



In this paper we showed just an example of our scientific projects. The Laboratory of the Department of Machinery Design Fundamentals provides a wide range of structural calculations using a lot of mathematical and simulation software. We propose a service in the field of machinery, solution of different problems concerning mechanical, control system and other equipment's issues.

REFERENCES

1. Johnson, K.L. (1985) Contact Mechanics, Cambridge University Press, 1985.
2. Xiaolin, C., Yijun, L. Finite Element Modeling and Simulation with Ansys Workbench, CRC Press, 2014, 411 p.

УДК 681.3.01

МЕТОДИКА ПОПЕРЕДНЬОЇ ОБРОБКИ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

В.С. Ловейкін¹, Ю.О. Ромасевич²

¹доктор технічних наук, завідувач кафедри конструювання машин, Національний університет біоресурсів і природокористування України, м. Київ, Україна

²кандидат технічних наук, докторант кафедри конструювання машин, Національний університет біоресурсів і природокористування України, м. Київ, Україна, e-mail: d.um@mail.ru

Анотація. В роботі наведено алгоритми попередньої обробки експериментальних даних, які представлені у вигляді багатомірного масиву. Розроблені алгоритми дозволяють автоматично визначати моменти початку та закінчення досліджуваного процесу, виконувати калібровку нуля, згладжувати дані та формувати двомірні масиви „час - вимірювана величина”.

Ключові слова: експериментальні дані, масив, обробка, цикл.

METHODS OF EXPERIMENTAL DATA PRETREATMENT

Vyacheslav Loveikin¹, Yuriy Romasevich²

¹Doctor of technical Sciences, Head of Department of Machinery Design, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine

²Ph.D., Doctoral candidate of Department of Machinery Design, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine, e-mail: d.um@mail.ru

Abstract. The algorithms for experimental data preprocessing presented in the form of a multidimensional array are stated in the article. The designed algorithms allowed automatically to determine the start and end points of studied process, to perform calibration of zero, to smooth the data and to form two-dimensional arrays „time - measured value”.

Keywords: experimental data, array, processing, cycle.





Вступ. Відомо, що експериментальні дослідження дають змогу отримати важливу інформацію про досліджуваний процес. Після проведення експериментів наступним етапом аналізу досліджуваного процесу є обробка масиву зібраних даних. Звичайно така обробка виконується за допомогою статистичних методів [1-3]. Однак, статистичні методи можна використовувати лише для масивів даних у яких дискретне значення вимірюваної величини відповідає деякому моменту часу. Отже, перед виконанням статистичного аналізу необхідно провести попередню обробку „сирого” експериментального матеріалу. Звичайно такі процедури проводяться з використанням персонального комп’ютера. Тому методика попередньої обробки експериментальних даних повинна бути реалізована у вигляді сукупності певних алгоритмів, які легко реалізуються у відомих математичних програмних продуктах [4, 5].

Мета роботи. Метою роботи є розробка алгоритмів попередньої обробки експериментальних даних, які представлені у вигляді багатомірного масиву, для подальшого статистичного аналізу.

Матеріал та результати досліджень. Як правило експериментальні дані, які записані на персональному комп’ютері, представляються у такій формі:

$$\{ \{x_{11}, x_{12}, \dots, x_{1i}, x_{1i+1}, \dots, x_{1n}\}, \{x_{21}, x_{22}, \dots, x_{2i}, x_{2i+1}, \dots, x_{2n}\}, \{x_{m1}, x_{m2}, \dots, x_{mi}, x_{mi+1}, \dots, x_{mn}\} \}, \quad (1)$$

де m – кількість параметрів, які вимірювались при проведенні експерименту; n – довжина масиву (кількість вимірів); x – величина вимірюваного параметра.

Опишемо процес попередньої обробки експериментальних даних, які представлені у приведеному вище вигляді (1), за допомогою програми Mathematica. Спочатку дані експортують у робочий документ Mathematica. Це виконується шляхом присвоєння змінній *InputData* вмісту файлу з записаними у нього експериментальними даними.

Після того, як дані завантажені у робочий документ Mathematica виконується вибір даних з потрібного каналу шляхом присвоєння змінній *DataList* елементу *InputData* з номером потрібного каналу. Попередня обробка даних для аналогових сигналів включає наступні операції з масивом *DataList*:

- 1) визначення моментів початку та закінчення процесу;
- 2) калібровка нуля;
- 3) згладжування даних;
- 4) формування двомірного масиву „час - вимірювана величина”.

Для цифрових сигналів (сигнали енкодерів) операції попередньої обробки даних такі:



- 1) визначення моментів початку та закінчення процесу;
- 2) формування двовірного масиву „час - вимірювана величина”;

Опишемо процедуру визначення моментів початку та закінчення процесу. Для цього використовують алгоритм, блок-схема якого показана на рис. 1. Алгоритм працює таким чином. Задають змінну i рівною одиниці. Далі встановлюють чутливість (*sensitivity*) до зміни даних. Наступний етап – порівняння модуля різниці сусідніх значень у масиві з величиною чутливості *sensitivity* (порівняється i -тий та $(i+1)$ -ий елементи масиву). Якщо модуль цієї різниці менший ніж величина чутливості *sensitivity*, то це свідчить про те, що вимірюваний параметр при переході від i -того до $(i+1)$ -ого дискретного значення не змінюється. Надалі збільшують значення змінної i на одиницю і знову повторюють порівняння. Якщо модуль цієї різниці i -того та $(i+1)$ -ого елементів масиву більший ніж чутливість *sensitivity*, то це значить, що вимірюваний параметр почав змінюватись. Тоді встановлюють значення початку процесу як відношення поточного значення змінної i до частоти опитування f . Процедура визначення моменту часу t_T закінчення процесу аналогічна описаній вище. Відмінність проявляється лише в тому, що порівняння сусідніх значень масиву *DataList* виконується у напрямку з кінця масиву до його початку.

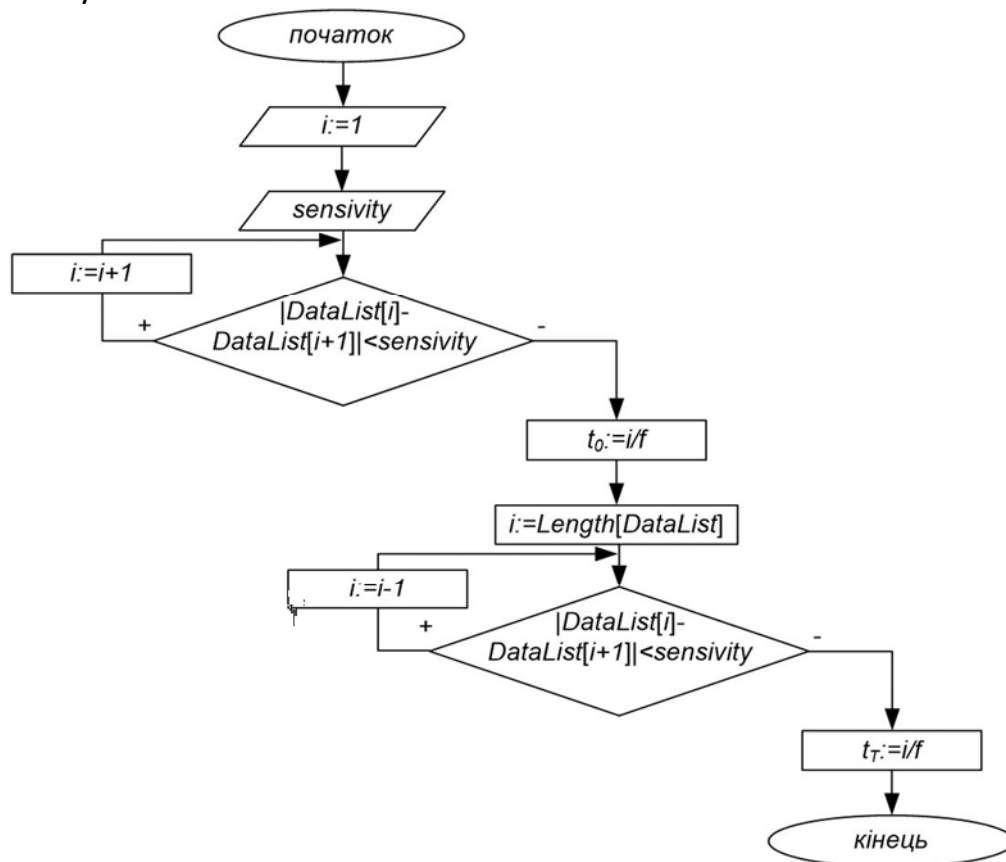


Рисунок. 1. Блок-схема алгоритма для визначення моментів початку t_0 та закінчення t_T процесу



Для калібровки нуля виконують дві операції: 1) визначення середнього значення параметра на проміжку від початку масиву *DataList* до значення, яке відповідає початку процесу t_0 (отримане значення є величиною „дрейфу нуля”); 2) віднімання всіх даних у масиві *DataList* від величини „дрейфу нуля”. Калібровку нуля виконують у випадку коли апіорі відомо, що вимірювальний параметр до початку процесу має нульове значення (наприклад, сила струму до вмикання двигуна у мережу).

Іноколи у даних спостерігаються промахи, тобто занадто великі значення, які не відповідають фізичній дійсності процесу. Вони, як правило, є наслідками електромагнітної дії на вимірювальне обладнання силового устаткування (двигунів, контакторів, автоматичних вимикачів тощо). Для усунення цих викидів використовується процедура згладжування даних. Блок-схема алгоритму згладжування даних зображена на рис. 2.

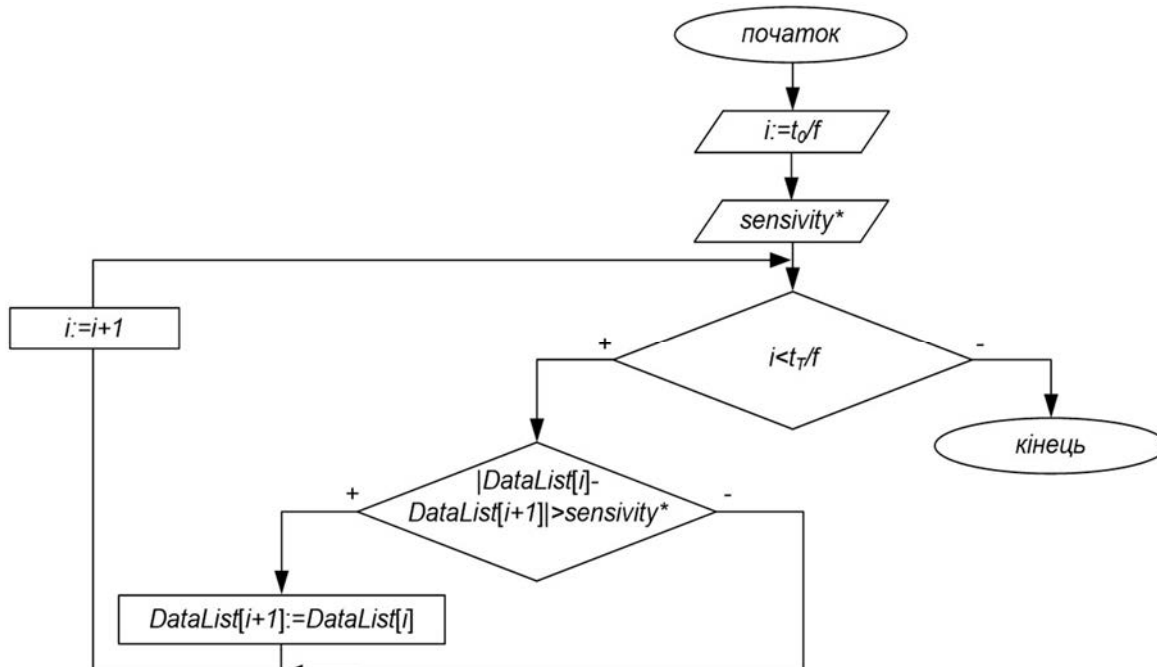


Рисунок. 2. Блок-схема алгоритму згладжування даних

Алгоритм працює так: встановлюють значення змінної i рівне t_0/f , що відповідає моменту початку процесу. Далі встановлюють значення чутливості $sensitivity^*$, яке виступає критерієм фізичної можливості збільшення вимірюваного параметра при переході від i -того елементу масиву до $(i+1)$ -го. Після цього виконується цикл, який полягає у порівнянні між собою сусідніх значень елементів масиву. Якщо модуль їх різниці більший ніж величина чутливості $sensitivity^*$, що вказує на наявність промаху у вимірюваннях, то $(i+1)$ -шому елементу масиву *DataList* присвоюють i -те значення елементу масиву. При кожному переході між значеннями вимірюваного параметра перевіряють поточне значення змінної i . Якщо змінна i стала рі-



вною t_T/f , що відповідає моменту закінчення процесу, то процедура згладжування даних закінчується.

Всі описані вище процедури виконуються із однимірним масивом. Для того, щоб встановити відповідність між значенням параметра та моментом його дії виконують формування двомірного масиву „час - вимірювана величина”. На рис. 3 показано блок-схему алгоритму формування двомірного масиву „час - вимірювана величина”. Спочатку об’являють однимірний двоелементний масив *element* та двомірний масив *NeededData*. Після цього встановлюють значення змінної *i*, що відповідає моменту початку t_0 процесу. Надалі для всього проміжку даних $[t_0/f; t_T/f]$ масиву *DataList* виконують такі операції: 1) у масив *element* додають значення дискретного часу та *i*-того елемента *DataList*, яка діє в цей момент часу; 2) у масив *NeededData* додають масив *element*; 3) очищають масив *element*. Після досягнення змінної *i* значення t_T/f , яке відповідає моменту закінчення процесу t_T , процедура формування двомірного масиву „час - вимірювана величина” закінчується.

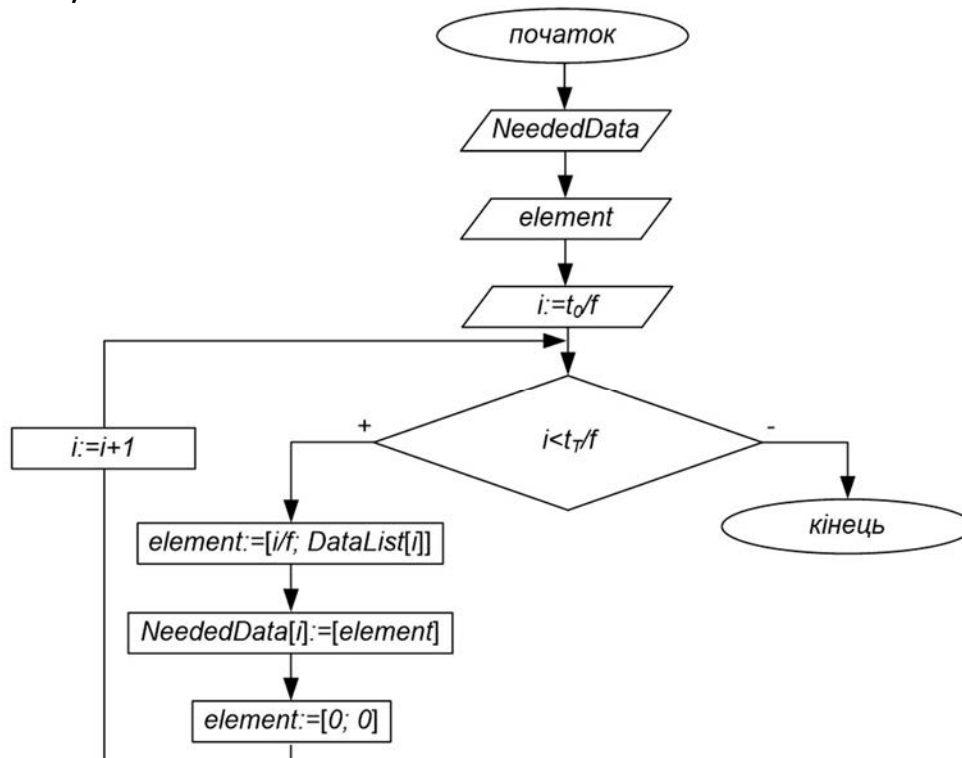


Рисунок. 3. Блок-схема алгоритму формування двомірного масиву „час - вимірювана величина”

Опишемо процедури, які виконуються для обробки сигналів з енкoderів (цифрових оптичних датчиків кутового або лінійного переміщення). Ми виокремлюємо процедури попередньої обробки даних з енкoderів оскільки вихідні сигнали з цих датчиків представляються у вигляді імпульсів напруги.



Перша процедура дозволяє визначити моменти початку та закінчення процесу. Опишемо алгоритм знаходження моменту початку процесу t_0 (рис. 4). Спочатку об'являють змінну i та величину чутливості $sensitivity^{**}$, яка вибирається меншою, ніж різниця рівнів логічної одиниці та логічного нуля енодера. Надалі перевіряють різницю $(i+1)$ -го та i -того значень елементів масиву $DataList$. Якщо ця різниця більша аніж встановлена чутливість $sensitivity^{**}$, то це означає, що у масиві зафіксовано перехід від логічного нуля до одиниці, тобто фронт першого імпульсу напруги енодера. Після цього присвоюють значення змінної i іншій змінній i_0 .

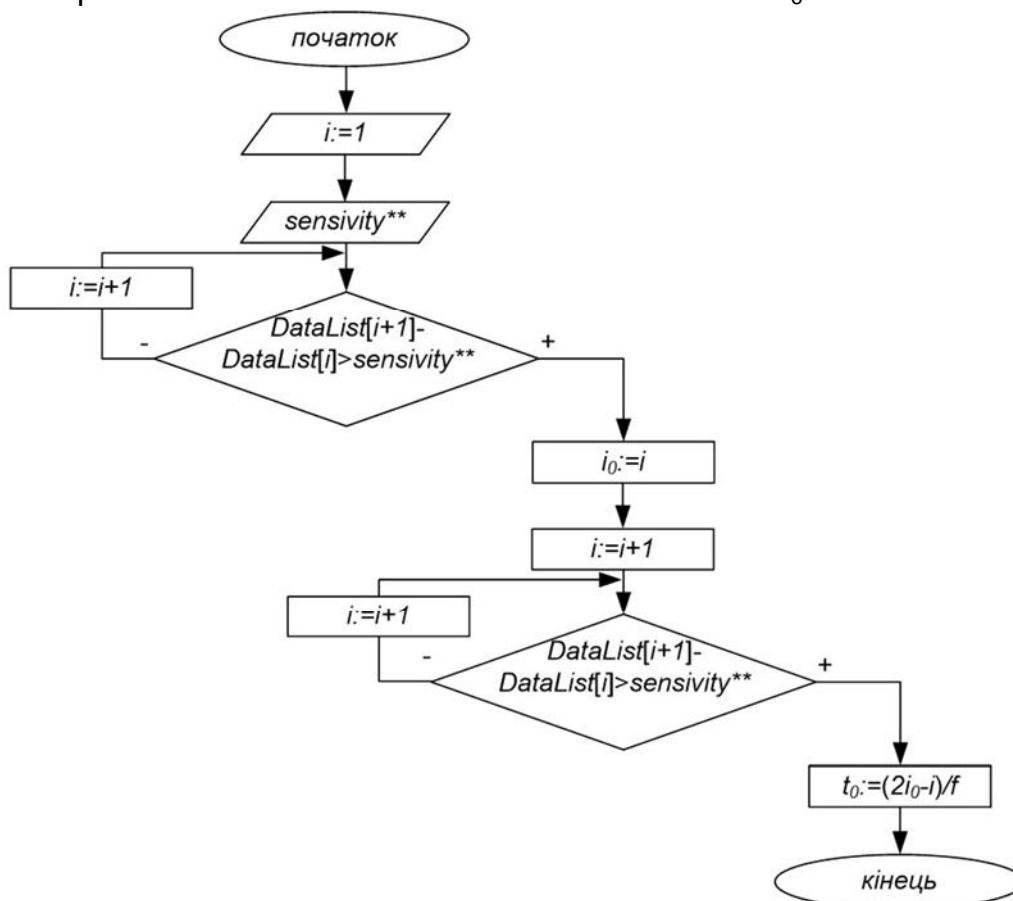


Рисунок. 4. Блок-схема алгоритму для визначення моменту початку t_0 процесу

Аналогічно відшукують номер елементу масива $DataList$ для фронту другого імпульсу. Момент початку процесу визначається за наступною формулою:

$$t_0 = \frac{2i_0 - i}{f}, \tag{2}$$

де f – частота опитування вимірювального каналу, Гц. Це означає, що момент початку процесу рівний різниці моменту настання фронту першого імпульсу та тривалості першого імпульсу.



Для моменту закінчення процесу t_T працює подібний алгоритм з тією лише різницею що „рух” в масиві *DataList* виконується з кінця на початок.

В результаті вимірювання кінематичних функцій руху елемента механізму отримують одномірний масив, який включає два рівні напруги: рівень логічного нуля та одиниці. Для того, щоб із вихідних даних сформува-ти дані положення елемента механізму у часі необхідно використати алго-ритм, блок-схема якого зображена на рис. 5.

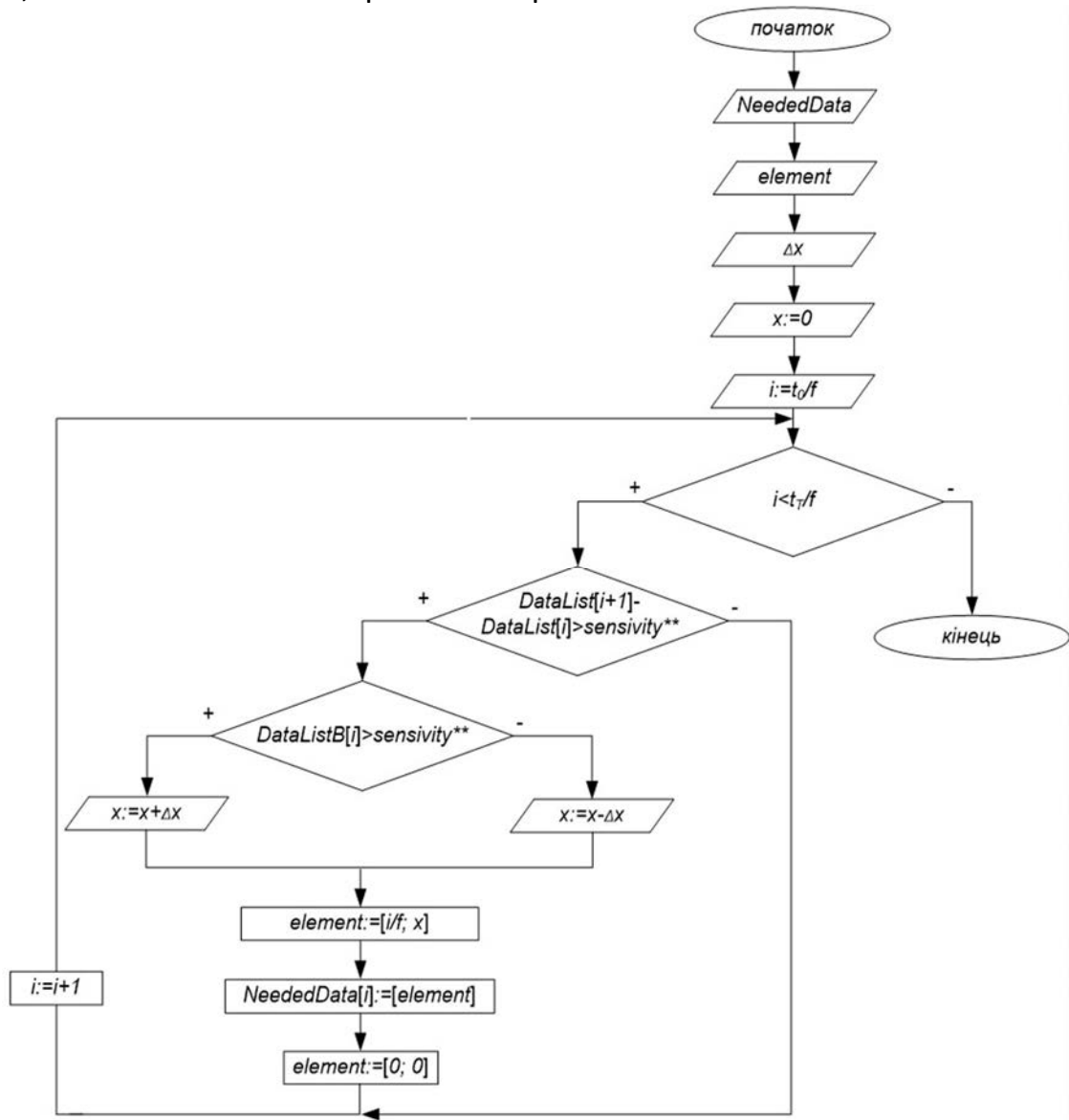


Рисунок. 5. Алгоритм формування двомірного масиву „час - вимірювана величина”

Для формування двомірного масиву „час - вимірювана величина” спочатку об’являють двомірний масив *NeededData*, двоелементний масив *element*, встановлюють ціну одного імпульсу напруги Δx , поточне значення положення елемента (наприклад $x=0$) та переходять у масиві *DataList* до змінної i , яка відповідає моменту початку процесу. Після цього йде послі-



довний перехід від одного елементу масиву *DataList* до іншого, що викликає збільшенням змінної *i*. При виконанні переходу йде перевірка умови досягнення значення змінної *i*, яке відповідає моменту закінчення процесу. Якщо умова $i < t_T/f$ не виконується, то це означає, що не всі дані оброблені і процедура повинна бути продовжена. У іншому випадку ($i = t_T/f$) процедура закінчується.

Тілом циклу для даного алгоритму є порівняння різниці значень сусідніх елементів масиву *DataList* із величиною чутливості *sensitivity***. Якщо умова $DataList[i+1]-DataList[i] > sensitivity**$ виконується, це значить, що у масиві *DataList* ми „напрапили” на фронт імпульса (перехід від логічного нуля до логічної одиниці). Фронт імпульса означає переміщення датчика на відстань Δx . Для того, щоб визначити напрямок переміщення необхідно перевірити рівень сигналу у каналі В енкодера. Якщо у каналі В енкодера діє рівень логічного нуля, то необхідно поточне положення датчика збільшити на величину Δx ($x := x + \Delta x$). Навпаки, якщо при фронті імпульса у масиві *DataList* у каналі В енкодера діє рівень логічної одиниці, то положення енкодера зменшують на величину кроку Δx ($x := x - \Delta x$). Сигнали з каналу В енкодера записані у масив даних *DataListB*.

Наступними діями у циклі є запис даних дискретного часу та поточного положення *x* у масив *element*, запис масиву *element* у масив *NeededData*, обнулення масиву *element* та збільшення змінної *i* на одиницю.

Висновок. Розроблені у дослідженні алгоритми попередньої обробки експериментальних даних дозволяють полегшити обробку даних представлених у вигляді багатомірних масивів. Алгоритми можуть бути реалізовані за допомогою математичних пакетів Mathematica, MathCad, MatLab тощо. Використання запропонованої методики особливо ефективно у випадках значного об’єму даних.

ЛІТЕРАТУРА

1. Дейпер Н. Прикладной регрессионный анализ. Кн. 1 / Н. Дейпер. – [пер. с английского Ю.П. Адлер, В.Г. Горский] – [2-е изд., перераб. и доп.]. – М.: Финансы и статистика, 1987. – 366 с.
2. Гайдышев И. Анализ и обработка данных: специальный справочник / И. Гайдышев. – СПб: Питер, 2001 – 752 с.
3. Мостеллер Ф. Анализ данных и регрессия / Ф. Мостеллер, Дж. Тьюки. – [пер. с английского Ю.Н. Благовещенский]. - М.: Финансы и статистика, 1987. – 320 с.
4. Дьяконов В.П. Mathematica 5.1/5.2/6. Программирование и математические вычисления / В.П. Дьяконов. – М.: ДМК-Пресс, 2008. – 576 с.
5. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В.П. Дьяконов, Круглов В. – СПб: Питер, 2001. – 480 с.