

С.А. Войцех, н.с.

С.М. Москвіна, к.т.н., доц.

Ю.В. Поремський, студ.

Вінницький державний технічний університет

## ПІДХІД ДО СТВОРЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МОДЕЛЮВАННЯ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ СИСТЕМ

*В даній статті виконується аналіз, що доводить неможливість використання існуючих інформаційних технологій (ІТ) для дослідження стійкості нелінійних нестационарних систем. Аналізується простір приросту параметрів, який і пропонується використовувати в розробці інформаційної технології для дослідження нелінійних нестационарних систем. Пропонується розроблений графічний інтерфейс пакета програм для моделювання нелінійних нестационарних систем (ННС), що є найбільш зручним і оптимальним для виконання досліджень різних типів систем автоматичного керування (САК) на ЕОМ. Наводиться алгоритм переходу від графічної до математичної моделі САК, який надасть можливість впровадження простору приросту параметрів в інформаційну технологію, що розробляється на даний час.*

### Вступ

Дослідження процесів в нелінійних системах із змінними первинними параметрами проводиться, в основному, за допомогою безпосереднього моделювання на комп'ютері при заданому наборі первинних параметрів, повторюючи його для кожного нового набору. Звичайно, це приводить до збільшення часу розв'язання задач аналізу поведінки і не завжди вдається простежити вплив параметрів, що змінюються, завдяки тому, що неможливо промоделювати поведінку систем при всіх елементах числових множин, які задають ці параметри. Тому виникає необхідність розробки нового простору, опису поведінки нелінійних нестационарних систем, в якому можливо було б оцінити вплив зміни первинних параметрів широкого класу систем на властивість їх стійкості.

### Аналіз існуючих інформаційних технологій для дослідження САК

Аналіз засобів моделювання дозволяє виділити два напрямки в області розробки інформаційних технологій моделювання систем автоматичного керування. Мови моделювання, до яких можна віднести GPSS – вміщає спеціальні засоби для опису динамічної поведінки систем, стан яких змінюється в дискретні моменти часу (наприклад, систем масового обслуговування), GASP – використовується для моделювання не тільки дискретних, але й змішаних неперервно – дискретних систем, SIMULA – використовує блочне представлення моделі системи, в якій вводиться поняття процесу при формалізації елементів блоків моделі, а процес задається набором ознак, які характеризують його структуру програмою, SLAM отримав широке розповсюдження за універсальність, використовувалася для моделювання як неперервних, так і дискретних систем. Та інформаційні технології для моделювання: ДИСПАС, MAC, Modos, MatLab, MathCAD та DCNET. Розглянемо та проаналізуємо кожний з цих засобів моделювання більш детально.

ДИСПАС – дозволяє отримувати [1] перехідний процес лінійних та нелінійних, стаціонарних та нестационарних, безперервних й дискретних логіко – динамічних систем, поведінка яких описується звичайними диференційними і диференційно – різностними рівняннями, також здійснюється оптимізація параметрів моделі, що аналізуються. Цей засіб моделювання працює в діалоговому режимі. Найбільш часто застосовують на етапі технічного проектування. Перевагою є орієнтація на користувача, який не володіє програмуванням.

Система машинного аналізу і синтезу систем (МАСС) досліджує широкий клас САК (лінійних, нелінійних, імпульсних, цифрових та таких, що виконують логічні функції) у часовій області. Системи, які досліджується, можуть бути як стаціонарними, так і нестационарними. Основним способом дослідження є побудова перехідного процесу при заданих

початкових умовах та вхідних діях (як детермінованих, так і стохастичних). Програмне забезпечення МАСС надає ітеративний режим взаємодії з ЕОМ та має оверлейну структуру. Користувачу потрібно лише відповідати на питання, які задаються ЕОМ, сама ж послідовність питань визначається пакетом програм. Лінгвістичне забезпечення системи МАСС відноситься до мов табличного типу і дозволяє описувати системи, математичні моделі яких задаються структурними схемами. При цьому система розбивається на типові функціональні блоки (лінійні динамічні ланки, нелінійні, логічні ланки, дискретні елементи та ін.), які утворюють бібліотеку вхідної мови. Відомо, що в склад ІТ [2] входять оптимізаційні процедури, які дозволяють вирішувати задачі параметричної оптимізації, за допомогою алгоритмів Розенброка, симплекс – метода, випадкового пошуку, Фібаначі та Гауса – Зейделя. Критерій оптимізації формується користувачем, за допомогою блоків вхідної мови і задається номером блока, вихід якого в кінці розрахунку повинен бути обов'язково мінімальним (вирішення задачі максимізації у прямій постановці неможливо). В процесі моделювання САК досліджується вплив зміни параметрів системи на перехідний процес, по виду якого процесу визначається стійкість системи, яка досліджується. Однак у випадку, коли параметри системи належать деякій області або змінюються в часі по будь-якому закону, відповідь на питання про стійкість таких систем можливо отримати лише шляхом аналізу перехідних процесів систем при різних фіксованих наборах параметрів. Для проведення такого аналізу необхідно велике число розрахункових експериментів і затрат машинного часу але, при цьому достатньо складно визначити, при яких умовах система стане нестійкою.

MatLab вміщає в собі IT Robust Control [3], що має засоби для проектування та аналізу багатопараметричних систем керування, стійкість яких має велике значення. Серед таких систем можуть бути системи з модельними похибками, динаміка яких відома не повністю, чи параметри яких можуть змінюватися в ході моделювання. Алгоритми пакета програм дозволяють виконувати складні обчислення, приймаючи до уваги зміну параметрів. До найбільш важливих властивостей можна віднести: синтез LQG регуляторів на основі мінімізації рівномірних та інтегральних норм, побудова моделі простору станів, перетворення моделей на основі сингулярних чисел, зниження порядку моделі і спектральна факторизація. Пакет Robust Control базується на функціях пакета Control System, має набір алгоритмів для проектування систем керування, забезпечує зв'язок між сучасною теорією керування і практичними роботами та включає велику кількість функцій, що реалізують сучасні методи проектування й аналізу багатопараметричних робастних регуляторів. Крім того IT Robust Control дозволяє оцінити багатопараметричну границю стійкості при різних невизначеностях.

Інформаційна технологія DCNET[4] була розроблена на Україні. Різницею між цією та розглянутими раніше ІТ є те, що DCNET орієнтована на дослідження систем за допомогою математичного апарату дискретно-неперервних мереж. Такі особливості даної розробки, як графічне введення дискретно-неперервної (ДН) моделі об'єкта у вигляді ДН – мереж, можливість вияву зміни в структурі об'єкта в процесі моделювання, наявність режиму покрокового моделювання з можливістю модифікації структури та стану неперервних частин об'єкта, а також зручність отриманих результатів дозволяють значно виділити його від інших ІТ.

На жаль використовувати більшість з вище наведених ІТ, всіх крім MatLab, MathCAD, DCNET та MODOS, в даний час практично немає можливості в зв'язку з їх орієнтацією на застарілі ЕОМ та операційні системи. Та всі ці засоби моделювання не призначені для моделювання нелінійних нестационарних систем і дослідження їх поведінки в просторі параметрів.

### **Вибір нового простору опису поведінки нелінійних нестационарних систем**

Існуючі методи оцінки стійкості САК, що використовуються в розглянутих інформаційних технологіях, розв'язують задачу дослідження нелінійних нестационарних систем для заданого набору значень загальної матриці системи, структура якої визначає характер перехідної матриці станів. При цьому під час проектування широкого класу систем необхідно часто розв'язувати обернену задачу, тобто визначити множину значень первинних параметрів системи, з якими

властивості стійкості зберігаються. Потрібно відзначити, що більш за все залежність між первинними параметрами систем, що аналізуються та значеннями елементів загальної матриці системи носить неявний нелінійний характер, що не дозволяє в рамках існуючих підходів оцінити вплив первинних параметрів, а не значень елементів матриці системи, на властивість стійкості.

Таким чином, виникає необхідність розробки нового простору, опису поведінки нелінійних нестационарних систем, в якому дозволило б оцінити вплив зміни первинних параметрів широкого класу систем на властивість їх, стійкості. При цьому, звичайно, необхідно відмовитись від класичного опису динамічних систем:

$$\mu = \langle Y(x(t), P(t), t), z(t), t, t_0) \rangle \tag{1}$$

де  $Y(x(t), P(t), t)$  – кінцевомірний вектор змінних стану системи розмірністю  $(1 \times m)$ ;  
 $x(t)$  – кінцевомірний вектор значення керуючих (вхідних) впливів розмірністю  $(1 \times n)$ ;  
 $P(t)$  – кінцевомірний вектор значень первинних параметрів системи розмірністю  $(1 \times p)$ ;  
 $z(t)$  кінцевомірний вектор спостережуваних координат розмірністю  $(1 \times k)$ ;  
 $t$  – поточний час;  $t_0$  – час початку опису системи.

З [4] випливає, що у випадку, коли вектор  $X(t)$ ,  $P(t)$  та  $Z(t)$  представляється кусочно – неперервними функціями, заданими на інтервалі часу  $[t_0, T]$ , то у просторі приростів параметрів опис поведінки системи має вигляд:

$$m^* = \langle Y_k(\bullet) \langle X(\bullet), P(\bullet), Z(\bullet), P_0 \rangle \in L_\infty([t_0, T]; R^{m \times p}) \times L_\infty([t_0, T]; R^n) \times L_\infty([t_0, T]; R^p) \times L_\infty([t_0, T]; R^k) \times R^k \tag{2}$$

де  $P_0$  – кінцевомірний вектор початкових значень первинних параметрів системи розмірністю  $(1 \times p)$ .

Якщо ж функція  $Y(X(t), P(t), t)$  неперервна, а також неперервні її частинні похідні  $Y_x(X(t), P(t), t)$  і  $Y_p(X(t), P(t), t)$ , то в цьому випадку опис поведінки системи має вигляд

$$m^{**} = \left\langle \frac{\partial Y(X(\bullet), P(\bullet))}{\partial P(\bullet)} \cdot X(\bullet), P(\bullet), Z(\bullet), P_0 \right\rangle \in C([t_0, T]; R^{m \times p}) \times C([t_0, T]; R^n) \times C([t_0, T]; R^p) \times C([t_0, T]; R^k) \times R^k \tag{3}$$

Опис нелінійних нестационарних систем у вигляді (2) та (3) дозволяє проводити аналіз їх, поведінки не у просторі змінних станів систем, як у класичному випадку опису у вигляді (1), а у іншому просторі, який називається простором приросту параметрів. Така назва зумовлена тим фактом, що поведінка систем при описі їх у вигляді (2) та (3) розглядається відносно приростів змінних станів, які зумовлюються приростом первинних параметрів систем.

В роботі [4] підкреслено, що опис системи у вигляді (2) та (3) проводиться за допомогою апарату теорії чутливості. Використання такого апарату дозволяє перейти від нелінійних нестационарних диференціальних рівнянь, що описують поведінку систем у просторі змінних станів, до лінійних нестационарних диференціальних рівнянь відносно зміни приросту змінних станів, зумовлених зміною первинних параметрів системи, що полегшують моделювання поведінки, розглянутих класів систем.

Відомо [5], що отримання досить повного подання про характер поведінки нелінійних систем без безпосереднього інтегрування диференціальних рівнянь можливо одержати за допомогою фазових траєкторій, побудова яких за допомогою ряду спеціальних пакетів прикладних програм не представляє особливих труднощів. Опис у вигляді (3) для випадку систем другого порядку дозволяє побудувати траєкторії, що є аналогами фазових траєкторій та оцінити вплив зміною первинних параметрів нелінійних систем на зміну цих первинних параметрів. Тобто, якщо для безперервних та диференційованих по первинних параметрах змінних станів для систем другого порядку, у випадку опису їх, поведінки у вигляді (1), фазові траєкторії представляють собою залежності, тоді для систем другого порядку у випадку їх опису у вигляді (3) аналогі фазових траєкторій представляють собою залежності:

$$\frac{dy}{dt} = f_j(y), (j = \overline{1, P}) \tag{4}$$

Отримані аналогі дозволяють, використовуючи розроблені методи аналізу нелінійних систем у просторі станів, досліджувати особливі точки на площині, аналогічної фазової

площини. Підкреслимо, що поява особливих точок у випадку, що розглядається, зумовлюються зміною первинних параметрів систем, що аналізуються  $P_j (j = \overline{1, P})$ . Крім того, використовуючи аналог, що розглядається, можливо досліджувати поведінку систем близько положенням рівноваги, використовуючи розроблені методи аналізу нелінійних систем.

### Структура графічного інтерфейсу пакета програм для моделювання ННС

Використання простору приросту параметрів [4] та однієї з мов візуального програмування дозволяє розробити пакет програм (ПП) для дослідження нелінійних нестационарних систем. Саме моделювання за допомогою такого ПП дає можливість більш наглядно будувати САК, які необхідно дослідити, від структурної схеми до математичної моделі проводити експериментальне дослідження при будь-яких вхідних даних та параметрів системи і отримувати результати в найбільш зручному для користувача вигляді. Тобто ПП може використовувати, як обізнана так і не володіюча достатньо комп'ютером людина.

В даній роботі розглядається ПП для моделювання нелінійних нестационарних систем, в структурі якого передбачений, як класичний підхід к аналізу і дослідженню САК, математичне моделювання якого задається структурними схемами, так і аналіз систем в просторі приросту параметрів [8]. При дослідженні систем ПП дозволяє розв'язувати наступні задачі: побудова і перетворення структурної схеми системи (графічний інтерфейс ПП); перехід від графічної до математичної моделі досліджуваної системи; математичне моделювання при різних вхідних даних та параметрах систем та аналіз результатів моделювання.

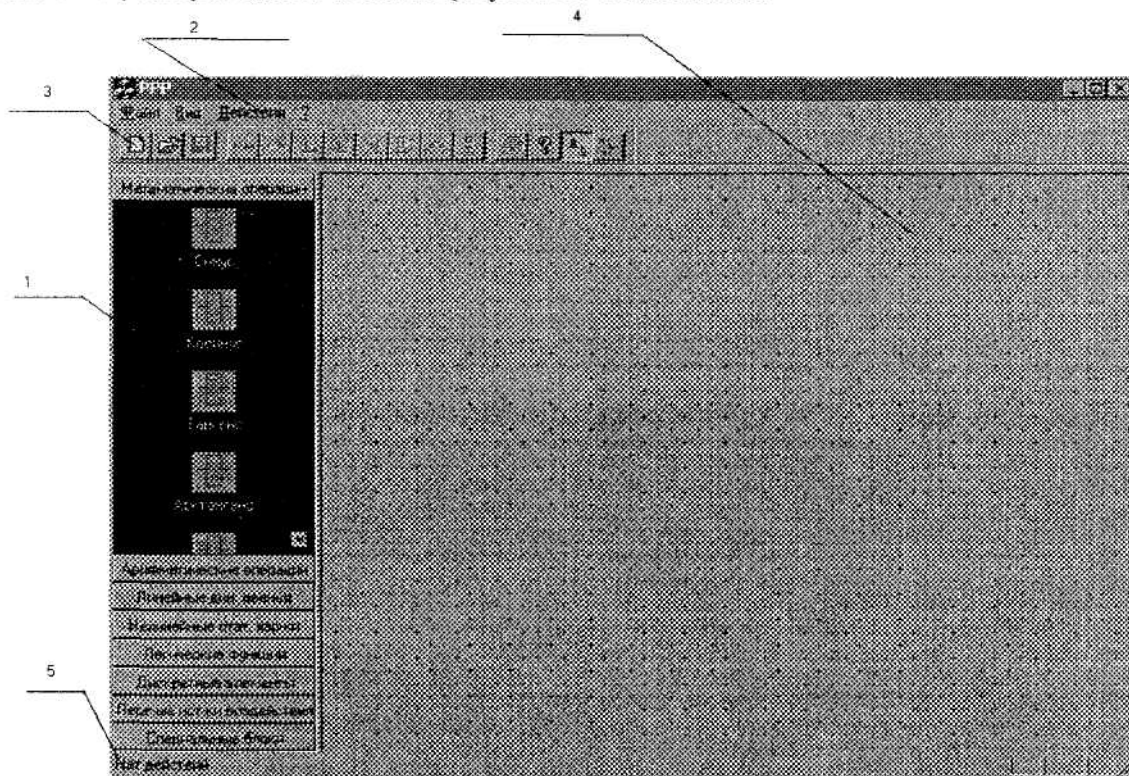


Рис. 1. Головне вікно пакета програм для моделювання ННС

На рис. 1 наводиться головне вікно графічного інтерфейсу, робота якого дозволяє побудувати структурну схему досліджуваної системи з використанням бібліотеки блоків, функціональної бібліотеки перетворень графічних об'єктів, базових команд меню та сервісних функцій, що відрізняє цей ПП від інших простотою в користуванні та можливістю розв'язувати велику кількість процедур. Графічний інтерфейс включає в себе: 1 – бібліотека блоків, 2 – панель меню, 3 – панель інструментів, 4 – листок для моделювання, 5 – панель підказки.

Бібліотека блоків представляє собою сукупність всіх блоків, що складені по розділах: арифметичні операції, математичні функції, лінійні динамічні ланки, нелінійні статичні характеристики, логічні функції, дискретні елементи, джерела.

Панель меню вміщає основні необхідні для моделювання команди: створити новий лист, зберегти лист, відкрити вже існуючий лист, видалити блок з листа, перемістити блок в інше місце на листі, автоматично з'єднати два елементи, покроково з'єднати два елементи, обернути елемент по часовій та проти часової стрілки та інші. Панель інструментів складається з кнопок, що вміщують в собі основні дії панелі меню. Листок представляє собою площину, на якій виконується безпосереднє створення графічної моделі САК. На рис. 2 зображений приклад створеної структурної схеми.

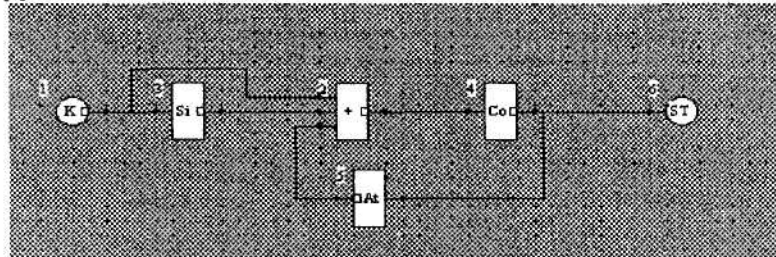


Рис. 2. Приклад створеної структурної схеми

З рис. 2 видно, що кожний елемент має порядковий номер. Номер відповідає тому, який з елементів був занесений на листок моделювання першим, який другим і т.д. Панель підказки – виводить символічне значення дії, що виконує кожна з кнопок, на якій зупиняється курсор миші.

#### Алгоритм переходу від графічної до математичної моделі

При створенні ІТ достатньо складним виявилась розробка алгоритму автоматичного переходу від структурної схеми, яка складається на основі блок-схеми системи та рівнянь, які описують її ланки, або тільки на основі рівнянь [6]. Ланка на структурній схемі не обов'язково зображає модель якого-небудь конкретного елемента. Вона може бути моделлю елемента, з'єднання елементів або взагалі будь-якої частини системи.

Звичайно при моделюванні використовуються три основні типи з'єднання ланок: послідовне, паралельне, із зворотнім зв'язком, для кожного з яких відомі математичні вирази перетворень [6].

Для визначення математичної моделі досліджуваної системи, перетворюють структурну схему на основі правил еквівалентних перетворень, в основу яких покладена вимога збереження незмінними вхідних та вихідних величин ділянки схеми, яка перетворюється. При виконанні цієї вимоги забезпечується еквівалентність перетвореної та початкової структурних схем в тому розумінні, що обидві вони відповідають одному і тому ж пристрою або диференційним рівнянням.

Математичну модель системи керування наочно також представити за допомогою орієнтованих графів [6]. Для спрощення графу та обчислення передаточних функцій системи керування за її графом звичайно використовують достатньо відомі правила та формулу Мейсона [6]. На жаль, жоден з цих способів побудови математичної моделі системи для алгоритмів автоматичного переходу від графічної до математичної моделі, використовувати виявилось неможливим. Крім того, застосування формули Мейсона, як відомо [6], ускладнюється і в деяких випадках, при дослідженні нелінійних нестационарних систем, є неможливим. Тому виникає необхідність у створенні алгоритму переходу від графічної до математичної моделі, який і буде застосовуватись в інформаційній технології, що розробляється.

В ІТ пропонується підхід до моделювання САК, який використовує послідовний аналіз передаточних функцій кожного з структурних елементів системи, визначається реакція кожного елемента на вхідний вплив. Тому структурна схема САК розбивається на частини, що досліджується на кожному кроці моделювання окремо. Так, з рис. 3 видно, що такими

частинами будуть елементи 1, 2, 3 та 4 відповідно. Тобто спочатку досліджується елемент 1, далі 2, 3 та 4.

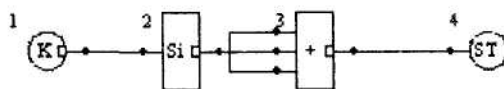


Рис. 3. Приклад структурної схеми САК

Виконання моделювання, з використанням такого підходу потребує визначити граф з'єднання елементів між собою, знайти передаточні функції для кожного елемента окремо за допомогою бібліотеки передаточних функцій, та на кожному кроці, за допомогою вже знайденого графу, аналізувати відповідні передаточні функції.

Розглянемо суть запропонованого алгоритму. На рис. 3 наведена структурна схема САК, на якій зображені чотири елементи: константа, функція синуса, суматор та логічний елемент кінця схеми. Граф з'єднання: перший елемент (K) з'єднується з другим елементом (Si), другий елемент з третім (+) та третій з четвертим (ST). Зобразимо граф з'єднання за допомогою матриці:

$$\begin{matrix}
 0 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0
 \end{matrix}$$

Рис. 4. Граф з'єднання елементів

Структурна схеми САК, що зображена на рис. 3, не потребує складного алгоритму знаходження послідовності елементів, тобто елемента, який досліджується перший, який другий і т.д. Але вже для дослідження САК, що наведена на рис. 5, необхідно використовувати алгоритм, що буде знаходити послідовність розглядання елементів.

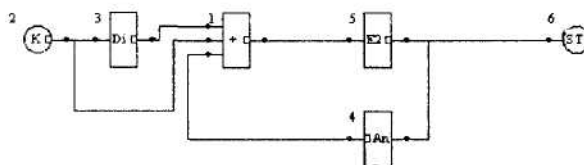


Рис. 5. Приклад більш складної структурної схеми САК

На рис. 6 зображено результуючий граф з'єднання елементів для САК, яка наведена на рис. 5.

$$\begin{matrix}
 0 & 0 & 0 & 0 & 1 & 0 \\
 1 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0
 \end{matrix}$$

Рис. 6. Граф з'єднання елементів САК

За допомогою бібліотеки елементів співставляємо кожний елемент з відповідною функцією або диференційним рівнянням. Наступний крок моделювання САК, пошук послідовності елементів, є більш складним у зв'язку з присутністю в САК оберненого зв'язку.

Саме для випадків з оберненим зв'язком, що є звичайними в дослідженні САК, пропонується алгоритм пошуку послідовності виконання елементів.

**Алгоритм**

Крок 1. Скласти структурну схему системи, що досліджується.

Крок 2. Отримати граф з'єднань елементів структурної схеми між собою у вигляді матриці.

Крок 3. Визначити початкові елементи САК.

Крок 4. За допомогою графу з'єднань, проводимо пошук шляху сигналу по схемі, починаючи з відомих елементів, що вже були знайдені.

Крок 5. Якщо знайшли елемент, переходимо до Кроку 6, інакше, при неможливості знайти інші елементи структурної схеми, переходимо до Кроку 7.

Крок 6. Якщо всі елементи знайдені, то переходимо до Кроку 12, інакше до Кроку 4.

Крок 7. Знаходимо кінцевий елемент структурної схеми та запам'ятовуємо його.

Крок 8. Починаючи зі збережених елементів, знаходимо елементи, що подають їм вхідні значення. Та запам'ятовуємо їх замість перших елементів. Якщо знайдені всі елементи, то переходимо до Кроку 12.

Крок 9. Якщо є можливість відкрити елемент по відомому вхідному впливу переходимо до Кроку 10, інакше до Кроку 8.

Крок 10. Відкриваємо всі можливі елементи за вхідними значеннями та переходимо до Кроку 11.

Крок 11. Якщо знайдені всі елементи, переходимо до кроку 12, інакше до Кроку 8.

Крок 12. Зберігаємо всі елементи та закінчуємо виконання алгоритму.

Розглянемо результати роботи ІТ моделювання нелінійних нестационарних систем, який вміщує в собі вищенаведений алгоритм, для структурних схем САК, що наведені на рис. 3 та 5.

#### Приклад 1.

Крок 1. Складаємо структурну схеми системи автоматичного керування (рис. 3).

Крок 2. Отримати граф з'єднань елементів структурної схеми між собою у вигляді матриці.

Крок 3. Визначаємо початкові елементи. В даному прикладі початковий елемент 1.

Крок 4. За допомогою графу з'єднань, проводимо пошук шляху сигналу по схемі, починаючи з відомих елементів, що вже були знайдені. Знайденим елементом є 1, який подає вхідні значення до елемента 2, а елементу 2 вхідні значення подає лише один елемент.

Крок 5. Відкриваємо елемент 2 і записуємо його до шуканої послідовності елементів. Переходимо до Кроку 6.

Крок 6. Так, як знайдені ще не всі елементи виконуємо Крок 4.

Крок 4. Елемент 2 подає вхідні значення до 3 елементу і 3 елемент приймає вхідні значення лише від одного елемента.

Крок 5. Тому заносимо елемент 3 до послідовності елементів. Переходимо до Кроку 6.

Крок 6. Не всі елементи знайдено, переходимо до Кроку 4.

Крок 4. Елемент 3 подає вхідні значення до 4 елемента і елемент 4 приймає вхідні значення лише від одного елемента.

Крок 5. Заносимо елемент 4 до послідовності елементів. Переходимо до Кроку 6.

Крок 6. Знайдені всі елементи, тому переходимо до Кроку 12.

Крок 12. Запам'ятовуємо послідовність елементів: 1, 2, 3 та 4. Закінчуємо виконання алгоритму.

Результати моделювання структурної схеми САК з рис. 3 наведені на рис. 7.



Рис. 7. Результат роботи алгоритму в ПП моделювання САК

#### Приклад 2.

Розглянемо більш складний приклад структурної схеми з оберненими зв'язками, що наведений на рис. 5.

Крок 1. Складаємо структурну схему САК, що досліджується (рис. 5).

Крок 2. Отримати граф з'єднань елементів структурної схеми між собою у вигляді матриці.

Крок 3. Визначаємо початкові елементи. В даному прикладі початковий елемент 2 (константа).

Крок 4. За допомогою графу з'єднань, проводимо пошук шляху сигналу по схемі, починаючи з відомих елементів, що вже були знайдені. Знайденим елементом є 2, який подає

вхідні значення до елемента 3 (диференціювання), а елемента 2 вхідні значення подає лише один елемент.

Крок 5. Відкриваємо елемент 3 і записуємо його до шуканої послідовності елементів. Переходимо до кроку 6.

Крок 6. Не всі елементи ще знайдені, тому переходимо до Кроку 4.

Крок 4. За допомогою графу з'єднань, проводимо пошук шляху сигналу по схемі, починаючи з відомих елементів, що вже були знайдені. Знайденими елементами є 2 та 3, вони подають вхідні значення елемента 1, але у зв'язку з тим, що невідомий елемент 4, який також подає вхідне значення елемента 1. Переходимо до Кроку 5.

Крок 5. Не можливо більш відкрити елементів, тому переходимо до Кроку 7.

Крок 7. Знаходимо кінцевий елемент структурної схеми. Кінцевим елементом є 6. Запам'ятовуємо його.

Крок 8. Починаючи зі збережених елементів, знаходимо елементи, що подають їм вхідні значення. З структурної схеми видно, що вхідний сигнал елемента 6 подає елемент 5 (ланка другого порядку). Запам'ятовуємо його замість елемента 6.

Крок 9. Так, як є можливість відкрити елемент 4 по відомому вхідному впливу 5, то переходимо до Кроку 10.

Крок 10. Відкриваємо елемент 4. Переходимо до Кроку 8.

Крок 8. Починаючи зі збережених елементів, знаходимо елементи, що подають їм вхідні значення. З структурної схеми видно, що вхідний сигнал елемента 5 подає елемент 1, (ланка другого порядку). Запам'ятовуємо його замість елемента 5. Оскільки знайдені всі елементи, то переходимо до Кроку 12.

Крок 12. Запам'ятовуємо всю послідовність елементів: 2,3,1,5,4 та 6. Закінчуємо виконання алгоритму.

Результати моделювання структурної схеми САК з рис. 5 наведені на рис. 8.

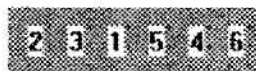


Рис. 8. Результат роботи алгоритму для САК з оберненим зв'язком

Отриманні результати моделювання доводять працездатність алгоритму для будь-якої складності структурних схем.

Наступним кроком при моделюванні САК є співставлення на кожному кроці вихідних результатів одного елемента з передаточними функціями іншого для послідовності елементів, що була знайдена за допомогою вище наведеного алгоритму. З цією метою була створена в інформаційній технології окрема процедура, що вмещає в собі бібліотеку передаточних функцій. Як наслідок виникає необхідність в аналізі передаточних функцій елементів, які використовуються при моделюванні. Якщо передаточна функція, наприклад, є функцією синуса або суми, то запрограмувати її не складно, але в випадку диференціальних рівнянь необхідно застосовувати один з відомих чисельних методів: модернізований метод Ейлера, Рунге-Кутта, прогнозу та корекції або інші. Використання запропонованого алгоритму дозволяє отримати результати моделювання в табличному вигляді і для наступного дослідження дає можливість застосовувати традиційні методи.

### Висновки

В роботі виконано аналіз відомих інформаційних технологій, що використовуються для дослідження САК та доведена неможливість їх використання для дослідження нелінійних нестационарних систем в просторі приросту параметрів

Запропонована структура ІГ для моделювання нелінійних нестационарних систем в просторі станів, робота якого ґрунтується на класичних підходах та використання простору приросту параметрів. Наведено опис ПП програм для моделювання ННС та його графічного інтерфейсу, який на відміну від розглянутих ПП є зручним та достатнім для користувача.

Запропоновано алгоритм переходу від графічної до математичної моделі САК, який дасть можливість проводити дослідження в просторі приросту їх параметрів.



**ЛІТЕРАТУРА:**

1. *Згуровський М.З., Денисенко В.А.* – Дискретно – неперервні системи з керованою структурою. Теорія моделювання застосування. Київ, видавництво “Наукова думка”, 1998. – 350с.
2. *Юхимчук С.В., Лысогор В.Н., Марущак В.Ю.* Автоматизация проектирования систем автоматического управления: Учебн. Пособие. – К.: УМК ВО, 1898. – 172с.
3. *Д.Мезьюз, К.Финк.* – Численные методы. Использование MATLAB. Вильямс. 2001 (перевод с английского Л.Ф.Казаченко, под редакцией д.ф.-м.н., проф. Ю.В. Козаченко).- 346с.
4. *Юхимчук С.В.* Математические модели оценки устойчивости нелинейных нестационарных систем – Винница, издательство ВГТУ “УНІВЕРСУМ – Вінниця”, 1997г.- 141с..
5. *Гелбднер К., Кубик С.* Нелинейные системы управления – М.: Мир, 1987. – 368с.
6. *Самотокин Б.Б.* Лекції з теорії автоматичного керування. Навчальний посібник для студентів вищих навчальних закладів.- Житомир: ЖІТІ, 2001. – 508с.
7. *Юхимчук С.В., Москвіна С.М., Войцех С.О.* Побудова графічного і текстового інтерфейсу пакета програм для моделювання поведінки нелінійних нестационарных систем. ВІСНИК ВПИ. 3(36) 2001. – 165с.

ВОЙЦЕХ Сергій Анатолійович – науковий співробітник, інженер-програміст Вінницького державного технічного університету.

Наукові інтереси:

– розробка методів та алгоритмів дослідження ННС.

МОСКВІНА Світлана Михайлівна – кандидат технічних наук, доцент кафедри комп'ютерних систем управління Вінницького державного технічного університету.

Наукові інтереси:

– розробка методів та алгоритмів дослідження ННС.

Тел. (0432) – 44-87-84; [mms@freemail.ru](mailto:mms@freemail.ru).

ПОРЕМСЬКИЙ Юрій Віталійович – студент кафедри інтелектуальних систем Вінницького державного технічного університету.

Наукові інтереси:

– розробка методів та алгоритмів дослідження ННС.

Подано 17.08.2002