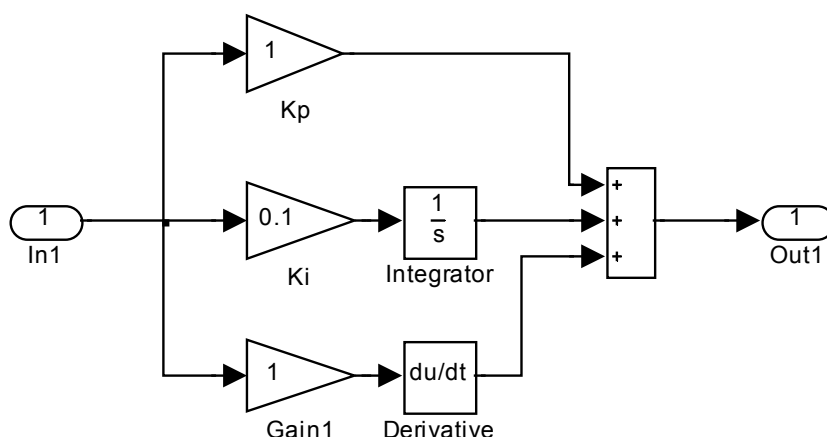


Рис. 4.3 – S-модель ПІД-регулятора

2. Створити з схеми, що реалізує ПІД-регулятор, підсистему (**Subsystem**). Для цього виділити усі елементи схеми, натиснути на виділеній області праву кнопку миші і у вікні, що з'явилося, виберіть пункт *Create Subsystem*.

3. Відредагувати назву блоку, що утворився, назвавши його **PID**.

4. Перетягнути блоки **Transfer Fcn** (передавальна функція) і **Transport Delay** (Запізнювання) з вікна Simulink Library Browser (група *Continuous*) у вікно моделі. Блок **Transfer Fcn** дублювати.

5. Введіть чисельник і знаменник передавальної функції, а також час запізнювання *Time Delay* для блоку **Transport Delay** за наступною схемою:

<b>Transfer Fcn</b>	<b>Transfer Fcn1</b>	<b>Transport Delay</b>
<i>Numerator</i> [K]	<i>Numerator</i> [1]	<i>Time Delay</i> tau
<i>Denominator</i> [T1 1]	<i>Denominator</i> [T2 1]	

Значення K, T1, T2 і tau взяти відповідно до номера варіанту з таблиці 4.1.

6. Перетягнути два блоки **Sum** (суматор) з Simulink Library Browser у вікно моделі. Для організації негативного зворотного зв'язку відредагуйте властивість *List of signs* одного з блоків (задайте + –).

7. Перетягнути у вікно моделі блок **Scope** (осцилограф) з групи *Sinks* і встановити його в правій частині вікна.

8. Перетягнути у вікно моделі 2 блоки **Step** з групи **Sources** і встановити його ліворуч від суматора. Дайте назву першому блоку **Task**(завдання), другому, - **Disturb**(збурення).

Таблиця 4.1 – Варіанти завдань

№	$K$	$T1$	$T2$	tau	№	$K$	$T1$	$T2$	tau
1	1,5	10	5	2	11	1,5	13	2	3
2	2,0	15	4	3	12	2,0	12	3	4
3	3,0	17	3	4	13	3,0	17	4	5
4	3,5	18	4	2	14	3,5	14	4	3
5	3,2	12	5	3	15	3,2	11	5	2
6	0,5	13	7	4	16	0,5	16	3	3
7	1,2	14	6	3	17	1,2	15	4	2
8	2,5	11	3	2	18	2,5	14	5	4
9	1,7	18	3	3	19	1,7	16	4	5
10	2,1	12	4	4	20	2,1	17	3	3

9. Встановити в обох блоках час подачі сигналу рівний 0.

10. Виконайте з'єднання блоків відповідно до схеми, представленої на рис. 4.4.

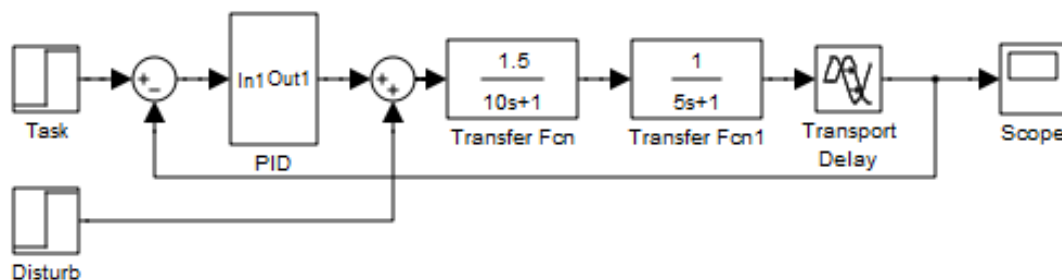


Рис. 4.4 – S-модель системи

11. Зберегти модель у своїй теці під ім'ям lab4.mdl

12. Встановити час моделювання 500 секунд (далі цей час корегувати так, щоб осцилограма нагадувала рис. 4.2).

13. У блоці з ім'ям **Task** (тип блоку **Step**) встановити величину сигналу (*Finalvalue*), що подається, рівною 1, а в блоці **Disturb** - 0. При цьому при виконанні моделювання імітуватиметься ситуація відпрацювання зміни завдання на 1. При відпрацюванні збурення необхідно навпаки встановити 0 для блоку **Task** і 1 - для блоку **Disturb**.

14. Виконайте моделювання. Вид перехідного процесу можна спостерігати у вікні **Scope**.

15. Підберіть налаштувальні параметри регулювальника, що забезпечують стійкість замкненої системи регулювання. Налаштування регулювальників ( $K_p$ ,  $K_i$ ,  $K_d$ ) встановлюються при редагуванні відповідних блоків **Gain** в підсистемі **PID**.

16. Вчислити параметри перехідного процесу і занести результати в табл. 4.2.

Таблиця 4.2

Параметри регулювання	$X_{вх},$ В	$X_{уст},$ В	$\Delta X,$ В	$\varepsilon_{ст},$ В	$T_k,$ мс	$t_p,$ мс	$\sigma, \%$	Характер перехідного процесу
Значення параметрів								

17. Звіт повинен містити мету роботи, рисунок початкової структури моделі, схеми SIMULINK, таблицю з результатами, розрахункові формули і графіки перехідних характеристик, а також висновки про проведені дослідження.

### Контрольні питання

1. Яка мета використання регуляторів в САУ?
2. Які основні параметри перехідного процесу?
3. Як зміниться час регулювання  $t_p$ , якщо збільшити точність регулювання до  $\pm 1\%$ ?
4. Що називається статичною погрішністю?
5. Як створити підсистему (Subsystem)?



## Лабораторна робота № 5

### НАЛАШТУВАННЯ ПІД-РЕГУЛЯТОРІВ

**Мета роботи:** ознайомлення з інтерактивними методами розрахунку налаштувальних параметрів ПІД-регулятора.

Загальні рекомендації по налаштуванню ПІД-регулятора наступні. Пропорційний регулятор  $K_P$  ефективно зменшує час наростання вихідного сигналу і зменшує, але не зводить до нуля, сталу помилку. Інтегруючий регулятор  $K_I$  ефективно усуває сталу помилку, але сильно погіршує перехідну характеристику. Диференціальний регулювальник  $K_D$  збільшує стійкість системи, зменшує перерегулювання і покращує перехідну характеристику. Вплив кожного регулювальника в замкнутій системі показаний у табл. 5.1.

Таблиця 5.4

Тип ланки	Час наростання	Перерегулювання	Час перехідного процесу	Статична помилка
$K_P$	Зменшує	Збільшує	Слабо впливає	Зменшує
$K_I$	Зменшує	Збільшує	Збільшує	Исключает
$K_D$	Слабо впливає	Зменшує	Зменшує	Слабо впливає

Відмітимо, що ці залежності можуть бути не дуже точними, тому що  $K_P$ ,  $K_I$  і  $K_D$  залежать один від одного. Фактично, зміна однієї з цих змінних може змінити ефект інших двох. Тому таблиця 5.4 може бути використана тільки як рекомендація при виборі величин  $K_P$ ,  $K_I$  і  $K_D$ .

Ручне налаштування ПІД-регулятора є досить трудомісткою процедурою. Ручні методи віднімають багато часу, а якщо вони застосовуються до реального устаткування, то можливе його ушкодження. Формалізовані ітеративні процедури не завжди сумісні з нестійкими об'єктами, об'єктами високого порядку, а також з об'єктами з малою постійною часу. Тому більшість фірм, що випускають промислові ПІД-регулятори, забезпечують їх засобами автоматичного налаштування. Для цього розроблено велике число методів, частина з яких є ноу-хау. У сучасних системах автоматизованого проектування і моделювання також реалізовані можливості по автоматичному налаштуванню ПІД-регуляторів. Ці інструменти можуть бути "заточені" під ПІД-регулятори, або бути часткою випадком загальніших оптимізаційних процедур. Зокрема, пакет Матлаб містить декілька інструментів, що дозволяють з тією або іншою мірою точності проводити налаштування ПІД-регуляторів по математичних моделях об'єктів управління.

Ймовірно, найпростішим є метод, що базується на використанні блоків PID, - Controller в Simulink і на алгоритмі налаштування ПІД-регуляторів в інтерфейсі Simulink Control Design.

Для використання цього методу необхідно скласти в Simulink замкнуту систему управління, включивши в неї блок PID Controller (рис. 5.1).

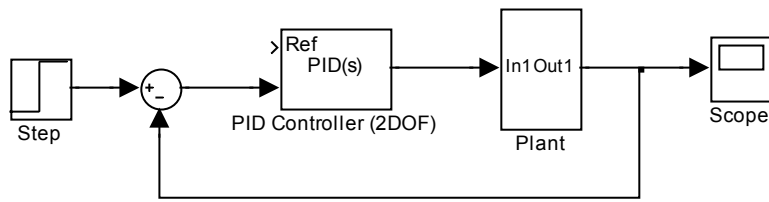


Рис. 5.1 – Схема автоматизованого налаштування ПІД-регулятора в інтерфейсі Simulink Control Design

Для того, щоб перейти до налаштування регулювальника необхідно відкрити діалогове вікно блоку PID Controller, показане на рис. 5.2.

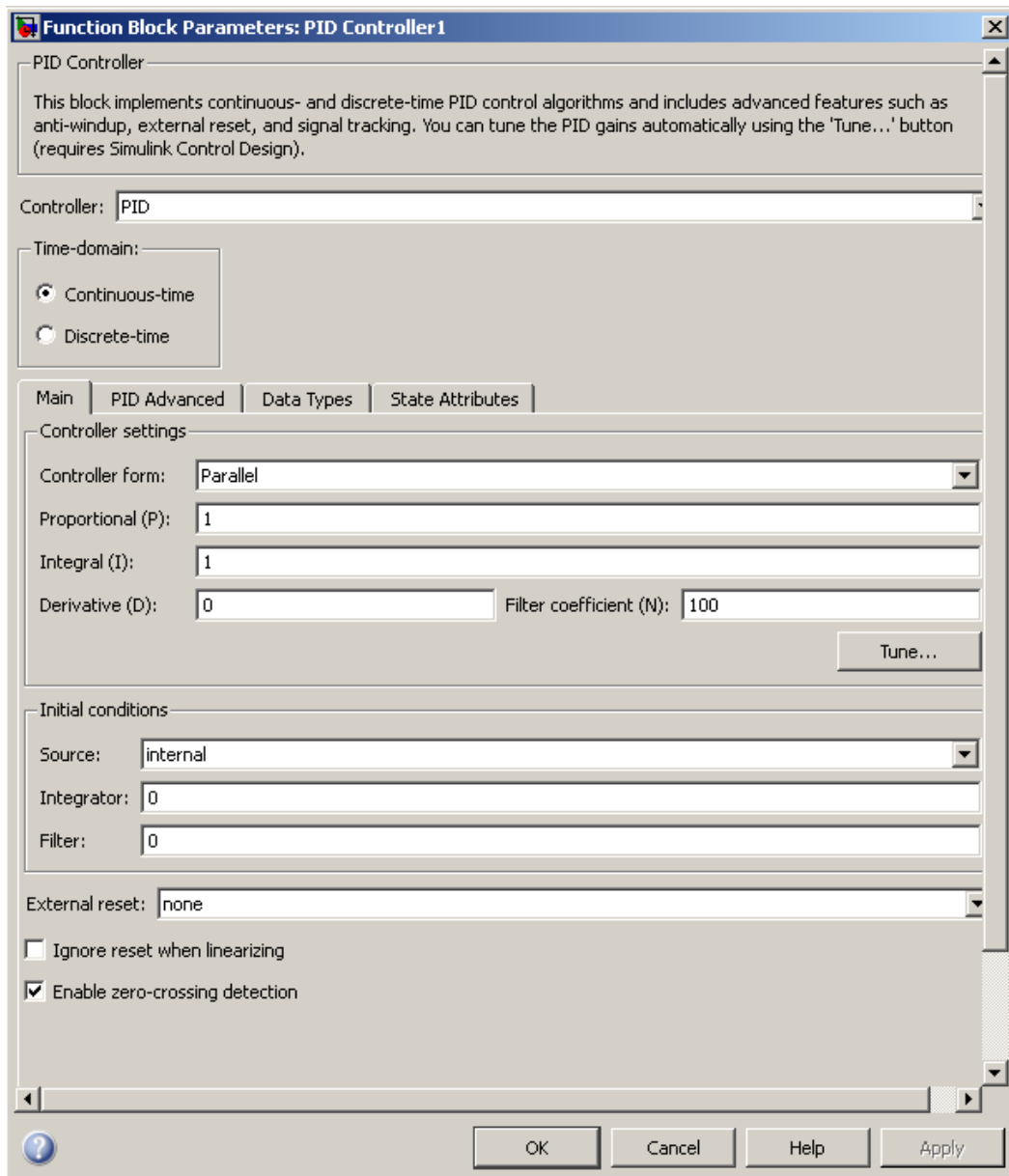


Рис. 5.2 – Діалогове вікно блоку PID Controller

У опціях блоку необхідно вказати тип регулювальника, (дискретний або безперервний), а також його структуру. Регулювальник може бути паралельним або ідеальним. По суті, це прив'язка структури до різного типу формулам. Коефіцієнти для різних типів регулювальника будуть різними.

Необхідно виставити початкові значення коефіцієнтів регулювальника або залишити те, що задано за умовчанням.

Щоб перейти до налаштування, слід натиснути кнопку Tune ("Налаштування"). Відкриється вікно PID Tuner (рис. 5.3). При цьому відразу спрацює алгоритм підстроювання параметрів регулятора для деякого автоматично вибраного часу перехідного процесу.

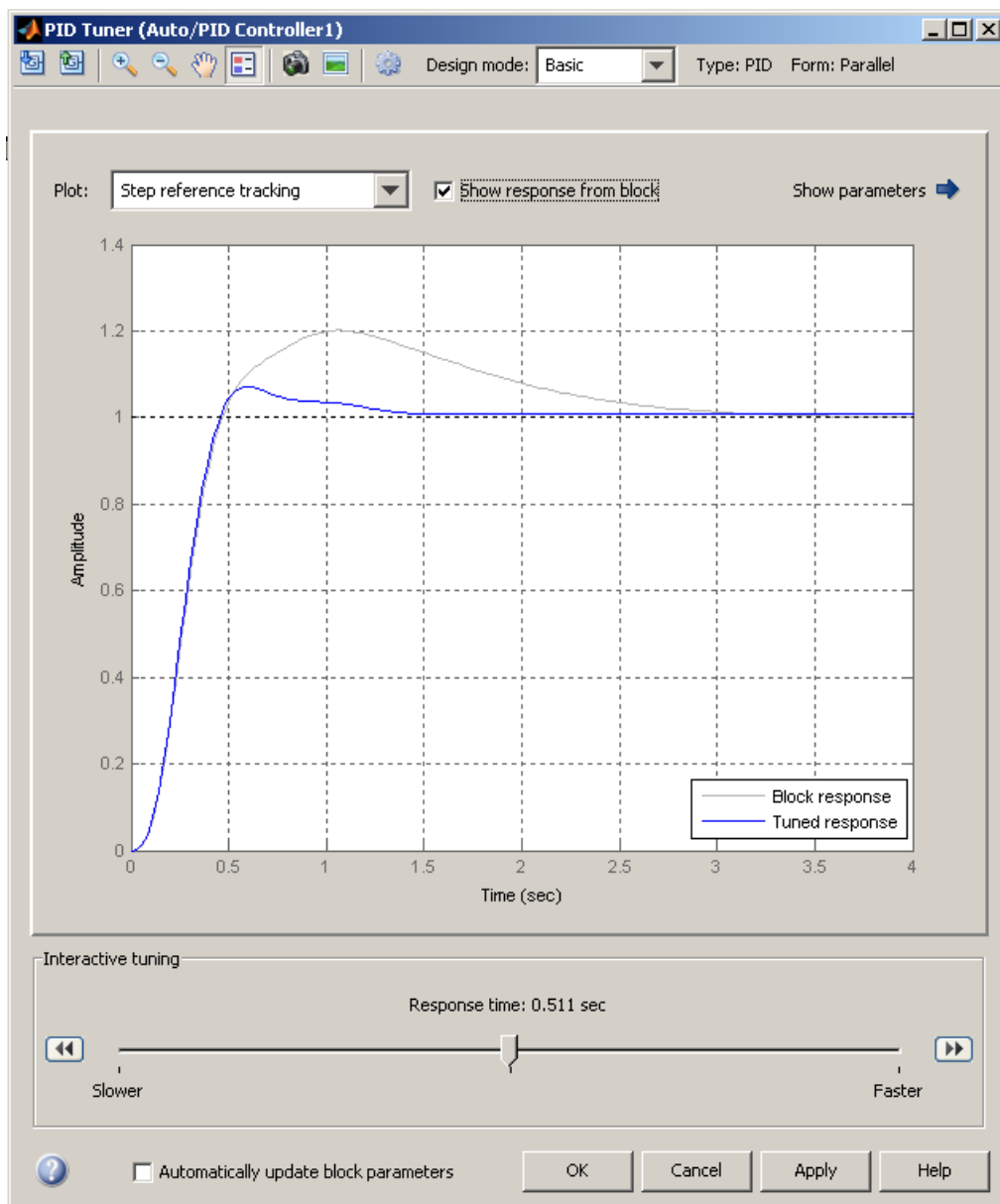


Рис. 5.3 – Діалогове вікно PID Tuner

Програма налаштування працює як з лінійними, так і з нелінійними моделями об'єктів. У другому випадку інтерфейс налаштування Simulink Control Design лінеаризує об'єкт в поточній робочій точці і будує лінійну модель об'єкту з постійними параметрами (LTI), з якою взаємодіє блок ПІД-регулятора в

замкнутому контурі управління. Використовуючи метод автоматичного налаштування, інтерфейс Simulink Control Design обчислює первинні коефіцієнти посилення ПД-регулятора. Цей метод не накладає ніяких обмежень на порядок об'єкту або постійну часу, причому він працює в областях як неперервного, так і дискретного часу.

### Порядок виконання лабораторної роботи

1. Створити S-модель системи управління, включивши в неї блок **PID Controller** ("ПД-регулятор") з бібліотеки Simulink/Continuous, як показано на рис. 5.1.

2. У моделі під ім'ям lab4.mdl, збереженої при виконанні роботи № 4, виділити три блоки **Transfer Fcn**, **Transfer Fcn1** та **Transport Delay** об'єкта управління. Клацнувши правою клавішею миші, вибрати з контекстного меню опцію Create Subsystem. Скопіювати створену підсистему у нову модель (п. 1) і дати підсистемі ім'я **Plant**. Зберегти модель під ім'ям lab5.

3. Двічі клацнувши по блоку **PID Controller**, відкрити діалогове вікно, в якому натиснути кнопку Tune. У вікні PID Tuner, пересуваючи повзунок Response Time, досягти бажаного виду перехідного процесу. Порівняти з перехідним процесом, отриманим у лабораторній роботі № 4.

4. Відкрити додаткове вікно, натиснувши на стрілку біля напису Show Parameters. Переписати параметри налаштування регулятора (Controller parameters) і характеристики перехідного процесу (Performance and robustness).

5 Побудувати різні види відгуку системи і її частин, вибираючи різні опції у випадаючому списку Response.

6 Побудувати діаграми Бode, вибравши у випадаючому списку Plot опцію Bode.

7. Звіт повинен містити мету роботи, передатну функцію об'єкта управління, схеми SIMULINK, таблицю з отриманими числовими параметрами, графіки перехідних характеристик для різних видів відгуку, діаграми Бode, а також висновки про проведені дослідження.

### Контрольні питання

1. Які параметри налаштування має ПД-регулятор?
2. Як впливає на характеристики перехідного процесу параметр  $K_P$ ?
3. Як впливає на характеристики перехідного процесу параметр  $K_D$ ?
4. Як впливає на характеристики перехідного процесу параметр  $K_I$ ?
5. Які види відгуку системи дозволяє спостерігати PID Tuner?
6. Які характеристики розраховує PID Tuner?

## Лабораторна робота № 6

### ОПТИМІЗАЦІЙНИЙ СИНТЕЗ РЕГУЛЯТОРА ЗА ДОПОМОГОЮ SIMULINK DESIGN OPTIMIZATION

**Мета роботи.** Опанувати методи оптимізації налаштування одноконтурних систем управління в середовищі SIMULINK

#### Можливості Simulink Design Optimization

Пакет *Simulink Design Optimization* забезпечує графічний інтерфейс користувача GUI (Graphical User Interface) для здійснення проектування систем управління (СУ) в часовій області. За його допомогою можна знайти оптимальні параметри налаштування для лінійної або нелінійної моделі системи в SIMULINK, які забезпечать потрібний перехідний процес по графічно заданих обмеженнях у вікні часової області. Будь-яке число змінних SIMULINK, включаючи скалярні, векторні і матричні величини, можуть бути оголошені такими, що налаштовуються або оптимізуються. Для цього потрібно ввести імена змінних у відповідному діалоговому боксі. Межі невизначеності можна задати для будь-яких змінних моделі робасної СУ.

Пакет робить процес досягнення необхідного перехідного процесу і оптимізації параметрів, що налаштовуються, інтуїтивним і легким. Для його використання необхідно лише включити в схему моделі SIMULINK спеціальний блок, який сполучається з необхідним сигналом моделі, щоб показати, що ви хочете встановити якісь обмеження на цей сигнал в часовій області.

Пакет автоматично перетворить обмеження в часовій області в обмеження задачі оптимізації параметрів СУ і потім вирішить задачу, використовуючи алгоритми і процедури, закладені в пакеті *Optimization*. Завдання оптимізації змінних з урахуванням накладених обмежень, послідовно викликає симуляцію роботи моделі СУ в SIMULINK, порівнює отримані результати із заданими обмеженнями перехідного процесу і, використовуючи градієнтний метод, змінює параметри, що налаштовуються, для досягнення заданої форми процесу.

Пакет дозволяє вводити невизначеність в завдання динамічних параметрів об'єкту управління, проводити симуляцію методом Монте Карло, задавати верхні і нижні межі змінних, що настроюються, і змінювати критерій досягнення мети.

Хід процесу оптимізації можна бачити в графічному вікні завдання обмежень прямо при виконанні симуляції.

#### Принципи роботи з пакетом

Розглянемо використання можливостей пакету *Simulink Design Optimization* для пошуку параметрів ПІД-регулятора, які задовольняють критеріям якості управління в часовій області.

Для синтезу регулятора необхідно:

- задати вхідний сигнал, наприклад одиничну ступінчасту дію;

- задати вимоги для сигналу на виході об'єкту управління як реакцію на вхідну дію;
  - запустити процес оптимізації вибраних параметрів регулятора.
- Розглянемо модель СУ з ПД-регулятором (рис. 6.1).

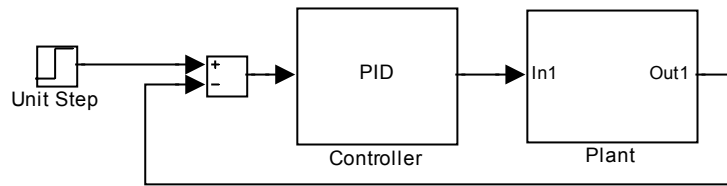


Рис. 6.1

Об'єкт управління представлений блоком **Plant**. Блок **Plant** є підсистемою, яка являє собою лінійну динамічну система другого порядку із запізнюванням, приклад якої показано на рис. 6.2.

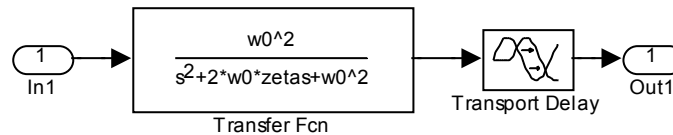


Рис. 6.2

Нехай реакція на системи на одиничний імпульс повинна відповідати наступним вимогам:

- перерегулювання не більше 10%;
- час наростання менш ніж 10 секунд;
- час встановлення менш ніж 30 секунд.

Додамо в модель оптимізаційний блок **Signal Constraint**. Для цього необхідно зайти у бібліотеку блоків *Simulink Design Optimization*.

Далі специфікуємо вимоги до вихідного сигналу системи в часовій області: час наростання, час встановлення і величина перерегулювання. Щоб ввести вимоги до якості управління, необхідно двічі клацнути по блоку **Signal Constraint**. Відкриється вікно, яке містить графік з обмеженнями для перехідного процесу. Відрізки ліній на графіку задають критерії якості управління – вимоги до реакції системи на вхідний сигнал (рис. 6.3). На цьому рисунку відрізки задають наступні вимоги до реакції на одиничну ступінчасту дію:

- амплітуда менша або є рівною  $-0.01$  до часу наростання 5 секунд, що відповідає недорегулюванню (*undershoot*) в 1%;
- амплітуда змінюється в межах між 0.9 і 1.2 до часу встановлення (*settling time*) 15 с;
- амплітуда не перевищує 1.2 до часу встановлення 15 с, що відповідає перерегулюванню (*overshoot*) в 20%;
- амплітуда знаходиться між 0.99 і 1.01 після часу встановлення, що відповідає сталій помилці (*settling*) в 2%.

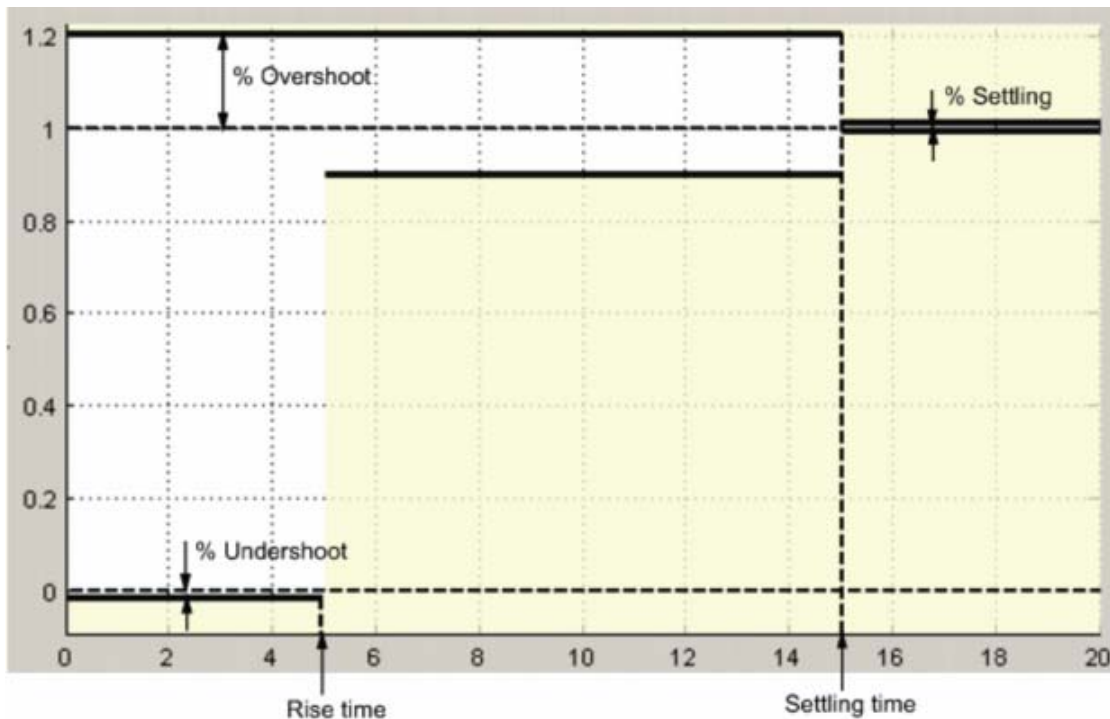


Рис. 6.3

Щоб змінити межі для осей графіка, необхідно клацнути правою кнопкою миші на графіці і вибрати *Axes Limits*.

Щоб вказати необхідні нам вимоги до якості регулювання, необхідно змінити час і амплітудні значення відрізків, як описано далі.

Двічі клацнемо жовту область на графіку. Ця дія відкриває діалогове вікно Edit Design Requirement, що показано на рис.6.4.

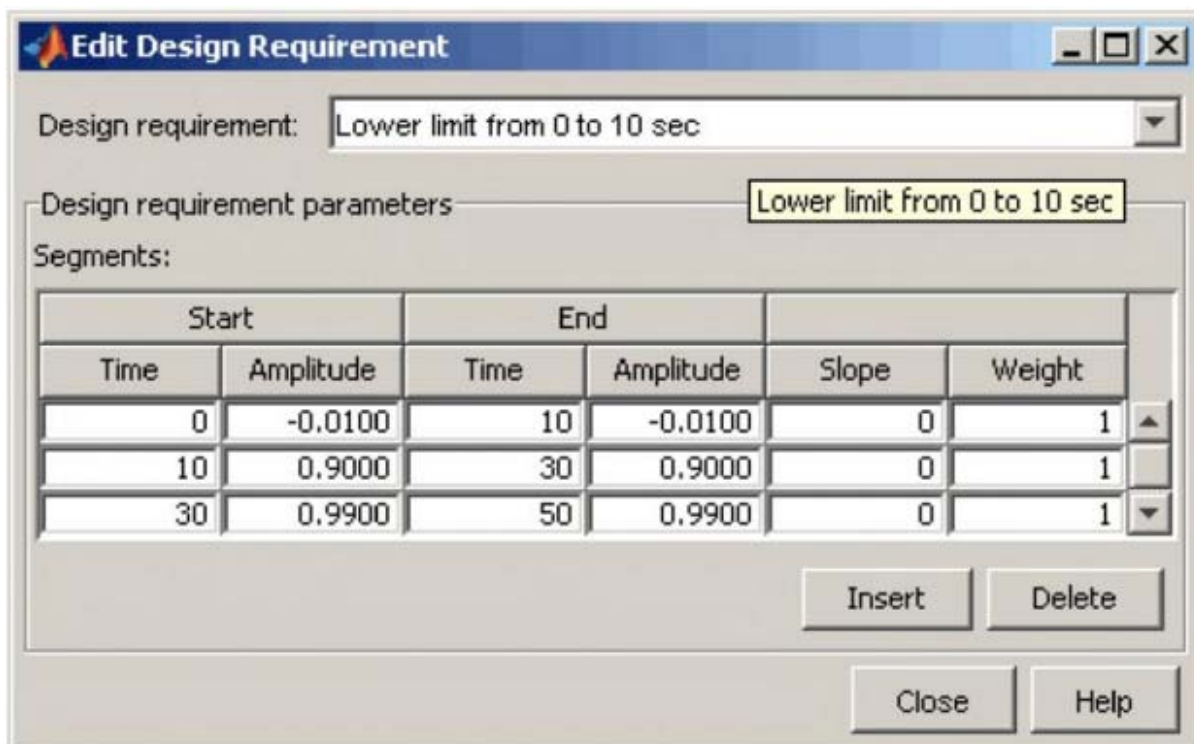


Рис. 6.4

У полі Design requirement можна вибрати Lower limit (Нижня межа) або Upper limit (Верхня межа). На рис. 6.4 показані координати кінців відрізків, що обмежують перехідний процес знизу. Аналогічно задаються параметри відрізків верхніх меж (рис. 6.5).

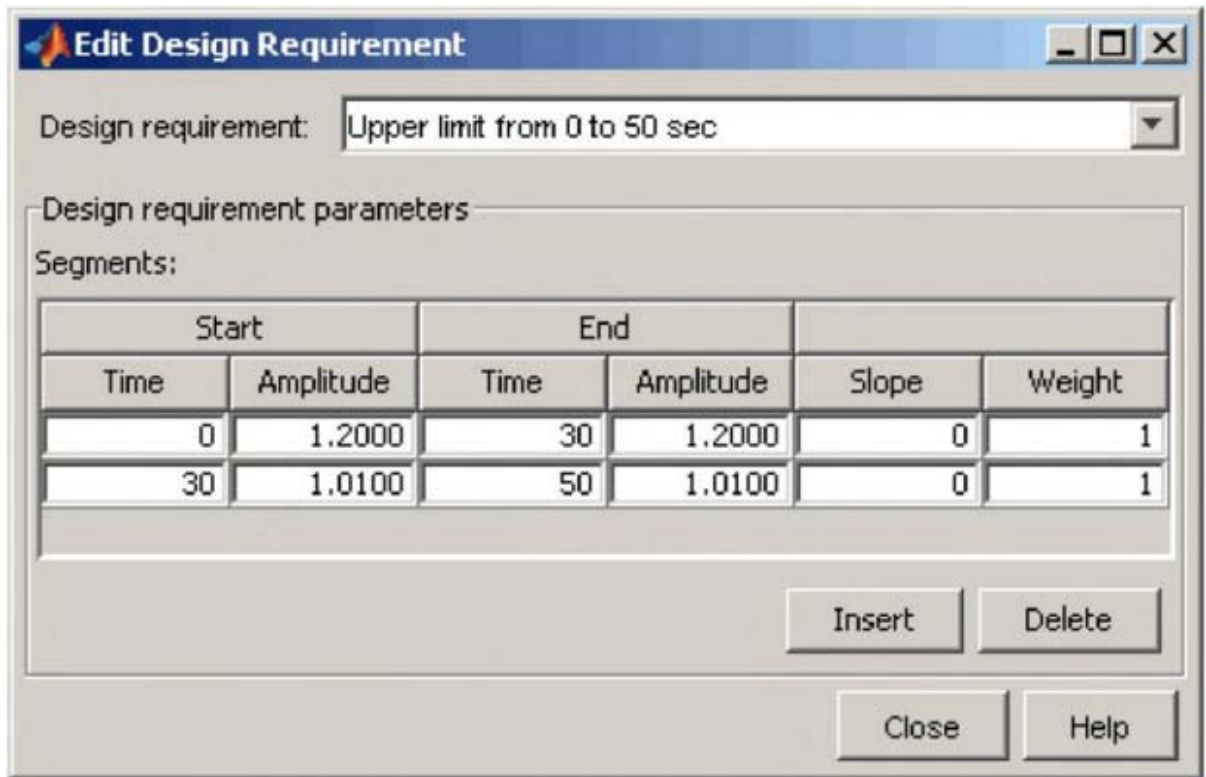


Рис.6.5

В результаті отримуємо специфікації на вихідний сигнал системи, представлені графічно у вікні параметрів Signal Constrain у відповідності до рис. 6.6.

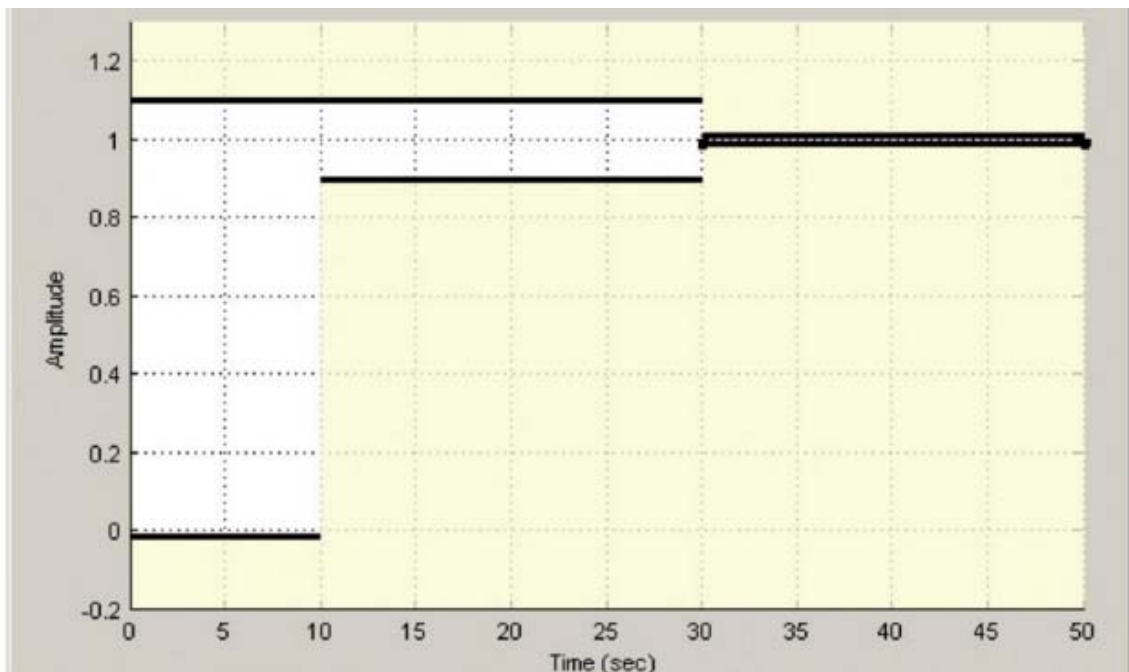


Рис.6.6



При оптимізації параметрів моделі програмне забезпечення *Simulink Design Optimization* змінює значення параметрів моделі для задоволення вимог якості регулювання, заданих обмежувальними лініями-специфікаціями.

Щоб задати параметри процедури оптимізації необхідно виконати наступні кроки. У вікні параметрів блоку *Signal Constraint*, виберемо у меню *Optimization* опцію *Tuned Parameters*. Ця дія відкриває діалогове вікно *Tuned Parameters* параметрів, що настраюються.

Натиснемо кнопку **Add**. Ця дія відкриває діалогове вікно *Add Parameters*. Це вікно відображає змінні з робочого простору (*workspace*) *MATLAB*. Змінні з робочого простору є параметрами моделі, що настраюються, тому вони мають бути попередньо задані, наприклад:

```
>>Kd = 0; Ki = 0; Kp = 1;
```

У діалоговому вікні *Add Parameter* виберемо змінні *Kd*, *Ki* та *Kp* і тиснемо кнопку **OK**. Ця дія додає *Kd*, *Ki* та *Kp* до вікна параметрів, що настраюються.

У правій частині вікна із заголовком *Optimization Settings* можна задати наступні налаштування оптимізації параметрів:

- а) *Value* – поточне значення параметра;
- б) *Initial Guess* – початкове значення параметра;
- в) *Minimum* і *Maximum* – межі зміни параметра;
- г) *TypicalValue* – коефіцієнт масштабування для параметра.

Щоб обмежити область зміни параметрів тільки позитивними значеннями, необхідно ввести в поле *Minimum* кожного параметра значення 0.

Запускаємо процес оптимізації параметрів моделі, вибравши в меню *Optimization* пункт *Start*. Ця дія відкриває вікно прогресу оптимізації. Під час кожної ітерації оптимізації програма запускає моделювання системи. Програма оптимізації (за умовчанням градієнтний спуск, але можна вибирати і інші) змінює параметри регулювальника з метою зменшити відстань між отриманою в результаті моделювання реакцією системи на виході і специфікаціями допустимих меж вихідного сигналу.

Після завершення оптимізації вікно *Optimization Progress* виглядає приблизно так, як показано на рис.6.7.

Повідомлення *Successful termination* вказує, що методом оптимізації знайдено рішення, яке відповідає вимогам до меж вихідного сигналу як вказано у властивостях блоку *Signal Constraint*.

У вікні налаштувань блоку *Signal Constraint* також виводяться графіки реакції моделі на вхідний сигнал протягом процесу оптимізації, щоб дозволяє перевірити, чи відповідає кінцева реакція (позначається чорним кольором) вимогам якості управління. Виводиться також графік первинної реакції системи в синьому кольорі.

Оптимізована реакція системи знаходиться в області, обмеженій сегментами ліній-специфікацій, і, таким чином, відповідає встановленим вимогам.

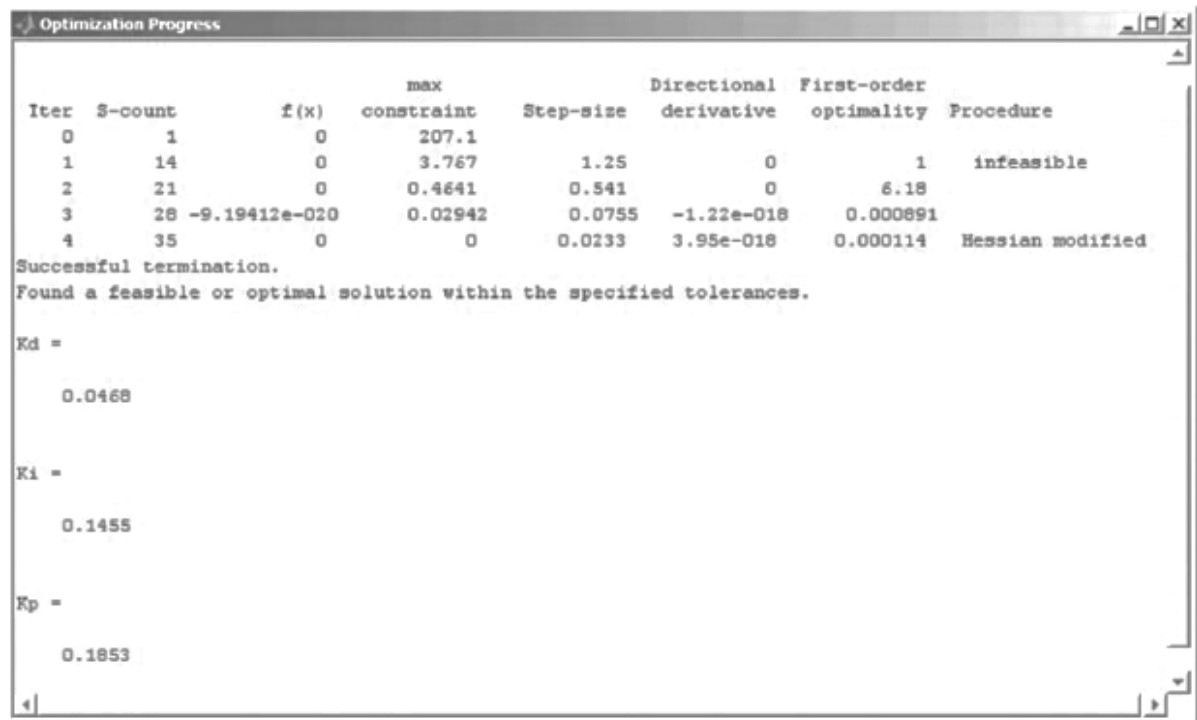


Рис. 6.7

### Синтез робасного ПД-регулювальника

Робасне управління – це загальний підхід до рішення завдань синтезу регулювальників, який в явному виді враховує структурні або параметричні невизначеності в об'єкті управління.

За допомогою *Simulink Design Optimization* можна задати конкретні межі зміни параметрів моделі і здійснити синтез регулювальника, який забезпечує задану якість управління при зміні вказаних параметрів об'єкту управління в заданих межах.

Для того, щоб здійснити налаштування регулювальника при наявності невизначеностей, необхідно виконати наступні кроки.

У вікні параметрів блоку Signal Constraint виберемо в меню Optimization опцію Uncertain Parameters. Ця дія відкриває діалогове вікно Uncertain Parameters (рис.6.8).

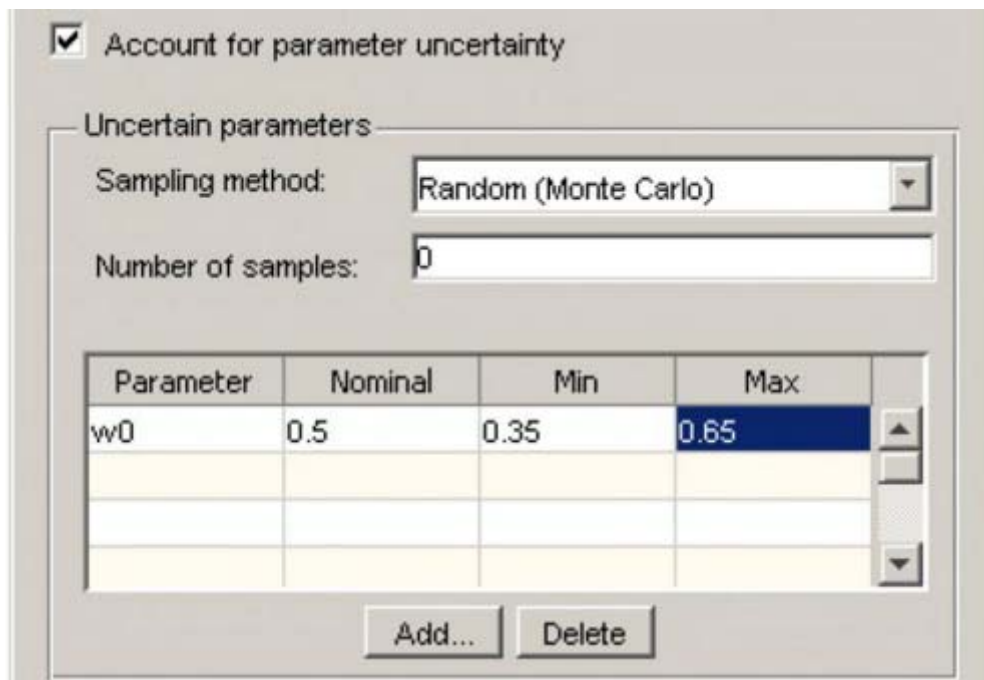


Рис. 6.8

У вікні налагоджуваних параметрів натиснемо кнопку **Add**. Ця дія відкриває діалогове вікно *Add Parameters*. По аналогії з вибором змінних для оптимізації, це вікно відображує усі змінні з робочого простору MATLAB, які при цьому не є параметрами, що оптимізуються (тобто ще не вибрані в *Tuned Parameters*). Після оголошення змінної  $w_0$ , аналогічно як це виконано з  $K_d$ ,  $K_i$  і  $K_p$ , і вибору її як невизначеної з межами зміни від 0.35 ДО 0.5, вікно *UncertainParameters* виглядає, як показано на рис.6.8.

Закривши вікно *Uncertain Parameters*, запустимо процес оптимізації. На відміну від попереднього випадку (без невизначеностей), на кожній ітерації оптимізації здійснюється генерування псевдовипадкової вибірки значень невизначених параметрів об'єкту управління, і моделювання виконується для кожного значення, тобто реалізується метод Monte Carlo.

Може виявитись, що усі обмеження тепер можуть бути і не досягнуті, але якщо максимальне відхилення від обмеження менше ніж 0.01, то, враховуючи міру невизначеності в динаміці об'єкту, такий результат можна вважати прийнятним.

### Порядок виконання лабораторної роботи

1. У S-моделі, отриманій в лабораторній роботі № 4, замінити числові значення параметрів передавальних функцій на відповідні імена змінних  $K$ ,  $T_1$ ,  $T_2$ . Додайте блок **Signal Constraint** (з бібліотеки *Simulink Desing Optimization*), як показано на рис. 6.9.

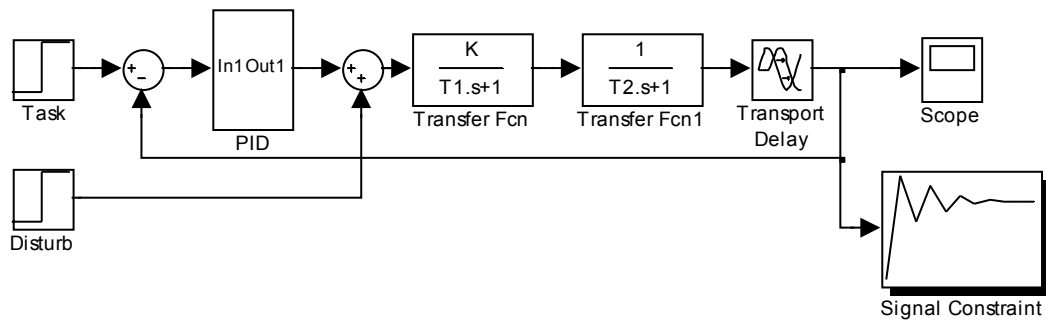


Рис. 6.9

2. Подвійним клацанням миші відкрити підсистему **PID** і замінити чисельні значення для блоків  $K_p$ ,  $K_i$  та  $K_d$  на відповідні імена змінних (див. рис. 6.10).

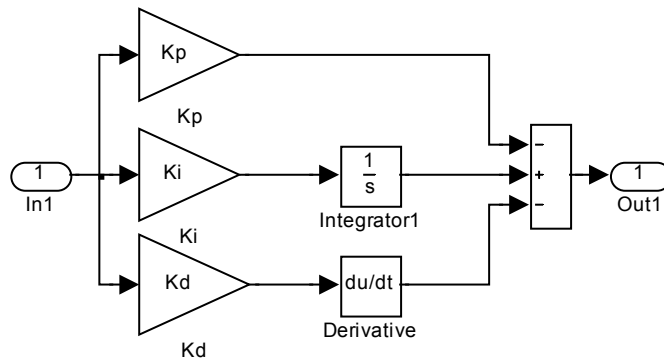


Рис. 6.10

3. Створити скрипт (m-файл) для завдання значень змінним, указаним у п.п. 1 і 2. Приблизний зміст файла наступний:

```
K=1.5
T1=10
T2=5
Kp=0.05
Ki=0.02
Kd=0
```

Параметри об'єкта беруться з табл. 4.1. Для приблизного розрахунку початкових значень налаштувань регулятора  $K_p$ ,  $K_i$  та  $K_d$  можна користуватись Додатком А.

Збережіть файл під іменем **init6.m**. Наберіть у командному вікні:

```
>>init6
```

4. Запустити моделювання кнопкою . Якщо вид процесу дуже далекий від бажаного, треба змінити початкові налаштування регулятора у файлі **init6.m** або у командному вікні Matlab

5. Двічі клацнувши мишею по блоку *Signal Constraint*, відкрийте його. Виберіть пункт *Tuned Parameters* меню *Optimization* і у вікні, що відкрилось, натисніть кнопку **Add** і виберіть по черзі  $K_p$ ,  $K_i$  та  $K_d$ .

Вибравши пункт *Simulation Options* меню *Optimization*, перевірте значення верхньої межі часу *Stop Time*. Воно повинне співпадати з часом моделювання S-моделі.

6. При необхідності змініть фігуру обмежень для перехідного процесу, переміщуючи обмежуючі лінії, наприклад, як показано на рис. 6.11.

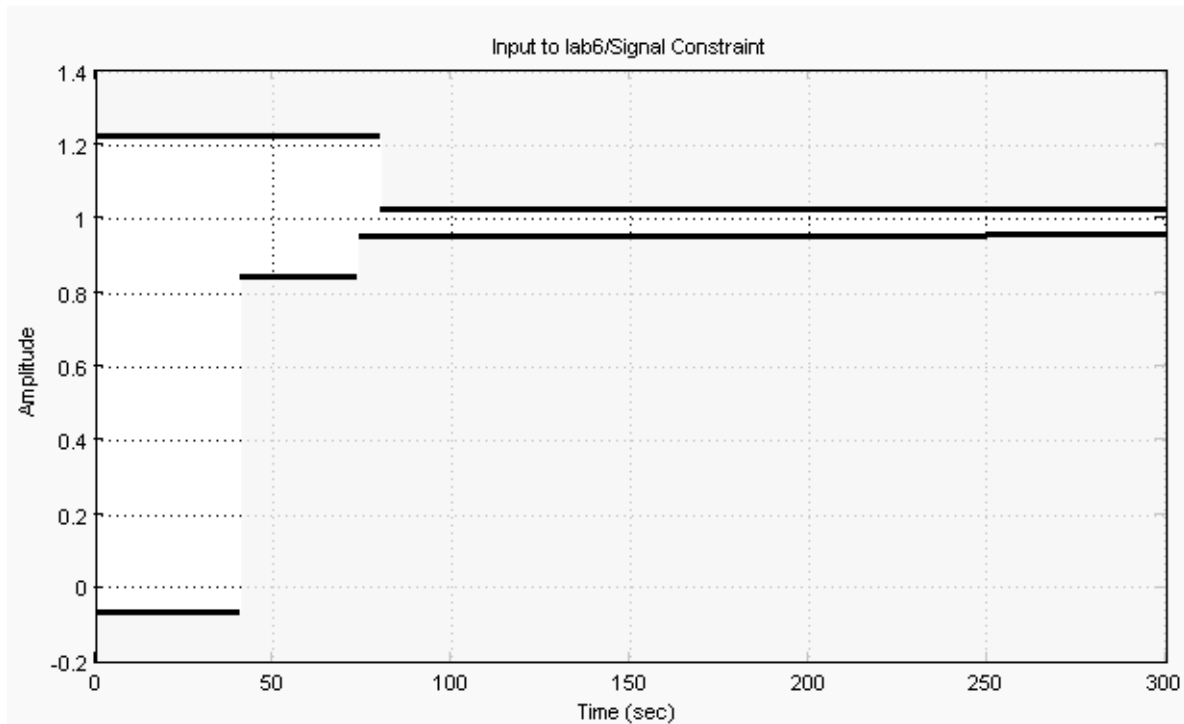



Рис. 6.11

7. Виберіть пункт *Start* меню *Optimization* або натисніть кнопку  на панелі управління вікна блоку **Signal Constraint**. По закінченню процесу оптимізації нові, оптимізовані значення параметрів налаштування регулятора можна вивести, набравши в робочому вікні MATLAB команду:

```
>>Kp, Ki, Kd
```

8. Задайте межі невизначеності для параметрів об'єкта. Для цього треба вибрати пункт *Uncertain Parameters* меню *Optimization*. Натискуючи кнопку *Add*, ввести параметри  $K$ ,  $T_1$ ,  $T_2$ . Для кожного параметра задати мінімальне і максимальне значення, що відрізняються на 5–10% від номінального значення.

Повторити процес автоматизації. Результатом є параметри робасного регулятора.

9. Запустити процес моделювання у вікні моделі. Порівняйте перехідний процес на осцилографі з результуючим графіком оптимізації.

10. Збережіть S-модель у своїй теці під ім'ям **lab6.mdl**.

11. Звіт повинен містити мету роботи, схеми S-моделей, початкові значення параметрів налаштування регулятора, лістинг файлу **init6.m**, графіки перехідних характеристик, копії вікон з результатами оптимізації, а також висновки про проведені дослідження.

### Контрольні питання для захисту

1. Склад і призначення пакету *Simulink Design Optimization*.
2. Запуск і робота з пакетом *Simulink Design Optimization*.
3. Завдання обмежень на вигляд перехідного процесу.
4. Завдання параметрів оптимізації перехідного процесу.
5. Виведення інформації про процес оптимізації.
6. Облік невизначеності параметрів моделі ОУ.

## Лабораторна робота № 7

### СИНТЕЗ І МОДЕЛЮВАННЯ ПОВНОГО СПОСТЕРІГАЧА

**Мета роботи:** опанувати методику синтезу модальної системи управління на основі спостерігача повного порядку

#### Теоретичні відомості

Для, щоб в системі управління можливо було використовувати зворотний зв'язок по стану об'єкта, необхідно оцінити вектор змінних стану, який часто є недоступним для виміру. Пристрої, що формують на виході вектор оцінки змінних стану, а також дозволяють відокремити корисний сигнал від шумів, називають спостерігачами.

Спостерігач може мати порядок, рівний порядку системи (спостерігач повного порядку), який оцінює стан об'єкта, враховуючи усі змінні стану, або менший (спостерігач зниженого порядку), який має розмірність на одну або кілька вимірюваних компонент менше, ніж система.

За відсутності завад за допомогою спостерігача зниженого порядку можна дуже точно оцінити координати вектора стану. Але в зашумлених системах такий спостерігач може невірно оцінити координати вектора стану, тому в умовах значного рівня завад і шумів використовують спостерігач повного порядку.

Нехай стаціонарний об'єкт описується традиційною системою рівнянь в просторі станів:

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}; \\ \mathbf{Y} = \mathbf{C}\mathbf{X}. \end{cases} \quad (7.1)$$

Припустимо, що матриці  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  відомі, тоді вектор  $\mathbf{x}$  можна замінити вектором  $\hat{\mathbf{x}}$  моделі яка має той же вхід, що і об'єкт (4.24).

$$\begin{cases} \dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}\mathbf{U}; \\ \mathbf{Y} = \mathbf{C}\hat{\mathbf{X}}. \end{cases} \quad (7.2)$$

Якщо модель (7.2) є ідеальною апроксимацією системи (7.1), в тому сенсі, що їх параметри і початкові умови ідентичні, то стани  $\mathbf{X}$  і  $\hat{\mathbf{X}}$  також співпадають. Якщо початкові умови для систем (7.1) і (7.2) різні, то  $\hat{\mathbf{X}}$  сходиться до  $\mathbf{X}$  тільки тоді, коли система (7.1) асимптотично стійка.

При відновленні по моделі (7.2) не використовується вимірюваний вихід. Якість відновлення покращується, якщо ввести в модель різницю вимірюного виходу і його оцінки  $\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}$  у вигляді зворотного зв'язку:

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}\mathbf{U} + \mathbf{L}(\mathbf{Y} - \mathbf{C}\hat{\mathbf{X}}). \quad (7.3)$$

Тут  $\mathbf{L}$  - деяка матриця, що забезпечує необхідний вигляд перехідних процесів оцінки вектору стану.

Вибором коефіцієнтів матриці  $\mathbf{L}$  можна забезпечити необхідний вигляд перехідних процесів спостерігача.

Загальний вигляд системи управління зі спостерігачем показаний на рис. 7.1.

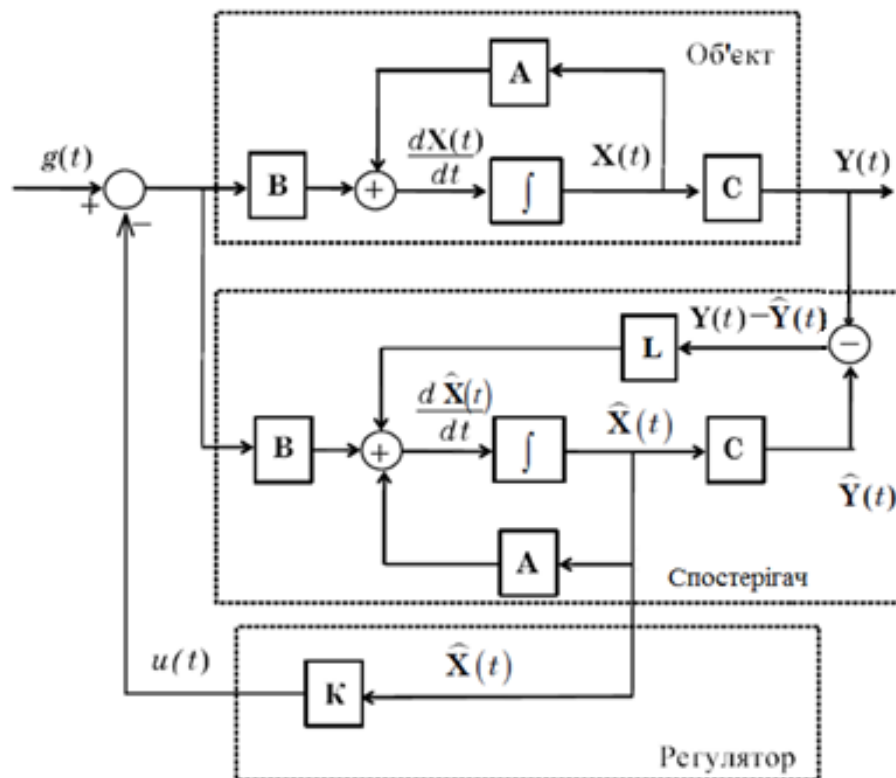


Рис. 7.1

Додаткове включення спостерігача впливає на загальну динаміку системи, збільшуючи тривалість перехідних процесів в порівнянні з тривалістю цих процесів в системі з регулятором стану без спостерігача. Тому необхідно при розрахунку системи одночасно розраховувати коефіцієнти матриць **K** та **L**. Це можна виконати за допомогою скрипту MATLAB наступного змісту:

```

W1=tf([1.2], [10 1]) % створення ПФ  $W_1(s)$ 
W2=tf([1], [3 1], 'ioDelay', 3) % створення ПФ із запізненням  $W_2(s)$ 
W=series(W1, W2) % послідовне з'єднання  $W_1(s)$  та  $W_2(s)$ 
sys=ss(W) % - модель об'єкта управління
[A,B,C,D]=ssdata(sys) % визначення матриць об'єкта
P=[-5 -5]; % бажані полюси спостерігача
K=acker(A,B,P); % матриця регулювальника
C_ = [C; 1,0] % модифікація матриці C, щоб отримати 2 виходи: x та y
D_ = [0; 0]; % теж саме для матриці D
L=acker(A',C',P); % матриця коефіцієнтів зворотних зв'язків спостерігача L
L=L' % транспонування матриці

```

У цьому скрипті використовується передавальна функція об'єкта з лабораторної роботи № 4. Скрипт треба скоректувати відповідно до номера варіанту з таблиці 4.1 і зберегти під ім'ям **lab7.m**.

## Порядок виконання роботи

1. Створити фрагмент S-моделі у відповідності з рівнянням динаміки спостерігача (7.3), як показано на рис. 7.2. Усі підсилювачі **Gain** налаштовані на матричне множення установкою параметра Multiplication: Matrix ( $K \cdot u$ ). Кожен з них виконує множення відповідно до рівняння (7.3): блок **Gain\_A** – множення  $A\hat{X}$ , блок **Gain\_B** – множення  $Bu$ , блок **Gain\_L** – множення  $L(Y - \hat{Y})$ , блок **Gain\_C** – множення  $\hat{Y} = C\hat{X}$ . Двічі клацнувши по кожному блоку, занести у поле *Gain* ім'я відповідної матриці **A**, **B**, **C**, **L**.

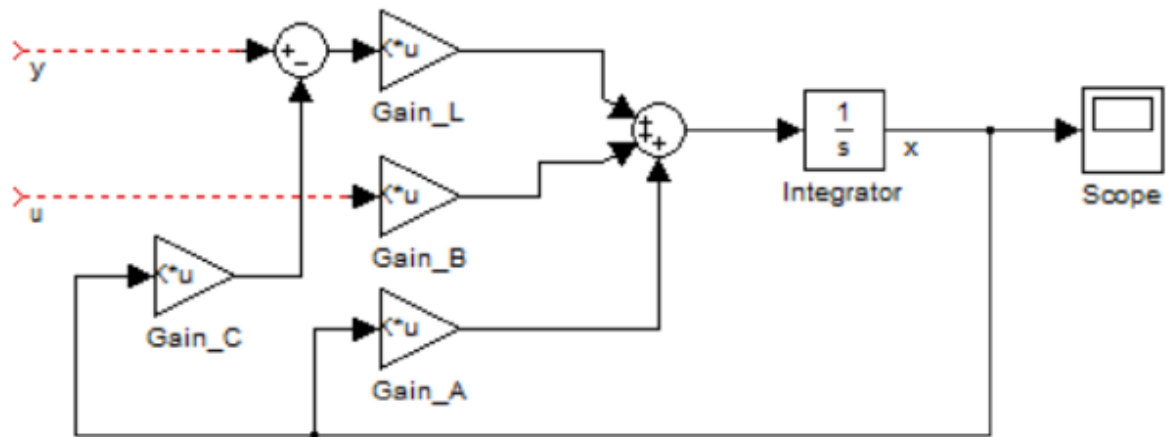


Рис. 7.2

2. Створити з її елементів підсистему, додавши проводи, відповідні входам спостерігача  $u$  і  $y$ , після чого виділивши усі елементи спостерігача (окрім осцилографа) і в контекстному меню по натисненню правої кнопки виберемо **Create Subsystem**. Після виконаного, усі виділені елементи виявляться в усередині блоку **Subsystem**, внутрішність якого розкривається але подвійному натисненню. Відкрийте підсистему і перейменуйте її входи і виходи, задавши осмислені імена блокам введення-виведення (зображені округленими прямокутниками): вхід **In1** –  $Y$ , вхід **In2** –  $U$ , вихід **Out1** –  $X^{\wedge}$  (рис. 7.3)

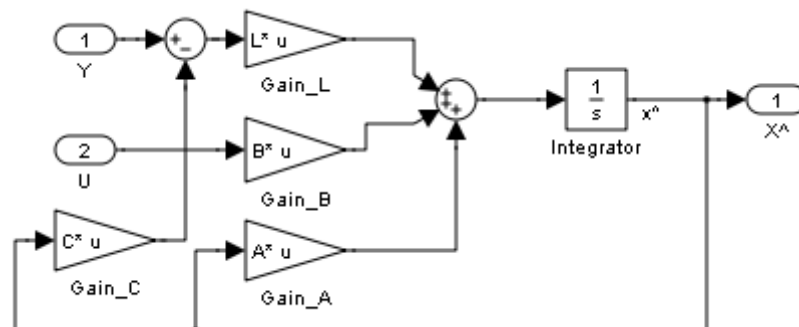


Рис. 7.3

3. Додати в початкову модель блок **State-Space** для моделювання об'єкта управління. Також додайте блоки демультиплексор **Demux** для витягання окремих виходів з вихідного вектора моделі  $[y, x]$ . На вхід подайте ступінчасте збурення, яке створюється блоком **Step**. Виходи контролюйте осцилографами **Scope**. Як регулятор використайте підсилювач **Gain\_K** (рис. 7.4).



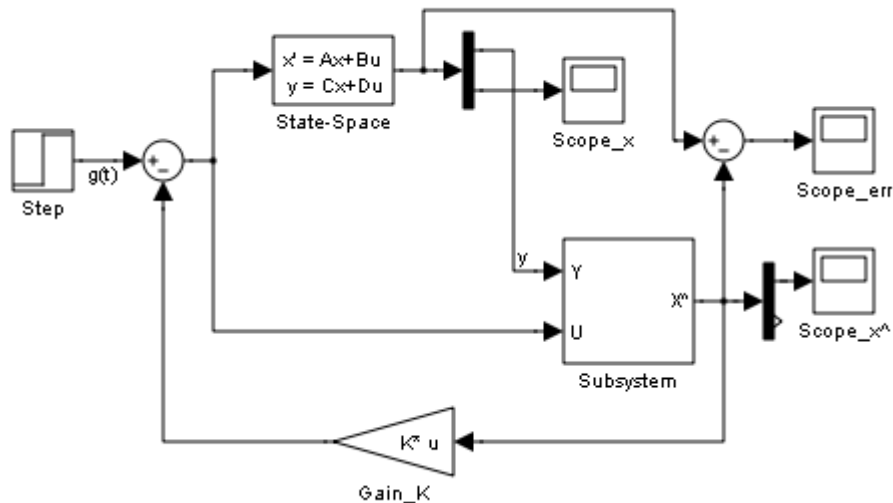



Рис. 7.4

4. У налаштуваннях блока **State-Space** ввести імена матриць **A**, **B**, **C**, **D**, для блока **Gain\_K** ввести ім'я матриці **K**.

5. Запустити на виконання скрипт **init7.m**. для формування у робочому просторі MATLAB усіх матриць.

6. Запустити моделювання кнопкою . Порівняти поведінку змінних стану об'єкта і моделі на екранах осцилографів **Scope\_x** та **Scope\_x^**. Осцилограф **Scope\_err** призначений для контролю помилки відтворення змінної стану **X** і виходу **Y**.

7. Звіт повинен містити мету роботи, схеми S-моделей, лістинг файлу **init7.m**, хід виконання цього файлу (у командному вікні), графіки перехідних характеристик, а також висновки про проведені дослідження.

### Контрольні питання

1. Що таке спостерігач повного порядку?
2. Що таке спостерігач зниженого порядку?
3. Для чого спостерігач вводиться у систему управління?
4. Яку роль виконує матриця **K** на схемі 7.1?
5. Яку роль виконує матриця **L** на схемі 7.1?
6. Яке призначення функції `acker(A',C',P)`?
7. Як впливає включення спостерігача на загальну динаміку системи?

## Лабораторна робота № 8

### СИСТЕМИ НЕЧІТКОГО РЕГУЛЮВАННЯ

**Мета роботи:** Опанувати методи синтезу регуляторів нечіткої логіки

#### Короткі теоретичні положення

Традиційний підхід до постановки і рішення завдань управління і регулювання ґрунтується на припущеннях, що модель об'єкту управління (ОУ) відома і задана у вигляді передатних функцій його окремих каналів або у формі системи диференціальних рівнянь, що зв'язують його вхідні і вихідні змінні, і що при відомій меті управління алгоритм функціонування керуючого пристрою однозначно зумовлюється цією моделлю і метою управління. Причому, вирішується це завдання синтезу алгоритму управління за допомогою строгих аналітичних методів математики.

Проте в реальній практиці автоматизації складних технологічних процесів і виробництв рідко вдається побудувати досить точну математичну модель об'єкту управління і застосувати строгую однозначну процедуру синтезу алгоритму управління. Тому більшість тих, що функціонують в промисловості автоматичних і автоматизованих систем управління створено з використанням не лише методів теорії управління, але і досвіду і інтуїції фахівців з автоматизації, евристичних знань технологів.

Принцип функціонування і методологія побудови системи фазі-регулювання показані на схемі на рис. 8.1.

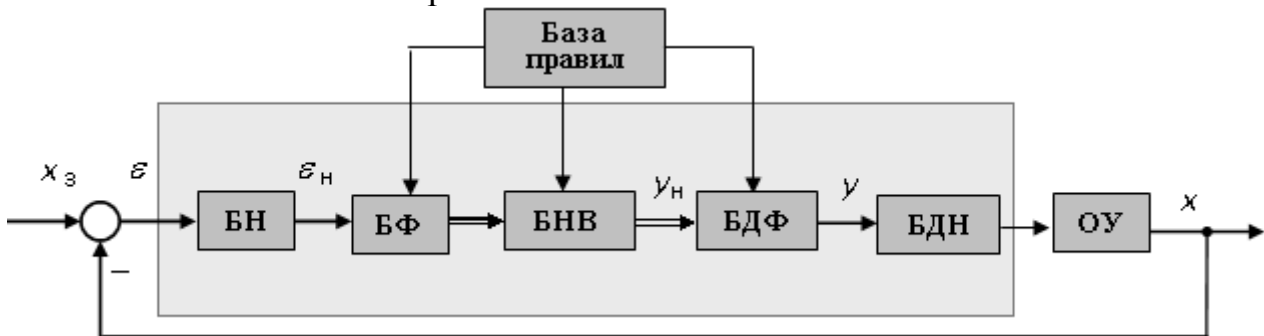


Рис. 8.1 – Функціональна структура системи фазі-регулювання

Уся початкова інформація про стратегію управління зберігається у базі правил (англ. – *rule-base*) у вигляді правил умовного логічного висновку "ЯКЩО..., ТО " (англ. – "IF..., THEN"), які формулюються на основі ретельного вивчення об'єкту і завдання управління, шляхом анкетного опитування фахівців-технологів, що добре знають об'єкт управління.

Центральною ланкою фазі-регулятора є блок нечіткого виводу (БНВ), або блок ухвалення рішень (англ. – *decision making*), в якому на основі нечіткої інформації про сигнал помилки регулювання  $\varepsilon$  робиться висновок про відповідну нечітку множину значень керуючої дії  $u$ . У блоці БНВ реалізується так звана інференц-процедура, в процесі якої агрегуються (об'єднуються) висновки окре-

мих правил про керуючу дію. Результатом агрегації є "усічена" нечітка множина керуючих дій.

У блоці нормування (БН) вимірний чіткий (англ. – crisp) сигнал помилки регулювання  $\varepsilon$  множенням на масштабний коефіцієнт  $k_{\varepsilon,н} < 1$  приводиться до обмеженого інтервалу (наприклад,  $[-1,+1]$ ). Блок фазифікації (БФ) визначає значення функції приналежності, що відповідає нормованому значенню  $\varepsilon_н$ .

Блоки дефазифікації (БДФ) і денормування (БДН) виконують зворотні процедури: по декількох "усічених" функціях приналежності обчислюється чітке значення нормованої керуючої дії  $y_н$  і ненормоване значення, що відповідає йому:  $y = y_н / k_{y,н}$  ( $k_{y,н} < 1$ ).

До нечітких висловлювань, використовуваних у фазі-логіці, застосовні операції алгебри логіки, якщо кожне висловлювання, наприклад,  $A$ , характеризувати функцією істинності (ФІ)  $\mu(A)$ , аналогічній в певному значенні функції приналежності.

Можуть бути введені наступні аналоги звичайних логічних операцій:

а) фазі-логічне заперечення

$$\mu(\neg A) = 1 - \mu(A), \quad (8.1)$$

б) фазі-логічна диз'юнкція

$$\mu(A \vee B) = \max \{ \mu(A), \mu(B) \}, \quad (8.2)$$

в) фазі-логічна кон'юнкція

$$\mu(A \wedge B) = \min \{ \mu(A), \mu(B) \}, \quad (8.3)$$

г) фазі-логічна імплікація

$$\mu(A \Rightarrow B) = \min \{ 1, 1 + \mu(B) - \mu(A) \}. \quad (8.4)$$

При використанні фазі-технології для синтезу систем управління алгоритм функціонування керуючого пристрою, а іноді і об'єкту управління ОУ, представляють як сукупність нечітких правил, яка в загальному випадку має наступний вигляд :

$$\begin{aligned} & \{ \text{ЯКЩО... [посилка 1], ТО... [висновок 1] } \text{ АБО} \\ & \{ \text{ЯКЩО... [посилка 2], ТО... [висновок 2] } \text{ АБО ...} \\ & \{ \text{ЯКЩО... [посилка j], ТО... [висновок j] } \text{ АБО} \end{aligned} \quad (8.5)$$

.....

$$\{ \text{ЯКЩО... [посилка n], ТО... [висновок n] } \},$$

причому частина "ЯКЩО" є зазвичай висловлювання про нечітке (лінгвістичне) значення вхідних змінних, змінних стану  $x_i(t)$  об'єкту або помилки регулювання  $e_i(t)$ , а частина "ТО" – висловлювання про відповідне значення керуючої дії  $y_i(t)$ . Часто кожна посилка містить висловлювання відносно двох і більше вхідних змінних. сполучені союзом "І" або союзом "АБО" Наприклад:

ЯКЩО помилка регулювання  $\varepsilon(t)$  позитивно мала (PS) І похідна помилки  $\dot{\varepsilon}(t)$  позитивно велика (PB), ТО керуюча дія  $y_j(t)$  негативно середня (NM).

Складові частини посилок кожного  $j$ -го правила, сполучені між собою союзами "І", "АБО" і такі, що характеризується функціями приналежності  $\mu_j(x_i)$ , утворюють фазі-множину  $j$ -ої посилки, загальна ФІ якого визначається за допомогою формул (8.2) для союзу "АБО" і (8.3) відповідно для союзу "І".

Для знаходження ФІ  $\mu_j(x, y_i)$ , що характеризує конкретне  $j$ -е правило нечіткого умовного логічного висновку (англ. *Conditional Rules Inference*, скоро-

чено *CRI*), запропоновані декілька формул, серед яких найбільше поширення отримало правило імплікації Мамдані:

$$\mu_j''(x, y_j) = \min\{\mu_j(x), \mu_l(y_j)\}, \quad (8.6)$$

де  $\mu_j''(x, y_j)$  - ФП, що характеризує  $j$ -е правило і задана на прямому (декартовому) добутку множин  $X \times Y$ ;

$\mu_l(y_j)$  - ФП, що характеризує  $l$ -е лінгвістичне значення керуючої дії  $y$  в  $j$ -му правилі;

$x$  - вектор вхідних змінних.

Формула (8.6) отримана, виходячи з гіпотези, що наслідок кожного правила має істинне значення, не менше, ніж відповідна передпосилка.

Формулу сумісної ФП (8.6), що відповідає конкретному правилу "ЯКЩО..., ТО..." називають оператором імплікації.

При фіксованих значеннях  $x^*$  вибору мінімальних значень  $\mu_j(x^*)$  та  $\mu_l(y_j)$ , заданих на прямому добутку  $X \times Y$ , відповідає обмеження (усікання) графіків  $\mu_l(y_j)$  згори на рівні  $\mu_j(x^*) < 1$ .

Результуюча функція приналежності  $\mu_p(y)$ , яка характеризує всю сукупність  $j$ -х правил, сполучених між собою союзом "АБО", визначається відповідно до формули (8.2) як максимум серед усіх функцій (8.6) при заданих  $x^*$ , тобто

$$\mu_p(x^*, y) = \mu_p^*(y) = \max\{\mu_j''(x^*, y_j)\}, \quad j = 1, \dots, n. \quad (8.7)$$

Процедура об'єднання усіх правил називається інференц-процедурою (англ. – *composition of rules*). У разі спільного застосування формул (8.6) і (8.7) вона називається максимінною інференц-процедурою.

Завершальна процедура синтезу керуючого пристрою за фазі-технологією – дефазіфікація – зводиться до визначення точного значення керуючої дії  $y$ . Вона виконується найчастіше по методу центру площі, згідно з яким для неперервної функції  $\mu_p^*(y)$  шукане значення керуючої дії визначається як абсциса центру тяжіння площі фігури, створеною цією функцією і віссю  $y$ :

$$y^* = \int y \mu_p^*(y) dy / \int \mu_p^*(y) dy. \quad (8.8)$$

У фазі-технології відомі форми запису правил (8.5), у яких в частині "ТО" вказується не лінгвістичне значення керуючої дії (наприклад,  $y = PB$ ), а його конкретне, чітке значення (наприклад,  $y = y^*$ ). Так, за методологією Сугено і Тагаки  $j$ -е правило логічного висновку записується в наступній формі:

$$\text{ЯКЩО } x_1 = A_{j1} \text{ I } \dots \text{ I } x_k = A_{jk}, \text{ ТО } y_j = c_{j0} + c_{j1} + \dots + c_{jk}, \quad (8.9)$$

де  $A_{j1} \dots A_{jk}$  – фазі-множини типу NB, NM і т.д.;

$c_{j0}, c_{j1}, \dots, c_{jk}$  – постійні (чіткі) значення керуючої дії, задані у вигляді синглетонів, тобто у вигляді пар  $(y_j, \mu(y_j) = 1)$ .

Форма запису (8.9) називається моделлю Сугено нульового порядку.

Іноді застосовують модель Сугено першого порядку:

$$\text{ЯКЩО } x_1 = A_{j1} \text{ I } \dots \text{ I } x_k = A_{jk}, \text{ ТО } y_j = c_{j0} + c_{j1}x_1 + \dots + c_{jk}x_k. \quad (8.10)$$

Об'єднання складових частин посилок в (8.9) і (8.10) здійснюється так само, як і в методі Мамдані – за допомогою процедур максимізації (8.2) і мінімізації (8.3) відповідних ФП. Агрегація усіх  $n$   $j$ -х правил, сполучених між собою союзом АБО, відбувається теж аналогічно – по формулі (8.2). Але процедура імплікації і інференц-процедура, що виконуються в методі Мамдані по форму-

лах (8.6) і (8.7), тут в явному виді відсутні. Максимальні і мінімальні значення ФП  $\mu_j(x)$ , характеризуючі посилки правил, множаться на відповідні синглтони керуючої дії.

Результуюче значення керуючої дії, що відповідає усім  $n$  правилам, по Сугено визначається формулою

$$y = \frac{\sum_{j=1}^n \mu_j y_j}{\sum_{j=1}^n \mu_j} \quad (8.11)$$

Таким чином, складна процедура дефазифікації у чистому вигляді, наприклад, згідно формули (8.8), в методі Сугено-Такагі відсутня.

### Загальні відомості про Fuzzy Logic Toolbox

Одна з цілей пакету Fuzzy Logic Toolbox полягає в полегшенні розуміння і створення фазі-систем управління. Фазі-систему можна створювати, використовуючи командний рядок головного вікна MATLAB. Проте набагато простіше зробити це графічно за допомогою інструментів графічного інтерфейсу користувача (GUI або ГИП), що входить у Fuzzy Logic Toolbox. Є п'ять інструментів GUI для створення, редагування і спостереження фазі-систем у середовищі Fuzzy Logic Toolbox (рис. 8.2).



Рис. 8.2 – Структура фазі-інференційної системи – FIS

Це *Fuzzy Inference System Editor* або *FIS Editor* – Редактор фазі-інференційної системи, *Membership Function Editor* - Редактор функцій приналежності, *Rule Editor* - Редактор правил, *Rule Viewer* - Перегляд правил і *Surface Viewer* - Перегляд поверхні (простору управління). Ці інструменти GUI динамічно пов'язані між собою: використовуючи один з них у FIS і роблячи зміни, можна потім побачити їх дії в інших інструментах GUI. На додаток до цих п'яти інструментів Fuzzy Logic Toolbox включає графічний редактор GUI ANFIS,

який використовується для створення і аналізу Sugeno -типа адаптивних нейро-систем з нечіткою логікою – adaptive neuro-FIS (ANFIS).

FIS-редактор вирішує проблеми верхнього рівня системи: скільки вхідних і вихідних змінних і які їх імена?

Fuzzy Logic Toolbox не має обмежень по числу входів фазі-системи – все залежить від доступної пам'яті комп'ютера.

*Membership Function Editor* використовується для того, щоб визначити види усіх функцій приналежності, пов'язаних з кожною змінною.

*Rule Editor* використовується для редагування списку правил, який визначає функціонування фазі-системи.

*Rule Viewer* і *Surface Viewer* використовуються тільки для перегляду результатів, але не редагування фазі-системи. *Rule Viewer* показує діаграму функціонування. Використовуючи його як діагностичний засіб, можна побачити, наприклад, які правила активні, або як індивідуальні функції приналежності впливають на результати. *Surface Viewer* використовується для показу залежності виходу від одного або двох з входів системи, тобто показується діаграма поверхні простору управління системи. Усі п'ять інструментів GUI можуть взаємодіяти і обмінюватися інформацією між собою. Будь-який з них може прочитувати і зберігати дані в робочому просторі MATLAB і на жорсткому диску комп'ютера.

Розглянемо порядок створення фазі-системи на прикладі системи регулювання рівня рідини (рис. 8.3).

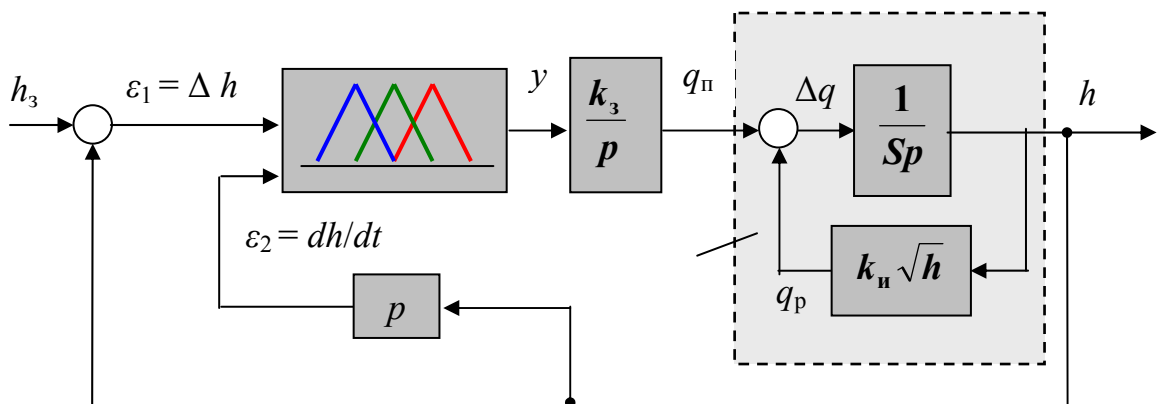


Рис. 8.3 – Структура системи фазі- регулювання рівня рідини

ОУ представлений ідеальною інтегруючою ланкою, яка охоплена нелінійним негативним зворотним зв'язком, що відповідає закону витікання рідини з резервуару. Резервуар характеризується коефіцієнтом  $1/S$ , де  $S$  - площа поверхні рідини,  $m^2$ . Подання рідини в резервуар  $q_n$  регулюється за допомогою засувки (3), степінь відкриття якої  $\varphi$  пропорційна  $q_n$ . Як керуюча дія у розглядається швидкість переміщення засувки, тобто  $y(t) = d(\varphi(t))/dt$ .

Якщо є можливість виміру швидкості зміни рівня, то один з простих алгоритмів управління може бути представлений у вигляді наступних п'яти нечітких правил:

ЯКЩО рівень нормальний ( $\varepsilon_1 = ZR$ ), ТО положення засувки не змінювати ( $y = ZR$ ) АБО

ЯКЩО рівень низький ( $\varepsilon_1 = PB$ ), ТО засувку відкривати швидко ( $y = PB$ ) АБО

ЯКЩО рівень високий ( $\varepsilon_1 = NB$ ), ТО засувку закривати швидко ( $y = NB$ ) АБО

ЯКЩО рівень нормальний ( $\varepsilon_1 = ZR$ ) І він збільшується ( $\varepsilon_2 = PB$ ), ТО засувку закривати повільно ( $y = NM$ ) АБО

ЯКЩО рівень нормальний ( $\varepsilon_1 = ZR$ ) І він зменшується ( $\varepsilon_2 = NB$ ), ТО засувку відкривати повільно ( $y = PM$ )

Побудову системи розпочинають з визначення кількості входів і виходів у редакторі властивостей фазі-системи – FIS Editor. Для його виклику необхідно в робочому вікні MATLAB ввести команду **fuzzy** з вказівкою редагованої (створеною раніше) FIS регулювання рівня - tank, тобто fuzzy tank.

Вікно редактора FIS Editor показано на рис. 8.4. У ньому вказуються назви усіх входів (input) та виходів (output). Для нашої системи це – відхилення рівня (level), швидкість зміни рівня (rate), швидкість переміщення засувки (valve). Нижче діаграми - назва системи (tank) і типу використовуваного виводу (mamdani АБО sugeno). Ще нижче вказані методи фазі-процедур:

- а) операції "І" (AND) - добуток (product - prod);
- б) операції "АБО"(OR) - імовірнісне АБО (probabilistic or – **probor**);
- в) імплікації (**prod**);
- г) агрегації (**max**);
- д) дефазифікації (**centroid**).

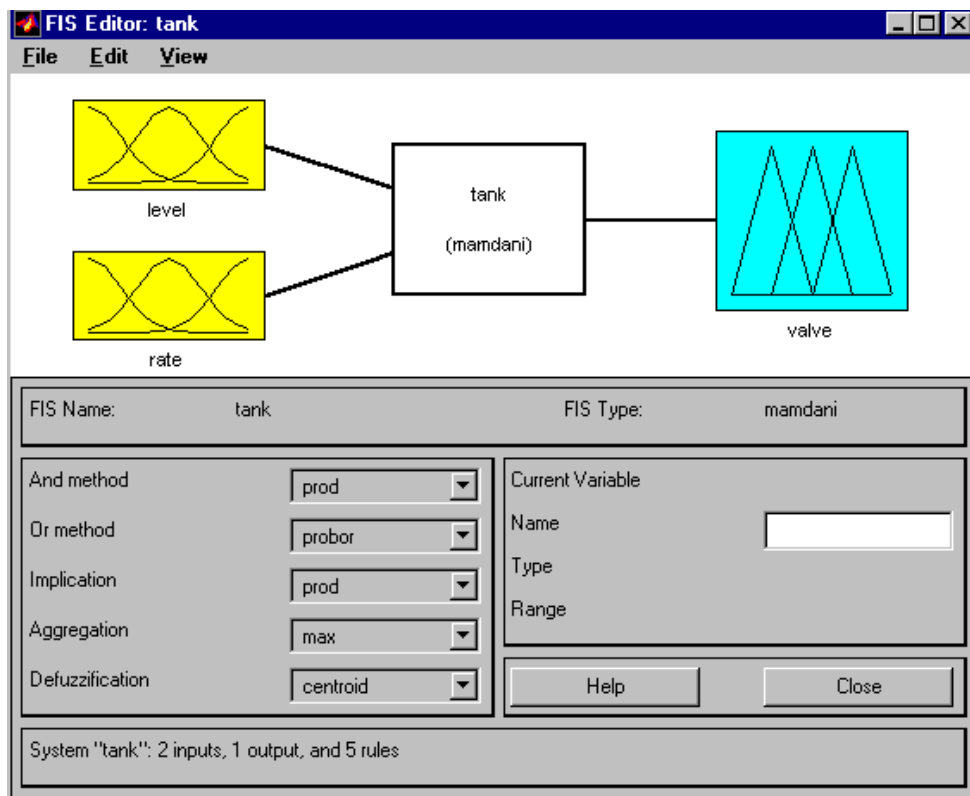


Рис. 8.4 – Вікно редактора FIS Editor

Fuzzy Logic Toolbox підтримує два вбудовані методи AND – *min* (*minimum*) та *prod* (*product*); два вбудовані методи OR – *max* (*maximum*) та *probor* (*probabilistic OR*), який визначається формулою підсумовування ймовірностей:

$$\text{probor}(a,b) = a + b - ab. \quad (8.16)$$

Для об'єднання або агрегації усіх правил Fuzzy Logic Toolbox використовує три вбудовані методи: **max** (*maximum*), *probor* (*probabilistic OR*) і *sum* (проста сума усіх виходів встановлених правил).

Для дефазікації використовуються п'ять вбудованих методів:

- а) *centroid* – координата (абсциса) центру тяжіння фігури під кривою;
- б) *bisector* – координата (абсциса), що ділить площу фігури під кривою навпіл);
- в) *mom* (*mean of the maximum*) – середина інтервалу максимальних значень);
- г) *lom* (*largest of maximum*) – верхня межа інтервалу максимальних значень);
- д) *som* (*smallest of maximum*) – нижня межа інтервалу максимальних значень).

Для ілюстрації на рис. 8.5 приведені різні результати дефазікації, проведеної перерахованими методами.

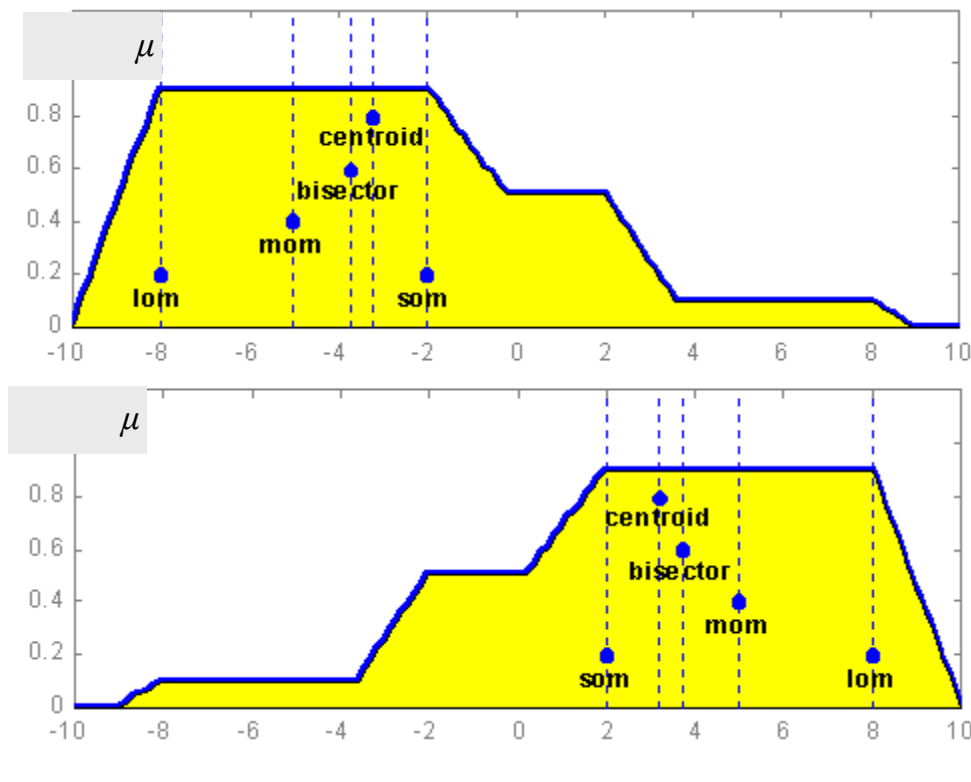


Рис. 8.5 – Методи дефазікації

Значення вихідної величини *y* на нижньому графіку рис. 8.5 складають 2, 3.3, 3.7, 5, 8 для методів дефазікації *som*, *centroid*, *bisector*, *mom* і *lom*, відповідно.



Найчастіше використовуваним методом дефазікації являється метод *centroid*, який визначає абсцису центру тяжіння плоскої фігури під кривою.

При необхідності можна застосовувати і інші методи, задані самим користувачем.

Кнопка допомоги **Help** викликає вікно допомоги MATLAB. У ньому дається опис *FIS Editor* і роботи з ним.

Для завдання функцій приналежності змінних використовується редактор *Membership Function Editor*, який викликається пунктом **Edit membership functions** меню **View** або подвійним клацанням по одному з входів або виходів діаграми FIS (рис. 8.6).

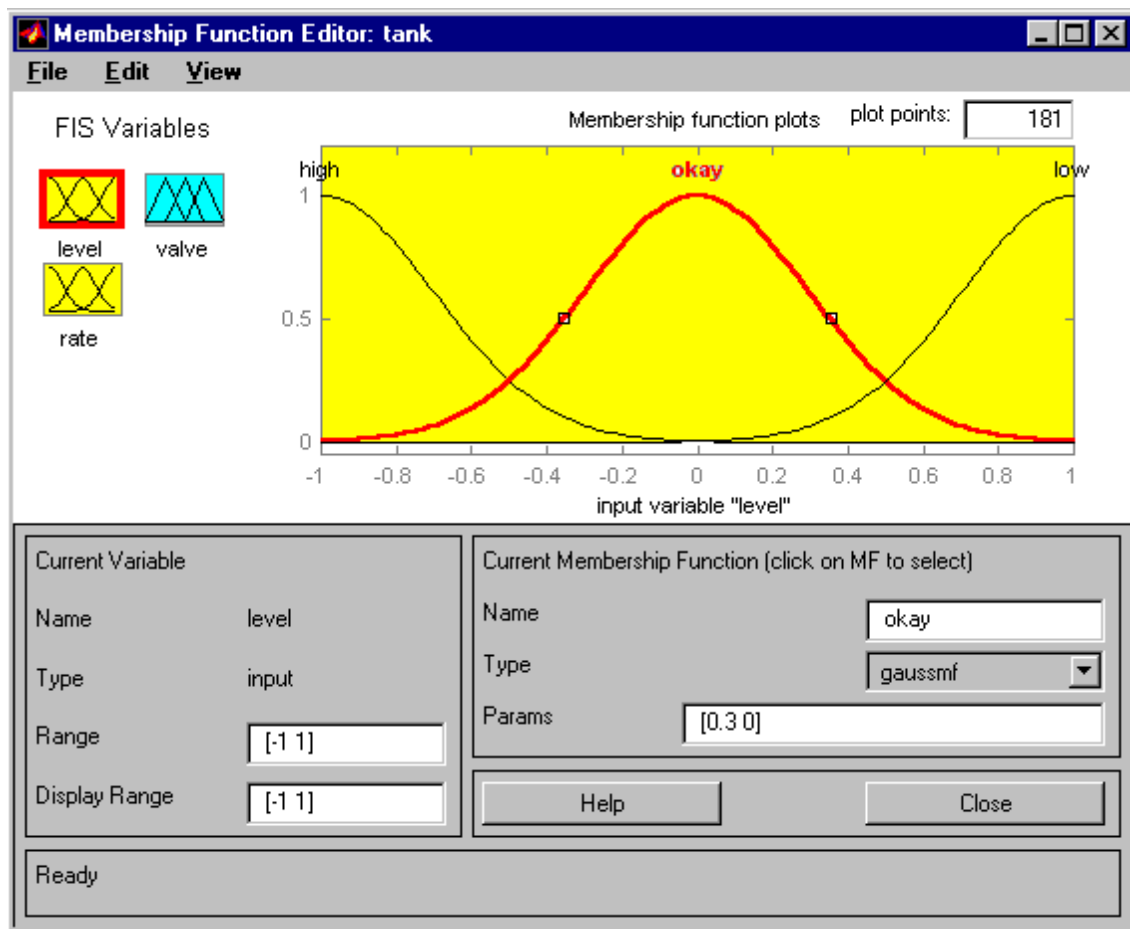


Рис. 8.6 – Вікно редактора Membership Function Editor

У цьому редакторі клацанням миші по блоку в лівій частині вибирається змінна, і її функції приналежності з'являються на графіку в правій верхній частині вікна редактора. При цьому в лівій нижній частині відбиваються ім'я (*Name*) – «Level», тип (*Type*) – «Input», межі зміни (*Range*) [-1 1] і межі відображення (*Display Range*) [-1 1] активованої змінної.

З меню **Edit** командою **Add MFs** можна додати нову функцію приналежності. Клацанням миші по графіку функції приналежності можна її активувати, потім в правій нижній частині вікна задати її ім'я і вибрати тип із спадаючого списку вбудованих функцій. Цей список досить великий, і графіки, що показують їх вид, приведені на рис. 8.7.

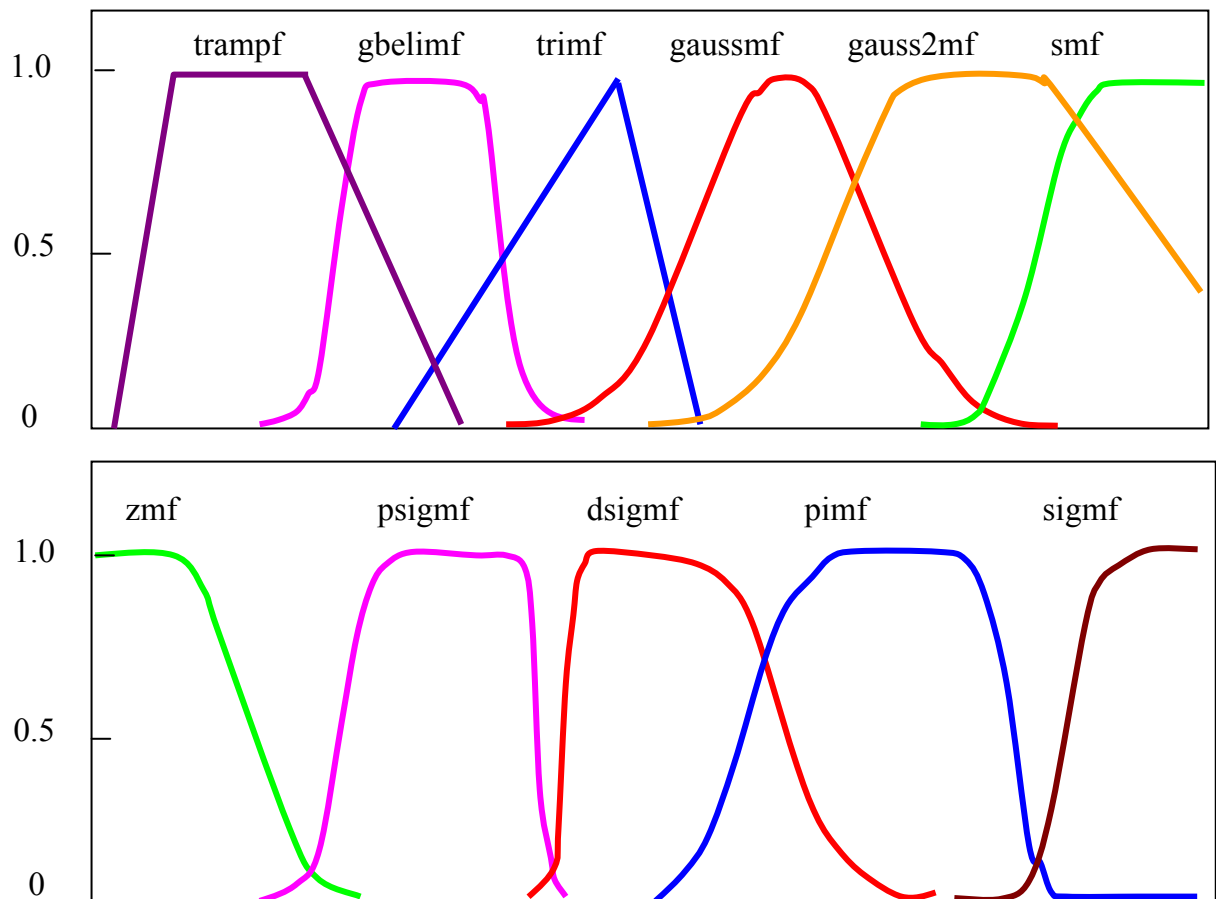


Рис. 8.7 – Види функцій приналежності – *Membership Function*

Після вибору типу функції приналежності можна змінити графік цієї функції, стиснувши або розтягнувши її. Для цього потрібно клацнути по її графіку, узяти мишею виділені прямокутники на графіці і переміщати їх в потрібному напрямі. Можливе завдання чисельних значень параметрів, що описують цю функцію приналежності, в рядку *Param*. Кнопка допомоги **Help** викликає вікно допомоги MATLAB. У ньому даний опис редактора *Membership Functions Editor* і роботи з ним.

Після визначення усіх змінних і завдання типу і виду функцій приналежності переходять до створення правил в редакторі *Rule Editor* (рис. 8.8). Для цього в меню **View** треба вибрати пункт **Edit Rules**.

Редактор *Rule Editor* використовує графічний інтерфейс, заснований на описах змінних входу і виходу. *Rule Editor* дозволяє будувати правила без особливих зусиль, просто і швидко, вибираючи по одному параметру змінної входу і виходу і клацаючи мишкою на них. Правило формується вибором опцій в лівій нижній частині вікна редактора, де приведена уся наявні вхідні та вихідні змінні з вказівкою усіх їх можливих значень, за типом: IF [змінна іє значення] AND/OR [змінна іє значення], THEN [змінна іє значення]. Для формування заперечення значень використовується опція NOT. Для додавання нового правила управління використовується кнопка **Add rule** в нижній частині панелі вікна, для видалення правила - кнопка **Delete rule**.

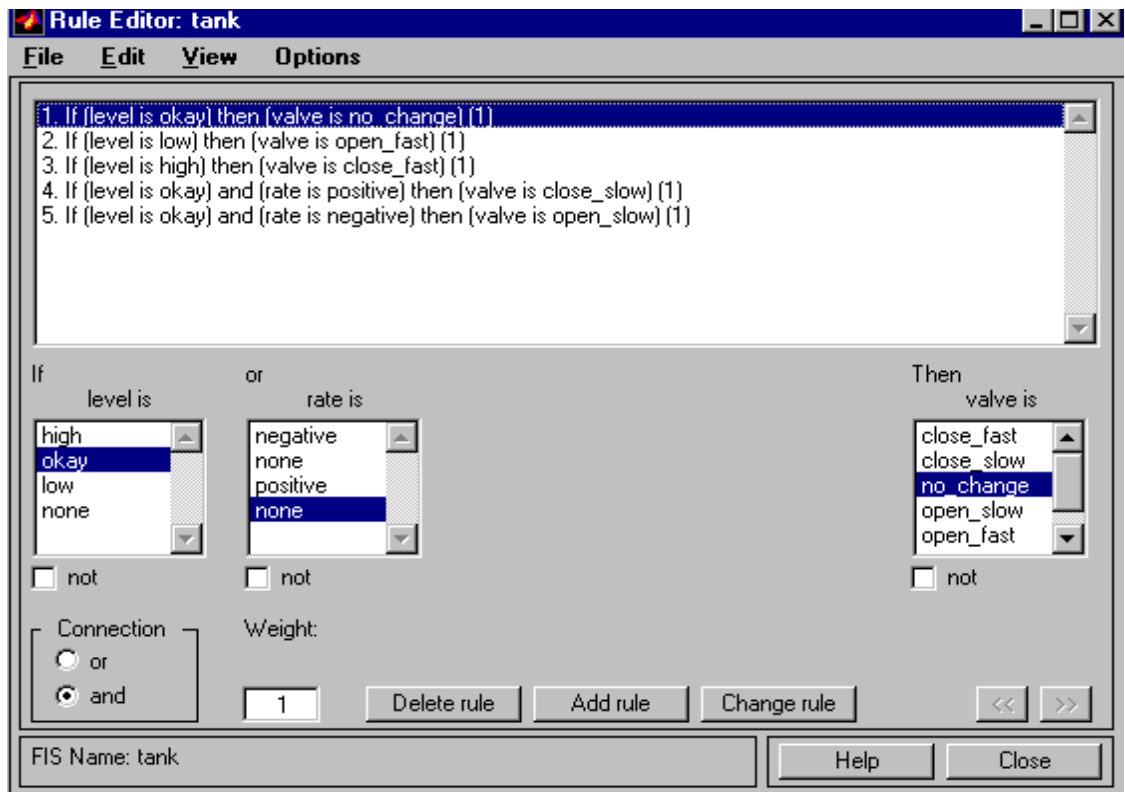


Рис. 8.8 – Редактор *Rule Editor*

При формуванні правила в *Rule Editor* можна не враховувати яку-небудь змінну. Створені правила можна змінювати, не видаляючи їх. Для цього необхідно вибрати правило, яке треба змінити, і виділити мишкою ті параметри змінних, які потрібно встановити. Після усіх проведених операцій натиснути **Change rule**. Зайшовши в пункт **Language** меню **Options**, можна міняти мови (англійську, німецьку і французьку). Окрім цього, в *Rule Editor* передбачена можливість зміни форми запису правил в пункті **Format** меню **Options**.

По закінченню створення правил фазі-управління можна проглянути залежність вихідної величини від значень вхідних параметрів системи. Для цього в меню **View** потрібно вибрати пункт **View rules**. У переглядачі *Rule Viewer* (рис. 8.9) під кожною вхідною і вихідною змінною показані у вигляді графіків її параметри і результат обчислень. Зручність *Rule Viewer* полягає в тому, що як при грубій, так і при точній зміні параметрів входів можна відразу побачити зміну вихідної величини.

Для зміни перших треба вказати мишкою на потрібний графік і клацнути у будь-якому місці. Червона лінія вказівки значення вхідних параметрів переміститься услід за мишкою. Можна задавати значення вхідних параметрів точніше, вказуючи їх в рядку *Input* в лівій нижній частині панелі.

Переглядач *Rule Viewer* дозволяє спостерігати дуже велику кількість змінних, а не тільки три (як в нашому випадку). Для цього призначені чотири клавіші, при натисненні яких екран діаграми переміститься в потрібну сторону відповідно. Наприклад, при натисненні **Left** екран пересуватиметься в лівий бік, а при натисненні **Down** – вниз.

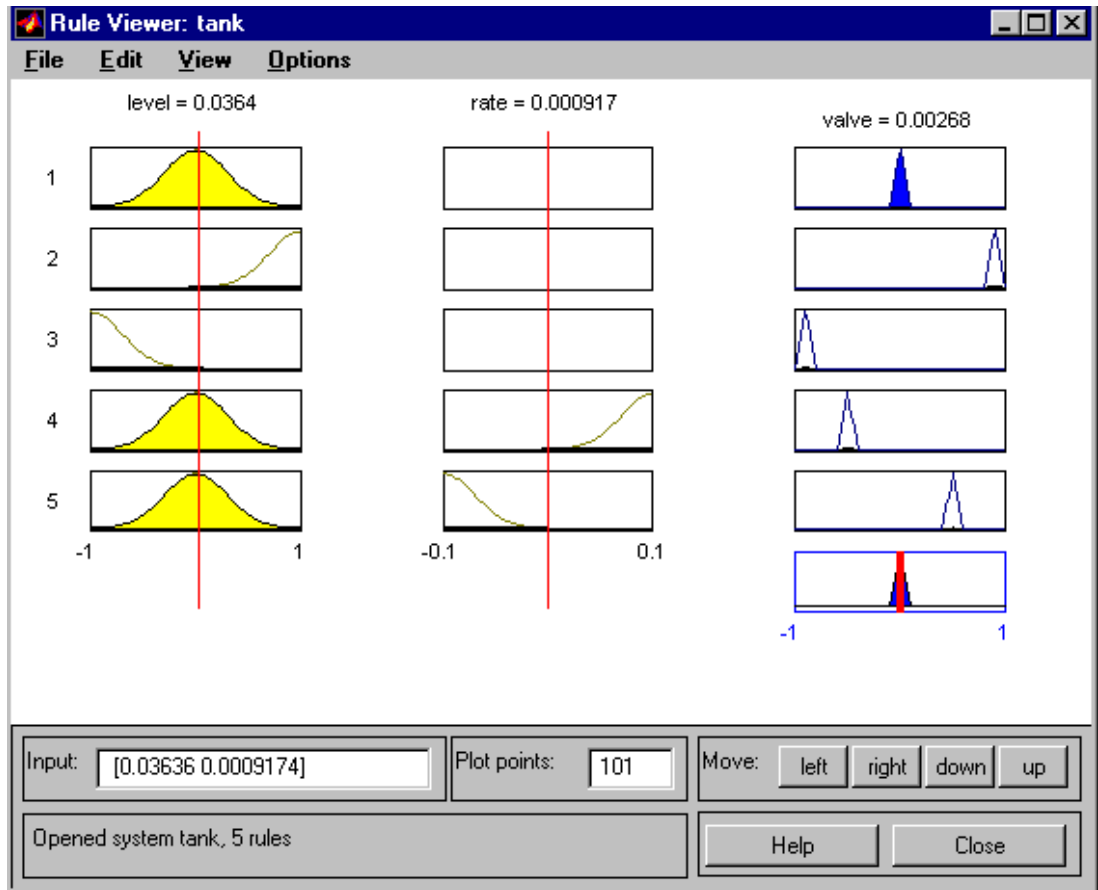


Рис. 8.9 – Переглядач *Rule Viewer*

Вибравши пункт **View Surface** в меню **View**, можна подивитися тривимірну залежність виходу від входу – простір управління фазі-систем. Переглядач *Surface Viewer*, як і *Rule Viewer*, призначений для виведення множини залежностей виходу від входу. На цій діаграмі можна побачити залежність виходу від чотирьох входів або залежність різного виходу від різних входів.

### Порядок виконання лабораторної роботи

1. Відкрийте FIS-редактор, набравши команду `fuzzy tank` у робочому вікні MATLAB. Ознайомтесь з параметрами вхідних і вихідних змінних, набором правил.

2. Відкрийте вікно переглядача *Rule Viewer*, вибравши в меню **View** пункт **View rules**. Змінюючи параметри входів, проаналізуйте зміни вихідної величини.

3. Відкрийте робоче вікно програми MATLAB і наберіть команду `sltank`.

4. У вікні моделі Simulink фазі-системи регулювання рівня рідини, що відкрилося, встановіть у блоці **const** перемикача **Switch** значення 1. Цим самим ви включите в контур регулювання блок PID Controller (насправді ПД-регулятор, оскільки коефіцієнт інтегральної складової дорівнює нулю).

5. Командою **Start** меню **Simulation** запустите симуляцію роботи моделі. Подивіться на вигляд перехідного процесу в системі на екрані блоку **Comparison** (Порівняння).

6. Після зупинки симуляції встановите у блоці **const** перемикача **Switch** значення  $-1$ . Цим самим включите в контур управління фазі-регулятор (блок **Fuzzy Logic Controller**).

7. Командою **Start** меню **Simulation** запустить симуляцію роботи моделі. Подивіться на вигляд перехідного процесу в системі на екрані блоку **Comparison** (Порівняння).

8. Після зупинки симуляції вивчите структуру субмоделей системи управління.

9. Звіт повинен містити мету роботи, схеми S-моделей, функції приналежності вхідних і вихідних величин, осцилограми перехідних характеристик, а також висновки про проведені дослідження.

### Контрольні питання

1. Які основні операції і процедури здійснюються і системі фазі-регулювання?

2. Яка функціональна структура системи фазі-регулювання?

3. Яким операціям алгебри логіки відповідають процедури фазі-об'єднання і фазі-перетину нечітких множин?

4. За допомогою яких процедур визначають ФП для складних посилок, які складаються з декількох простих, сполучених союзами "АБО", "І"?

5. За допомогою якої процедури знаходять результуючу ФП, відповідну сукупності  $j$ -х правил, сполучених союзом "АБО"?

6. Як по графіку  $\mu_p(y)$  можна знайти значення керуючої дії?

## Література

1. Дьяконов, В. SIMULINK 4 : Спец. справ.– СПб. : Питер, 2002. – 528 с.
2. Основы теории автоматического регулирования / Под ред. С. Крутова. – М. : Высш. шк., 1984. – 303 с.
3. Теория автоматического управления : В 2 ч. / Под ред. А. А. Воронова. 2-е изд. – М. : Высш. шк., 1986. – 364с.
4. Елизаров, И.А. Моделирование систем : учебное пособие / И.А. Елизаров, Ю.Ф. Мартемьянов, А.Г. Схиртладзе, А.А. Третьяков. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2011. – 96 с.
5. Поляков, К.Ю. Лабораторные работы «Исследование систем автоматического управления в Matlab» [Электронный ресурс] / К.Ю. Поляков. – Режим доступа : <http://kpolyakov.spb.ru/uni/labs.htm>. – Заголовок з екрану.
6. Борисевич, А.В. Теория автоматического управления : элементарное введение с применением MATLAB / А.В. Борисевич. – СПб : Изд-во СПбГПУ, 2011. – 199 с.
7. Лабораторная работа № 4 [Электронный ресурс]. – Красноярск : ИЦММ СФУ, 2009. – Режим доступа : [http://www.icm.institute.sfu-kras.ru/sites/icm.institute.sfu-kras.ru/files/lab\\_rab\\_4\\_Fuzzy\\_Logic\\_TV.doc](http://www.icm.institute.sfu-kras.ru/sites/icm.institute.sfu-kras.ru/files/lab_rab_4_Fuzzy_Logic_TV.doc). – Заголовок з екрану.

## ДОДАТОК А

### МЕТОДИ ВИЗНАЧЕННЯ ПАРАМЕТРІВ НАЛАШТУВАННЯ РІД КОНТРОЛЕРІВ

Метод *Ziegler-Nichols* полягає в наступному:

- встановлюють коефіцієнти інтегральної  $K_i$  і диференціальної  $K_d$  складових РІД-закону регулювання рівними нулю і збільшують коефіцієнт  $K_p$  пропорційної складової до тих пір, поки система стає нестійкою;

- позначають це значення  $K_u$  і вимірюють період коливань  $P_u$ ;

- приймають наступні значення:

$$K_p = 3 \cdot K_u / 5, K_i = 6 \cdot K_u / (5 \cdot P_u), \text{ и } K_d = 3 \cdot K_u \cdot P_u / 40.$$

Для об'єктів першого порядку можна користуватись формулами таблиці А.1.

Таблиця А.1—Орієнтовочні значення параметрів налаштування регулятора для об'єктів першого порядку із запізнюванням

Тип регулятора	Параметри налаштування	Тип перехідного процесу		
		аперіодичний	з 20%-м перерегулюванням	з $\min \int y^2 dt$
I	$K_p$	$1/(4,5 A) *$	$1/(1,7 A)$	$1/(1,7 A)$
II	$K_p$	$0,3 B *$	$0,7 B$	$0,9 B$
	$K_p$	$0,6 B$	$0,7 B$	$1,0 B$
III	$T_n$	$0,6 T_{об}$	$0,7 T_{об}$	$1,0 T_{об}$
	$K_p$	$0,95 B$	$1,2 B$	$1,4 B$
ПІД-	$T_n$	$2,4 \tau_{об}$	$2,0 \tau_{об}$	$1,3 \tau_{об}$
	$T_d$	$0,4 \tau_{об}$	$0,4 \tau_{об}$	$0,5 \tau_{об}$

\* Примітка:  $A = K_{об} T_{об}$ ;  $B = T_{об} / (K_{об} \tau_{об})$ .