

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

На правах рукопису

Олеськів Ольга Михайлівна

УДК 006: 004.4: 621.317

**МЕТРОЛОГІЧНА ПЕРЕВІРКА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ЗАСОБІВ ВИМІРЮВАННЯ**

05.01.02 – стандартизація, сертифікація та метрологічне забезпечення

Дисертація на здобуття наукового ступеня
кандидата технічних наук

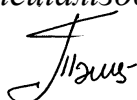
Наукові керівники:
доктор технічних наук,
професор **Микитин І. П.**

доктор технічних наук,
професор **Фрьогліх Т.**

Ідентичність всіх примірників дисертації

ЗАСВІДЧУЮ:

*Учений секретар спеціалізованої
вченої ради*



/Т. Бубела/

Львів - 2016

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ПРОЦЕДУР ТА МЕТОДІВ ПЕРЕВІРКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАННЯ	10
1.1 Інтегроване та прикладне програмне забезпечення	10
1.2 Вплив структури засобу вимірювання на можливість проведення метрологічної перевірки програмного забезпечення.....	12
1.3 Процедури перевірки програмного забезпечення засобів вимірювання.....	19
1.4 Методи перевірки програмного забезпечення засобів вимірювання.....	23
Висновки до розділу 1	35
РОЗДІЛ 2 ОСОБЛИВОСТІ МЕТРОЛОГІЧНОЇ ПЕРЕВІРКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАННЯ ТА КІБЕР-ФІЗИЧНИХ СИСТЕМ.....	36
2.1 Особливості метрологічної перевірки програмного забезпечення засобів вимірювання	36
2.2 Особливості метрологічної перевірки програмного забезпечення компонентів кібер-фізичних систем	39
Висновки до розділу 2	49
РОЗДІЛ 3 ГЕНЕРАТОР ЗРАЗКОВИХ ЦИФРОВИХ ПОСЛІДОВНОСТЕЙ.....	50
3.1 Математична модель генератора зразкових цифрових послідовностей.....	51
3.2 Алгоритм роботи генератора зразкових цифрових послідовностей.....	69

3.3 Інтерфейс користувача генератора зразкових цифрових послідовностей.....	80
Висновки до розділу 3	87
РОЗДІЛ 4 ДОСЛІДЖЕННЯ ПОХИБОК ТА НЕПЕВНОСТЕЙ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАННЯ НА ПРИКЛАДІ ФУНКЦІЇ ПЕРЕТВОРЕННЯ ФУР'Є.....	88
4.1 Дослідження похибки функції перетворення Фур'є програмного пакету MATLAB для даних типу double	90
4.2 Дослідження похибки функції перетворення Фур'є програмного пакету MATLAB для даних типу int.....	93
4.3 Дослідження похибки та непевності функції перетворення Фур'є програмних пакетів MATLAB та MathCAD з урахуванням похибок аналого-цифрового перетворювача	100
Висновки до розділу 4	114
ВИСНОВКИ	115
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	117
Додаток А Програмна реалізація генератора зразкових цифрових послідовностей	130
Додаток Б Акти впроваджень результатів дисертаційної роботи	145

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АЦП	аналого-цифровий перетворювач
ВСК	вбудована система керування
ЗВ	засіб вимірювання
ІВС	інформаційно-вимірювальна система
ІПП	інтелектуальний первинний перетворювач
КФС	кібер-фізична система
ПЗ	програмне забезпечення
ПК	персональний комп'ютер
fft()	функція перетворення Фур'є

ВСТУП

Актуальність теми. При проектуванні та розробленні сучасних засобів вимірювання (ЗВ) спостерігається тенденція до зменшення аналогової частини та спрощення цифрової частини за рахунок використання мікроконтролерів та мікропроцесорів, а основні обчислення, зокрема результату вимірювання, відбуваються із використанням програмного забезпечення (ПЗ). Тому «вага» ПЗ у сучасних ЗВ постійно зростає. Основні операції опрацювання результатів вимірювання, а саме усереднення, апроксимація, фільтрація, інтерполяція, перетворення Фур'є тощо, реалізуються переважно програмним способом. Некоректна програмна реалізація алгоритмів розрахунку та опрацювання результатів вимірювання, невідповідність ПЗ вимірювальній задачі, випадкова або навмисна зміна функцій ПЗ можуть призвести до невідповідності реальної похибки вимірювання задекларованим.

Під час метрологічної перевірки ПЗ ЗВ постає проблема вибору методів та процедур перевірки. Існуючі методи та процедури не завжди дозволяють коректно проводити метрологічну перевірку ПЗ ЗВ. Тому слід приділяти увагу процедурам, методам, засобам та методикам метрологічної перевірки ПЗ ЗВ, їх удосконаленню, виконувати аналіз їх переваг та недоліків.

Крім того, швидкими темпами розвивається новий напрямок – кіберфізичні системи. Такі системи без втручання людини можуть змінювати своє ПЗ залежно від поставленого завдання. Тобто, буде застосовуватися так зване гнучке ПЗ. У такому випадку, після зміни ПЗ, необхідним є проведення його метрологічної перевірки.

З урахуванням тенденцій розвитку вимірювальної техніки актуальним є розроблення засад метрологічної перевірки ПЗ ЗВ, створення еталонних засобів та методів метрологічної перевірки ПЗ ЗВ.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась в рамках пріоритетних напрямків розвитку науки і техніки в Україні, а також в рамках держбюджетної науково-дослідної роботи «Вимірювання температури мікро- та наноструктурованих об'єктів методом комбінаційного розсіювання світла» (номер державної реєстрації 0115U000431) з 01.01.2015 р. до 31.12.2016 р.

Мета і завдання дослідження. Метою дисертаційної роботи є розроблення засад метрологічної перевірки ПЗ ЗВ.

Для досягнення цієї мети необхідно виконати:

- 1) аналітичний огляд процедур та методів перевірки ПЗ ЗВ;
- 2) дослідження особливостей метрологічної перевірки ПЗ кібер-фізичних систем;
- 3) пошук шляхів покращення існуючих та розроблення нових методів метрологічної перевірки ПЗ ЗВ;
- 4) роботи з розроблення генератора зразкових цифрових послідовностей для метрологічної перевірки ПЗ ЗВ;
- 5) дослідження впливу похибок аналогової частини ЗВ на результати вимірювання, які обчислюються із використанням ПЗ ЗВ;
- 6) дослідження похибок та непевностей ПЗ ЗВ.

Об'єкт дослідження – метрологічна перевірка програмного забезпечення засобів вимірювань.

Предмет дослідження – генератор зразкових цифрових послідовностей для метрологічної перевірки програмного забезпечення засобів вимірювання.

Методи дослідження. Застосовані теоретичні та експериментальні методи досліджень, які ґрунтуються на загальній теорії вимірювання, теорії

вимірювальних сигналів, методах імітаційного моделювання, що дає змогу перевірити адекватність та точність побудованих математичних моделей, теорії похибок та непевності результатів вимірювань, теорії статистичного аналізу, що дає змогу оцінити похибки та непевності результатів вимірювання, що обчислюються із використанням ПЗ.

Наукова новизна одержаних результатів:

- 1) розвинуто класифікацію структур ЗВ за критерієм типу використовуваного ПЗ, що дозволяє оцінити можливість проведення метрологічної перевірки ПЗ ЗВ;
- 2) вперше запропоновано метод багаторівневої метрологічної перевірки ПЗ компонентів кібер-фізичних систем, що дає змогу покращити безпеку функціонування кібер-фізичних систем та полегшити процес метрологічної перевірки їх ПЗ;
- 3) вперше запропоновано метод метрологічної перевірки ПЗ ЗВ на основі методу «чорної скриньки» та методу генерування «еталонних» даних, який дозволяє уніфікувати процес метрологічної перевірки ПЗ ЗВ та суттєво зменшити його тривалість;
- 4) розроблено математичну модель генератора зразкових цифрових послідовностей із врахуванням параметрів вимірюваного сигналу та аналогової частини ЗВ, що дає змогу реалізувати генератор зразкових цифрових послідовностей практично в будь-якому ПЗ сучасних ЗВ, а також як окремий засіб метрологічної перевірки.

Практичне значення одержаних результатів:

- 1) розроблено генератор зразкових цифрових послідовностей, який дозволяє формувати зразкові цифрові послідовності для метрологічної перевірки ПЗ ЗВ;
- 2) розроблено інтерфейс користувача генератора зразкових цифрових послідовностей, який дає змогу у зручній формі задавати параметри сигналів та аналогової частини ЗВ;

- 3) створений генератор зразкових цифрових послідовностей може використовуватися на етапі розроблення програмного забезпечення як засіб відлагоджування та перевірки ПЗ;
- 4) використання генератора зразкових цифрових послідовностей забезпечує отримання залежності похибки результатів обчислення програмним забезпеченням від параметрів вимірювального сигналу та параметрів аналогової частини ЗВ, що дозволяє одержати оцінку точності результату обчислення даним ПЗ;
- 5) генератор зразкових цифрових послідовностей можна інтегрувати у будь-який програмний пакет сучасних ЗВ (типу MATLAB, LabVIEW тощо) як окремий блок, що призначений для відлагоджування та метрологічної перевірки програмного забезпечення.

Наукові та практичні результати виконаних досліджень використано в навчальному процесі, в лекційних курсах і лабораторних роботах, які проводяться для студентів кафедри «Інформаційно-вимірювальні технології» Національного університету «Львівська політехніка» освітньо-кваліфікаційного рівня «Магістр», зокрема в дисципліні «Тестування інформаційно-вимірювальних систем».

Особистий внесок здобувача. Основні теоретичні та експериментальні результати, що викладені в дисертації, отримані автором самостійно. У наукових працях, опублікованих у співавторстві дисертантові належать: [1] – аналіз процедур та методів метрологічної перевірки ПЗ ЗВ та представлення їх переваг і недоліків; [2] – класифікація структур ЗВ відповідно до можливості доступу до ПЗ; [3, 4] – аналіз первинних перетворювачів та засобів вимірювання кібер-фізичних систем, запропоновано проведення багаторівневої метрологічної перевірки ПЗ кібер-фізичних систем.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися на таких конференціях: III науково-технічна конференція «Обчислювальні методи і системи перетворення

інформації» 25 - 26 вересня 2014, Львів; Всеукраїнська науково-технічна конференція молодих вчених у царині метрології «Technical Using of Measurement – 2015», 2 - 6 лютого 2015, м. Славське; II Міжнародна науково-практична конференція «Управління якістю в освіті та промисловості: досвід, проблеми та перспективи», 28 - 30 травня 2015, Львів; Всеукраїнська науково-технічна конференція молодих вчених у царині метрології «Technical Using of Measurement – 2016», 1 - 5 лютого 2016, м. Славське.

Публікації. За тематикою дисертаційної роботи опубліковано 9 наукових праць, серед яких 4 статі у наукових журналах, що входять до переліку наукових фахових видань України, 1 стаття у науковому періодичному виданні іншої держави.

Структура та обсяг роботи. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та двох додатків. Загальний обсяг дисертації становить 147 сторінок друкованого тексту, у тому числі 107 сторінок основного тексту, містить 62 рисунки, 7 таблиць, список використаних джерел з 110 найменувань.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД ПРОЦЕДУР ТА МЕТОДІВ ПЕРЕВІРКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАННЯ

У першому розділі розглянуто типи програмного забезпечення засобів вимірювання, процедури і методи перевірки та тестування програмного забезпечення засобів вимірювання, проаналізовано можливість проведення метрологічної перевірки програмного забезпечення засобів вимірювання різних структур.

У роботі під **метрологічною перевіркою програмного забезпечення засобів вимірювання** розуміється сукупність методів та засобів, призначених для встановлення похибки результату вимірювання, який розраховується із використанням ПЗ, а також підтвердження придатності ПЗ для використання, у випадку, якщо виробник ПЗ надає інформацію про похибку розрахунків, що виконані даним програмним забезпеченням. Під **похибкою програмного забезпечення засобів вимірювання** розуміється кількісна оцінка впливу розрядності та частоти дискретизації АЦП, який використовується у ЗВ, на результат вимірювання, що обчислюється програмним забезпеченням. Під **генератором зразкових цифрових послідовностей** розуміється засіб (програма), що формує зразкові цифрові послідовності, які використовуються для метрологічної перевірки ПЗ ЗВ.

1.1 Інтегроване та прикладне програмне забезпечення

ПЗ ЗВ призначене для реєстрації, опрацювання, відображення та збереження результатів вимірювань, і є функціональною частиною засобів

вимірювання та постачається з комплектом програмних документів [5, 6]. За матеріалами рекомендацій та інших метрологічних документів [5-13] класифікація програмного забезпечення не є однозначною. За однією з класифікаційних ознак ПЗ поділяється на інтегроване та прикладне.

Інтегроване програмне забезпечення засобів вимірювання – це ПЗ у складі автономного приладу цільового призначення, що має строго встановлений набір вимірювальних функцій, причому перетворення фізичної величини, що вимірюється, та опрацювання результатів вимірювань виконується тільки з використанням апаратних та програмних засобів ЗВ [6].

Основні характеристики інтегрованого програмного забезпечення засобів вимірювання [5, 9]:

- розроблено для проведення вимірювань;
- інтерфейс користувача призначений тільки для вимірювальних цілей;
- переважно зберігається в постійній пам'яті;
- програмне середовище постійне, відсутні засоби для програмування або зміни програмного забезпечення;
- операційна система, що має призначену для користувача оболонку, відсутня;
- може підтримуватися збереження вимірюваних даних на інтегрованому або віддаленому накопичувачі даних;
- переважно доступ до програмного забезпечення відсутній.

Прикладне програмне забезпечення засобів вимірювання – це програмне забезпечення, що розміщується на жорстких носіях та працює під певною операційною системою [6].

Основні характеристики прикладного ПЗ ЗВ [5, 9]:

- може виконуватися на будь-якій операційній системі.
- на додаток до ПЗ ЗВ, яке не може модифікуватися після затвердження типу, в системі можуть також встановлюватися або постійно знаходитися в пам'яті інші ПЗ, які можуть бути модифіковані;

- у ПЗ може бути передбачена можливість роботи з файлами;
- можливі два режими роботи інтерфейсу користувача: режим вимірювання та режим загального користування;
- збереження даних може бути локалізованим або віддаленим;
- є можливість видалення і переустановлення ПЗ;
- переважно є доступ до файлів ПЗ.

Однією із основних характеристик, яка впливає на можливість проведення метрологічної перевірки, є наявність доступу до ПЗ. Тому доцільним є проведення аналізу ЗВ за структурою та дослідження впливу структури ЗВ на можливість доступу до ПЗ та проведення його метрологічної перевірки.

1.2 Вплив структури засобу вимірювання на можливість проведення метрологічної перевірки програмного забезпечення

Структура ЗВ може впливати на процес, а інколи, і можливість проведення метрологічної перевірки ПЗ. Не завжди є доступ до ПЗ, що ускладнює, а в деяких випадках і унеможлиблює перевірку ПЗ. З іншого боку, відсутність доступу до ПЗ покращує його захищеність, так як немає можливості його навмисної зміни або пошкодження.

Провівши аналіз структур засобів вимірювання [5, 10-24], умовно їх можна розділити на:

- автономні засоби вимірювання;
- засоби вимірювання, до складу яких входить персональний комп'ютер (ПК);
- інформаційно-вимірювальні системи;
- вбудовані системи керування;

- кібер-фізичні системи.

Автономні засоби вимірювання (рисунок 1.1) є цільовими пристроями, що мають встановлений набір вимірювальних функцій. Перетворення вимірюваної величини і опрацювання вимірювальних даних виконується з використанням внутрішніх апаратних і програмних засобів.

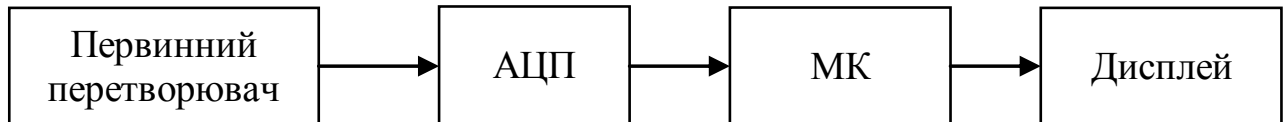


Рисунок 1.1 – Типова структурна схема автономного ЗВ

Автономні ЗВ характеризуються такими технічними особливостями:

- закритий корпус;
- інтерфейс для передавання даних на різні пристрої;
- результати вимірювань можуть зберігатися на зовнішніх накопичувачах, тощо [5, 14].

ПЗ автономного ЗВ є інтегрованим, а тому недоступним для метрологічної перевірки. Проводиться тільки метрологічна перевірка ЗВ в цілому. Метрологічна перевірка ПЗ можлива лише на етапі проектування ЗВ.

Як один з різновидів автономного ЗВ можна розглядати автономний ЗВ з додатковим входом (рисунок 1.2).

Додатковий вхід дозволяє реалізувати перевірку ПЗ не лише на етапі розроблення ЗВ, а й під час експлуатації шляхом подавання цифрових даних безпосередньо на ПЗ [10]. Без введення додаткового входу метрологічна перевірка інтегрованого ПЗ є неможливою.

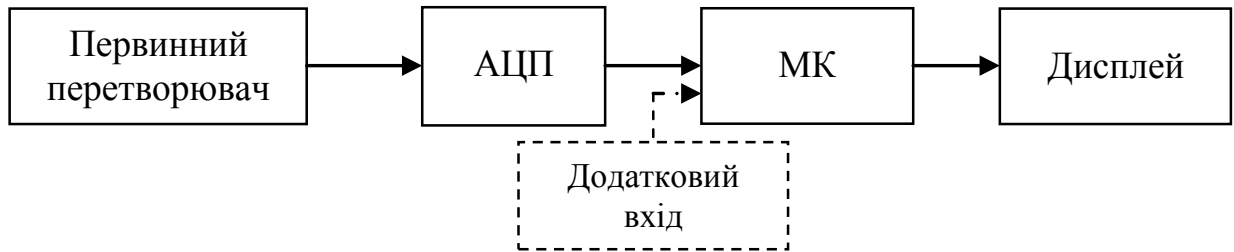


Рисунок 1.2 – Структурна схема автономного ЗВ з додатковим входом

На рисунку 1.3 представлено типову структурну схему ЗВ, до складу якого входить ПК [5, 14]. Для таких ЗВ програмне забезпечення може знаходитися:

- тільки на ПК;
- частково на мікроконтролері (МК) та частково на ПК.

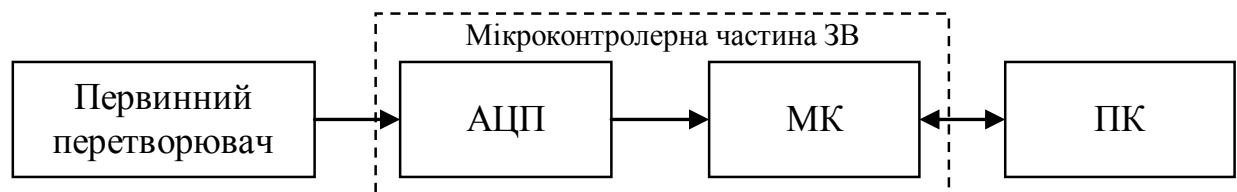


Рисунок 1.3 – Типова структурна схема ЗВ, до складу яких входять ПК

Обмін інформацією між ПК та мікроконтролерною частиною ЗВ може відбуватися через будь-який стандартний інтерфейс ПК. Якщо опрацювання результатів вимірювання відбувається лише на ПК, то немає перешкод для метрологічної перевірки ПЗ. У випадку, коли частина ПЗ розміщена на мікроконтролері, його перевірка є аналогічною перевірці ПЗ автономних ЗВ.

Інформаційно-вимірювальні системи (ІВС) (рисунок 1.4) є сукупністю функціонально об'єднаних вимірювальних, обчислювальних та інших допоміжних технічних засобів, що призначені для отримання

вимірювальної інформації, її перетворення, опрацювання і представлення споживачу в необхідному вигляді [15, 16, 27-29].

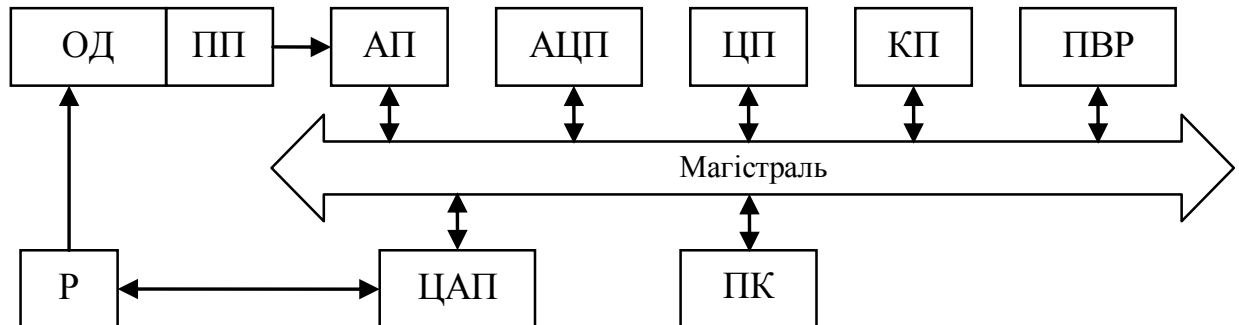


Рисунок 1.4 – Узагальнена структурна схема інформаційно-вимірювальної системи: ОД - об'єкт дослідження; ПП – первинні перетворювачі; Р – регулятори; АП – аналогові перетворювачі; АЦП – аналогово-цифрові перетворювачі; ЦАП – цифро-аналогові перетворювачі; ЦП – цифрові прилади; КП – канали передавання; ПВР – пристрої відображення та реєстрації результатів вимірювання; ПК – персональний комп'ютер [16].

ІВС використовуються для автоматичного контролю, діагностування, ідентифікації тощо [16, 17, 27]. При реалізації ІВС для конкретної вимірювальної задачі наявність всіх функціональних блоків (рисунок 1.4) не є обов'язковою. Характерними сферами застосування ІВС є управління технологічними процесами, наукові дослідження, тощо. У випадку, якщо змінюється структура ІВС або алгоритм опрацювання результатів вимірювання, то така система, і ПЗ зокрема, можуть потребувати повторної метрологічної перевірки.

Інформаційно-вимірювальні системи можуть використовувати як інтегроване, так і прикладне програмне забезпечення. У першому випадку доступ до програмного забезпечення переважно закритий і блок ІВС, який використовує таке ПЗ потрібно перевіряти в цілому. У другому випадку

переважно є доступ до програмного забезпечення і можлива реалізація метрологічної перевірки програмного забезпечення.

Вбудовані системи керування (ВСК) – це системи керування, конструктивно інтегровані в устаткування (рисунок 1.5).

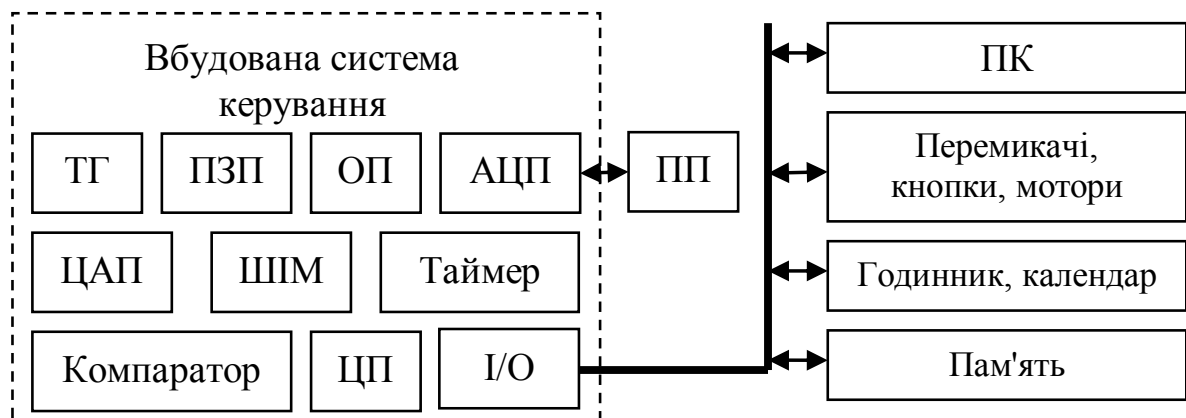


Рисунок 1.5 – Архітектура типової вбудованої системи: ПП – первинні перетворювачі; ОП – оперативна пам'ять; ПЗП – постійний запам'ятовуючий пристрій; ТГ – тактовий генератор; ШІМ – широтно-імпульсний модулятор; ЦП – центральний процесор; І/О – пристрої вводу-виводу

Обов'язковий для багатьох ВСК компонент – інтерфейс із системою керування більш високого рівня – промисловим комп'ютером. Наявність такого інтерфейсу дозволяє вирішувати завдання комплексної автоматизації групою одиниць технологічного устаткування, будувати розподілені системи керування [18, 23, 30-32].

Характерні особливості ВСК:

- вирішують певні специфічні завдання;
- будуються на основі різноманітного асортименту процесорів та архітектур;

- чутливі до обмежень за вартістю, габаритами, відмовами живлення, тепловиділенням тощо;
- є системами реального часу (з відповідним програмним забезпеченням, зокрема операційною системою);
- потребують менше ресурсів для свого функціонування, ніж звичайний комп'ютер;
- переважно зберігають ПЗ в постійній пам'яті.

Як і для інформаційно-вимірювальних систем, програмне забезпечення ВСК може бути як інтегрованим, коли опрацювання результатів вимірювання відбувається тільки у ВСК, так і прикладним, якщо частина опрацювання відбувається на ПК. Програмне забезпечення ВСК перевіряється так само, як і ПЗ автономних ЗВ, а для програмного забезпечення, яке розміщене на промисловому комп'ютері, можлива реалізація метрологічної перевірки.

Кібер-фізичні системи (КФС) – це системи [19-21, 33-37], що об'єднують інформаційні і програмні компоненти з механічними або електронними компонентами, які «спілкуються» за допомогою комунікаційної інфраструктури, такої як Інтернет в режимі реального часу (рисунок 1.6).

Базові технології КФС можна знайти в різних областях науки і техніки, зокрема у вбудованих системах керування, інтернет-технологіях і сенсорних мережах. Елементну базу КФС складають електронні, оптичні, механічні та інші фізичні компоненти. Контрольні компоненти КФС зазвичай реалізуються на вбудованих системах керування, які охоплюють автономне керування, функції регулювання і спостереження.

Основними особливостями КФС є:

- робота в реальному масштабі часу;
- різні, часто важкі, умови експлуатації;
- автономність роботи;
- високі вимоги до надійності і безпеки функціонування;

- обмежені ресурси;
- критичні застосування, пов'язані зі здоров'ям і життям людини [18].

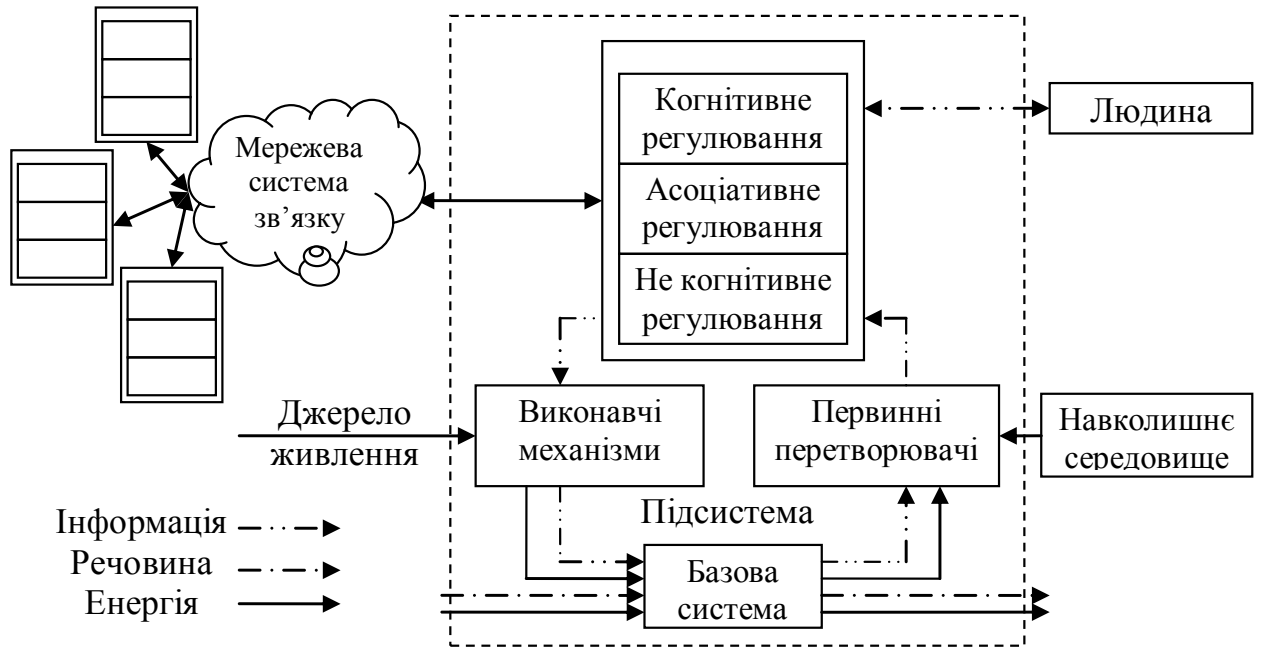


Рисунок 1.6 – Типова структурна схема кібер-фізичної системи [19].

Складність і питома вага програмної складової в КФС стрімко зростає. В перспективі компоненти КФС, використовуючи можливості Інтернету, можуть змінювати свою конфігурацію та об'єднуватися в єдину систему для виконання певної задачі. Під час конфігурації змінюються функції КФС та використовуються інші блоки та програмні засоби. Тому для коректної роботи такої системи після конфігурації потрібно проводити метрологічну перевірку ПЗ КФС.

На рисунку 1.7 представлено класифікацію ЗВ за критерієм типу використовуваного ПЗ. Аналіз типових структур засобів вимірювання показав, що метрологічну перевірку ПЗ ЗВ можна провести тільки у випадку, якщо є доступ до ПЗ (прикладне програмне забезпечення). Для ЗВ, які мають інтегроване програмне забезпечення, на даний час перевірку ПЗ провести

неможливо. Для реалізації метрологічної перевірки ПЗ таких ЗВ необхідно забезпечити додатковий цифровий вхід (рисунок 1.2), який дозволить подавати цифрові дані безпосередньо на ПЗ.

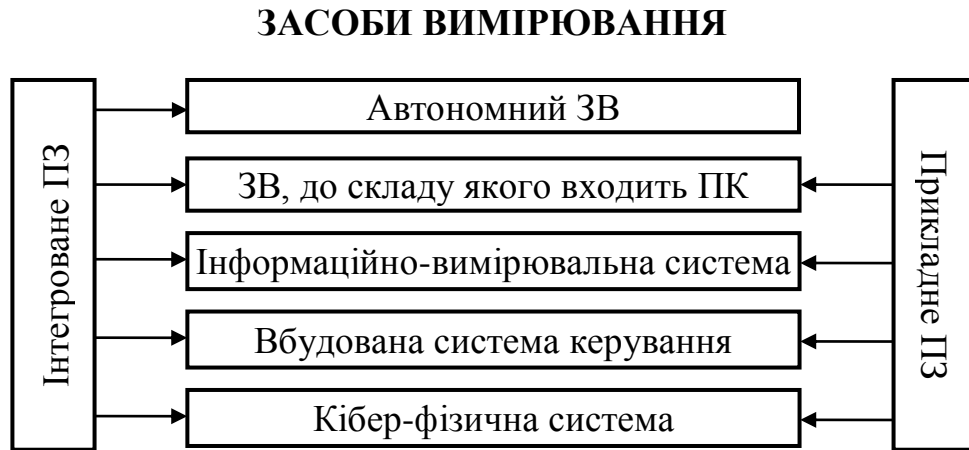


Рисунок 1.7 – Класифікація ЗВ за критерієм типу використовуваного ПЗ

Розглядалась необхідність проведення метрологічної перевірки ПЗ ЗВ під час експлуатації. ЗВ, які мають у своєму складі інтегроване ПЗ, або прикладне ПЗ, яке не змінюється, не потребують метрологічної перевірки ПЗ під час експлуатації. Кібер-фізичні системи під час експлуатації можуть змінювати свою структуру, також ПЗ [38-40], тому потрібно проводити метрологічну перевірку програмного забезпечення кібер-фізичних систем під час експлуатації.

1.3 Процедури перевірки програмного забезпечення засобів вимірювання

Результатом метрологічної перевірки ПЗ ЗВ має бути підтвердження або спростування того, що програмне забезпечення ЗВ відповідає вимогам,

які описані у нормативних документах та має характеристики, які описані в документації до даного ПЗ. Важливо зазначити, що метрологічній перевірці підлягає не все ПЗ, а та його частина, яка може впливати на метрологічні характеристики ЗВ. До неї відносяться програми і програмні модулі, що виконують функції збору, передавання, опрацювання, зберігання та представлення вимірювальної інформації [9].

За матеріалами роботи [9] до ПЗ, яке може впливати на метрологічні характеристики ЗВ відносяться:

- 1) програми і програмні модулі, які беруть участь в опрацюванні результатів вимірювань;
- 2) ПЗ ЗВ, що бере участь в обчисленні результату вимірювань;
- 3) програми і програмні модулі, що здійснюють представлення вимірювальної інформації, її зберігання та передавання, ідентифікацію та оновлення ПЗ, захист ПЗ і даних;
- 4) компоненти захищеного інтерфейсу для обміну даними між програмними модулями ЗВ.

Пункти 3 і 4 доцільніше віднести до загальної перевірки ПЗ, а не метрологічної, тому що ніяких вимірювальних чи обчислювальних дій, які впливають на результати вимірювання, вони не виконують. Дані пункти характеризують правильність функціонування програмного забезпечення в цілому [1].

Перевірка програмного забезпечення є процедурою встановлення правильності і однозначності контрольованих функцій ПЗ та даних. На рисунку 1.8 представлено класифікацію процедур перевірки ПЗ ЗВ [41].

Вибір процедури перевірки визначається вимогами законодавчих та нормативних документів, а також бажанням розробника або користувача ПЗ ЗВ підтвердити його відповідність встановленим вимогам [1, 5-10, 41-48, 50-72].

Процедура випробувань з метою затвердження типу ЗВ повинна передбачати ідентифікацію та оцінку впливу програми на метрологічні характеристики ЗВ, а також аналіз конструкції з метою запобігання несанкціонованих налаштувань програмного забезпечення ЗВ та втручань, які можуть призвести до отримання некоректних результатів вимірювань. Розроблений за результатами випробувань проект опису типу засобу вимірювань, крім характеристик ЗВ, повинен містити також опис програмного забезпечення, ідентифікаційні дані, кількісні характеристики оцінки впливу на метрологічні характеристики ЗВ і рівень захисту від ненавмисних і навмисних змін [1, 50, 55, 69, 71, 73].

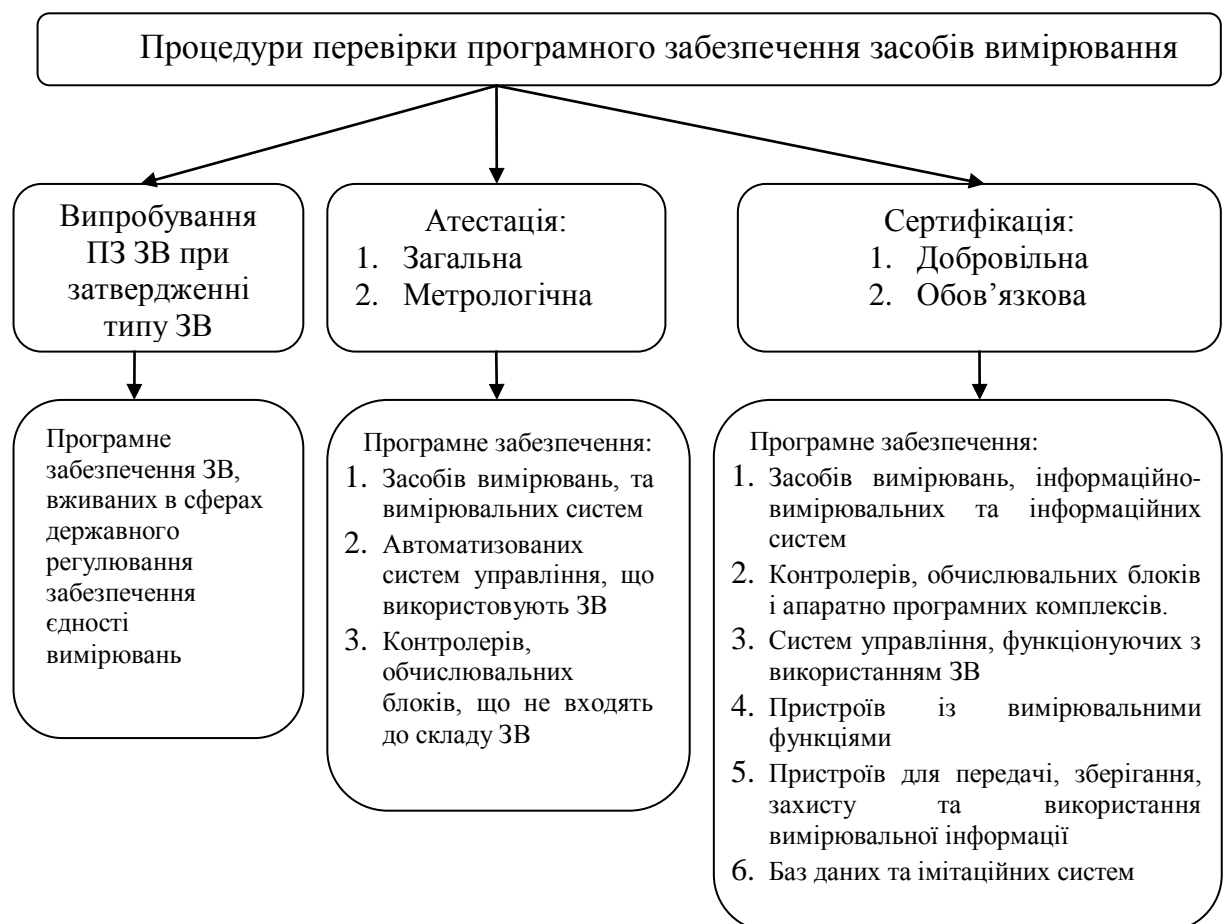


Рисунок 1.8 – Процедури перевірки програмного забезпечення засобів вимірювання

Атестація ПЗ ЗВ – це дослідження програмного забезпечення з метою визначення його характеристик, властивостей та ідентифікаційних даних, а також підтвердження відповідності встановленим вимогам. При атестації програмного забезпечення проводять його тестування з метою визначення однієї або декількох характеристик у відповідності з методикою атестації (аналіз документації, функціональні перевірки програми при контрольованих умовах, аналіз вихідного коду тощо) [41 - 44, 54-58].

Розрізняють два види атестації: загальну і метрологічну [45]. Об'єктом загальної атестації є повне програмне забезпечення ЗВ, дослідження якого проводяться для обґрунтування застосування алгоритму (програми) в конкретних завданнях. Метрологічна атестація досліджує ПЗ, яке є складовою частиною програмного забезпечення конкретних вимірювальних пристроїв, вимірювально-обчислювальних комплексів, вимірювальних систем з метою оцінки впливу ПЗ на похибку результатів вимірювань.

Сертифікація ПЗ ЗВ – це підтвердження відповідності програмного забезпечення та апаратно-програмних комплексів вимогам технічних регламентів та стандартів [1, 41, 69, 73].

Проведення атестації та випробувань ПЗ при затвердженні типу ЗВ регламентується переважно вимогами законодавчих і нормативних документів, а також низкою спеціальних документів на засоби вимірювань, що застосовуються в різних областях. Сертифікація ПЗ ЗВ є переважно добровільною за винятком ЗВ, що виконують особливо відповідальні функції, в яких недостатня якість, помилки або відмови можуть завдати великої шкоди, або є небезпечними для життя і здоров'я людей (авіація, атомна енергетика, системи управління органами влади, банківські системи тощо) [46]. Добровільна сертифікація визначається в основному бажанням розробників (користувачів) ПЗ ЗВ підтвердити його відповідність сучасним вимогам, підтвердити конкурентоспроможність, отримати додаткові економічні переваги тощо.

Отже, дані процедури охоплюють перевірку всього ПЗ, яке є складовою частиною певного ЗВ. У залежності від типу ЗВ та нормативних документів підбирається процедура перевірки ПЗ ЗВ. Дані процедури є важливими не тільки з законодавчого боку. Якщо ПЗ ЗВ пройшло перевірку і отримало документ про підтвердження якості і відповідності ПЗ документації, то це також суттєва перевага, яка підтверджує конкурентоспроможність та відповідність ЗВ сучасним вимогам.

Проте, жодна із розглянутих процедур не в достатній мірі регламентує методику метрологічної перевірки програмного забезпечення засобів вимірювання, яке використовується для обчислення та опрацювання результатів вимірювання. Крім того, не запропоновано уніфікованого підходу до метрологічної перевірки ПЗ ЗВ.

1.4 Методи перевірки програмного забезпечення засобів вимірювання

Регіональна Організація співробітництва державних метрологічних установ країн Центральної та Східної Європи опублікувала Рекомендацію КОOMET «Програмне забезпечення засобів вимірювань. Загальні технічні вимоги» [47]. Європейська комісія опублікувала директиву для засобів вимірювань. Директива 2004/22/ЕС містить основні вимоги до засобів вимірювань [48]. Європейська кооперація із законодавчої метрології WELMЕС розробила Керівництво з програмного забезпечення [14]. У звіті [49] описано методи перевірки (аналізу та випробувань) ПЗ, що застосовується до різних типів ЗВ.

При перевірці ПЗ ЗВ важливо підібрати правильний метод перевірки. При виборі методу береться до уваги насамперед можливість його реалізації, затрати на реалізацію, якість проведеної перевірки тощо [49]. Основними

методами перевірки програмного забезпечення засобів вимірювання є [1, 49, 65, 68, 82, 83, 86]:

- аналіз документації та специфікації ПЗ ЗВ;
- застосування опорного ПЗ;
- використання моделей вихідних даних;
- генерування «еталонних» даних;
- випробування модулів ПЗ;
- порівняння програм;
- аналіз вихідного коду ПЗ;
- метод «білої скриньки»;
- метод «чорної скриньки».

Аналіз документації та специфікації ПЗ ЗВ проводиться на основі документації виробника на ЗВ. В залежності від потреб ця документація повинна мати відповідний зміст:

- технічний опис функцій зовнішнього доступу до приладу в загальному вигляді;
- технічний опис функцій і інтерфейсів ПЗ повинен виявити і пояснити всі функції ПЗ, які можуть впливати на метрологічні властивості ЗВ;
- документація повинна включати повний список команд або сигналів, які ПЗ здатне інтерпретувати. Має бути описано, як ЗВ реагує на неописані команди;
- повинна бути надана додаткова документація на ПЗ для складних вимірювальних алгоритмів, криптографічних функцій або критичних тимчасових обмежень, якщо це необхідно для розуміння і оцінювання функцій програмного забезпечення;
- коли неясно, як атестувати функції ПЗ, відповідальність за доопрацювання методу тестування має покладатися на виробника. Крім того, експерту повинні бути надані послуги програміста з метою відповідей на питання.

Експерт оцінює функції та властивості ЗВ, використовуючи опис ПЗ, і приймає рішення про їх відповідність вимогам. Даний метод проводить оцінку тільки на основі документації, не враховуючи експериментальної перевірки ПЗ ЗВ.

Переваги методу:

- забезпечує детальний аналіз документації ПЗ ЗВ, на основі якого можна оцінити відповідність програмного забезпечення вимірювальній задачі приладу.

Недоліки методу:

- відсутність експериментальної перевірки ПЗ та оцінки похибок і/або непевностей ПЗ ЗВ;
- результати перевірки в певній мірі є суб'єктивними, тому що висновок робить експерт.

Застосування опорного ПЗ використовується при наявності ПЗ, яке допомагає ідентифікувати функції ПЗ, що перевіряється. В якості опорного ПЗ може бути застосоване:

- атестоване або/і сертифіковане ПЗ ЗВ, функціональне призначення якого аналогічне ПЗ, що перевіряється;
- спеціально розроблене ПЗ з функціями, ідентичними ПЗ, що перевіряється;
- ПЗ для вирішення завдань технічних обчислень (наприклад, електронні таблиці, ПЗ для математичних та статистичних обчислень тощо).

До розробки опорного ПЗ вдаються у тих випадках, коли ПЗ, що перевіряється є не складним і реалізація його алгоритмів є простою. Опорне ПЗ не повинно відтворювати усі функціональні можливості ПЗ, яке перевіряється, а може містити тільки функції і параметри, що впливають на метрологічні характеристики ЗВ. У деяких випадках не враховуються особливості графічного інтерфейсу користувача, а також функції, що не

беруть участь в опрацюванні результатів вимірювань (наприклад, функції відображення, зберігання даних тощо).

Переваги методу:

- надає можливість максимально враховувати особливості ПЗ, що перевіряється, та метрологічні характеристики ЗВ.

Недоліки методу:

- якщо немає ПЗ, функціональне призначення якого аналогічне ПЗ, що перевіряється, то необхідним є створення опорного ПЗ, що суттєво збільшує час перевірки ПЗ;
- часто складність ПЗ робить реалізацію даного методу недоцільною через значні затрати на розробку опорного ПЗ.

Метод використання моделей початкових даних рекомендований для атестації алгоритмів опрацювання результатів вимірювань. Метод дозволяє оцінювати можливості алгоритмів, проводячи порівняння результатів опрацювання моделей початкових даних алгоритмами, які тестуються, із заданими параметрами цих моделей. Метод моделей початкових даних є різновидом методу генерування «еталонних» даних, тільки дані не генеруються спеціально розробленими програмами, а програмно задаються на вході ПЗ, яке тестується. На вхід ПЗ подають вхідні дані, які відповідають вимірюваним даним. Вихідні дані порівнюють із даними, описаними в специфікації ЗВ. Відхилення одержаних даних від «зразкових», тобто описаних в специфікації, характеризує стан ПЗ ЗВ.

Моделі початкових даних вибираються таким чином, щоб вони максимально відповідали вимірювальному завданню. При цьому моделі початкових даних повинні охоплювати якомога більший діапазон можливих значень. До моделей початкових даних можуть бути включені:

- дані, які повністю перекривають діапазон можливих значень;
- дані, близькі до найбільших і найменших значень, а також ряд проміжних значень;

- дані, що виходять за межі діапазону можливих значень;
- особливі значення вхідних змінних – точки різкого зростання або розриву похідних, нульові, одиничні, гранично малі чисельні значення змінних тощо, а також комбінації зазначених вище даних.

Якщо значення деякої змінної залежать від значень іншої змінної, то тестування проводять для певних поєднань цих змінних, наприклад рівність обох змінних, найбільше і найменше значення, нульові значення змінних.

Переваги методу:

- є простішим в реалізації, ніж метод генерування «еталонних» даних.

Недоліки методу:

- для розроблення моделей початкових даних необхідна апріорна інформація про алгоритми роботи ПЗ та їх програмну реалізацію, а вона не завжди відома.

Метод генерування «еталонних» даних, як і метод моделей початкових даних, застосовується в разі відсутності або неможливості використання опорного ПЗ. Однією з необхідних умов застосування методу генерування «еталонних» даних є наявність апріорної інформації про відповідне вимірювальне завдання. «Еталонні» дані отримують шляхом генерування даних за допомогою спеціально розробленої програми – генератора «еталонних» даних на основі заданих початкових даних. Генератор «еталонних» даних реалізують на одній з мов програмування або за допомогою стандартного математичного або статистичного програмного пакету. Дані для тестування, в тому числі і для генерування «еталонних» даних, формуються з урахуванням властивостей алгоритмів ПЗ, які перевіряються. Метод генерування «еталонних» даних є альтернативою методу з використанням опорного ПЗ. З іншого боку, розробка генератора «еталонних» даних є доцільною, коли вона є дешевшою, ніж реалізація інших методів.

Переваги методу:

- є простішим у реалізації, порівняно з застосуванням опорного ПЗ.

Недоліки методу:

- неможливий в реалізації, якщо відсутня інформація про алгоритми роботи програмного забезпечення;
- є складнішим в реалізації, ніж метод початкових даних.

Метод випробування модулів програмного забезпечення доцільно застосовувати тільки тоді, якщо потрібний високий рівень відповідності ПЗ. Цей метод застосовується, коли функції програми не можна перевірити виключно на основі записаної інформації. Він доречний і економічно вигідний при перевірці алгоритмів динамічних вимірювань. Випробуваний програмний модуль ЗВ інтегрується у випробувальне оточення, тобто в спеціальний модуль тестової програми, який викликає його і забезпечує йому всі необхідні вхідні дані. Тестова програма отримує вихідні дані від випробуваного модуля і порівнює їх з очікуваними «еталонними» значеннями.

Переваги методу:

- забезпечує детальну перевірку певної обраної частини ПЗ.

Недоліки методу:

- для реалізації необхідні навички в сфері програмування та метрології;
- висока вартість реалізації.

Метод порівняння. За наявності декількох реалізацій програм одного і того ж вимірювального алгоритму та за відсутності опорного ПЗ доцільним є проводити перевірку програмного забезпечення методом порівняння (на входи програмних продуктів подаються однакові набори «еталонних» даних і проводиться порівняння відповідних тестових результатів на виході).

Переваги методу:

- простота реалізації;
- не потребує написання додаткових програм.

Недоліки методу:

- ЗВ, які мають однакові функції ПЗ, зустрічаються доволі рідко.

При тестуванні на основі аналізу вихідного коду ПЗ перевіряється:

- відповідність структури алгоритмів наданій документації;
- правильність запису алгоритмів обраною мовою програмування;
- відповідність вибраних алгоритмів вимірювальній задачі (виявлення нестійких алгоритмів).

За допомогою цього методу можна перевірити практично будь-яку характеристику ПЗ, якщо необхідний поглиблений рівень перевірки. Експерт послідовно крок за кроком аналізує відповідну частину вихідного коду, щоб визначити, чи виконані вимоги і чи відповідають документації програмні функції та властивості. При перевірці відповідності структури алгоритмів наданій документації за текстом програми можуть бути складені блок-схеми алгоритмів, які порівнюються з алгоритмами, описаними в документації. У разі знаходження відмінностей в структурі алгоритмів проводиться додатковий аналіз елементів блок-схем. При перевірці правильності запису алгоритмів обраною мовою програмування встановлюється відповідність коду правилам програмування, наявність невизначених змінних і операторів, правильність організації циклів тощо. Відповідність вибраних алгоритмів вимірювальному завданню може бути оцінена шляхом математичного аналізу програмно реалізованих алгоритмів. При цьому можуть досліджуватися різні характеристики реалізованих алгоритмів, зокрема, може проводитися аналіз оптимальності чисельних методів розв'язання вимірювального завдання.

Переваги методу:

- забезпечує проведення детальної оцінки ПЗ ЗВ.

Недоліки методу:

- необхідно задіяти спеціалістів з галузі програмування та метрології;
- витрати на реалізацію є набагато вищими у порівнянні з іншими методами;

- результати перевірки в певній мірі є суб'єктивними, тому що висновок робить експерт.

Тестування ПЗ [86] використовується для того, щоб продемонструвати розробникам і замовникам, що програма відповідає вимогам, а також для виявлення помилок у роботі ПЗ. Можна виділити два основні методи:

- метод «білої скриньки»;
- метод «чорної скриньки».

Метод «білої скриньки» виконується з метою виявлення проблем у внутрішній структурі програми. Це вимагає глибокого знання внутрішньої структури, тому не може бути виконано звичайним користувачем. Загальна задача такого тестування – забезпечити перевірку кожного кроку за алгоритмом програми. У процесі застосування методу «білої скриньки» тестувальник не тільки має доступ до коду, а й сам може писати код ПЗ. Як правило, проводиться тестування окремих частин програми. Цей метод забезпечує перевірку стійкості і працездатності системи.

Переваги методу:

- береться до уваги структура всієї програми, що полегшує виявлення помилок навіть у тому випадку, коли специфікації програмного забезпечення неповні;
- забезпечує перевірку стійкості і працездатності системи.

Недоліки методу:

- є громіздким і вимагає значних розумових та фінансових затрат;
- результати перевірки в певній мірі є суб'єктивними, тому що висновок робить тестувальник.

Метод «чорної скриньки» проводиться без знання внутрішньої структури та алгоритмів роботи ПЗ. Під «чорною скринькою» розуміється об'єкт дослідження, внутрішній устрій якого невідомий. Метою тестування «чорної скриньки» є з'ясування обставин, в яких поведінка програми не

відповідає специфікації. Для виявлення всіх помилок у програмі необхідно виконати вичерпне тестування, тобто тестування «всеохоплюючих» наборів даних. Для більшості програм таке неможливо, тому тестування програми обмежується невеликими підмножинами різних наборів даних. При цьому вибираються найбільш підходящі підмножини з найвищою імовірністю виявлення помилок. Даний метод є простіший в реалізації, порівняно з методом «білої скриньки». Основне завдання тестувальника полягає у перевірці відповідності ПЗ вимогам. Також необхідно перевірити ПЗ у критичних ситуаціях (стрес тестування), при навантаженнях, які значно перевищують очікувані, з метою визначити витривалість та стійкість ПЗ.

Переваги методу:

- простота реалізації;
- невисока вартість реалізації.

Недоліки методу:

- складність створення підмножин наборів даних, які дозволяють з найвищою імовірністю виявити помилки ПЗ.

На рисунку 1.9 представлено зв'язок між типом програмного забезпечення засобів вимірювання та методами перевірки програмного забезпечення засобів вимірювання.

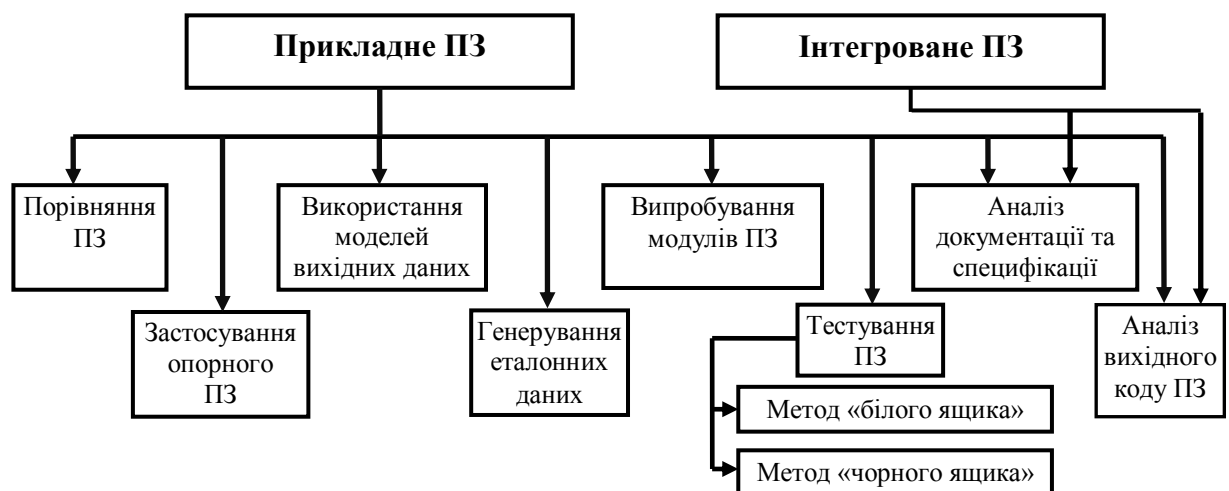


Рисунок 1.9 – Зв'язок між типом ПЗ ЗВ та методами перевірки ПЗ ЗВ

У таблиці 1.1 систематизовано інформацію щодо переваг та недоліків методів перевірки програмного забезпечення засобів вимірювання.

Таблиця 1.1 – Порівняльна характеристика методів перевірки програмного забезпечення засобів вимірювання

Метод перевірки	Переваги	Недоліки
1	2	3
Аналіз документації та специфікації ПЗ ЗВ	<ul style="list-style-type: none"> забезпечує детальний аналіз документації ПЗ ЗВ, на основі якого можна оцінити відповідність ПЗ вимірювальній задачі приладу 	<ul style="list-style-type: none"> відсутність експериментальної перевірки ПЗ та оцінки похибок і/або непевностей ПЗ ЗВ; результати перевірки в певній мірі є суб'єктивними, тому що висновок робить експерт
Застосування опорного ПЗ	<ul style="list-style-type: none"> надає можливість максимально враховувати особливості ПЗ, що перевіряється, та метрологічні характеристики ЗВ 	<ul style="list-style-type: none"> якщо немає ПЗ, функціональне призначення якого аналогічне ПЗ, що перевіряється, то необхідним є створення опорного ПЗ, що суттєво збільшує час перевірки ПЗ; часто складність ПЗ робить реалізацію даного методу недоцільною
Використання моделей початкових даних	<ul style="list-style-type: none"> є простішим в реалізації, ніж метод генерування «еталонних» даних 	<ul style="list-style-type: none"> для розроблення моделей початкових даних необхідна апріорна інформація про алгоритми роботи ПЗ та їх програмну реалізацію, а вона не завжди відома
Генерування «еталонних» даних	<ul style="list-style-type: none"> є простішим у реалізації, порівняно з застосуванням опорного ПЗ 	<ul style="list-style-type: none"> неможливий в реалізації, якщо відсутня інформація про алгоритми роботи програмного забезпечення; є складнішим в реалізації, ніж метод початкових даних

Продовження таблиці 1.1

1	2	3
Випробування модулів ПЗ	<ul style="list-style-type: none"> • забезпечує детальну перевірку певної обраної частини ПЗ 	<ul style="list-style-type: none"> • для реалізації необхідні навички в сфері програмування та метрології; • висока вартість реалізації
Порівняння ПЗ	<ul style="list-style-type: none"> • простота реалізації; • не потребує написання додаткових програм 	<ul style="list-style-type: none"> • ЗВ, які мають однакові функції ПЗ, зустрічаються доволі рідко
Аналіз вихідного коду ПЗ	<ul style="list-style-type: none"> • забезпечує проведення детальної оцінки ПЗ ЗВ 	<ul style="list-style-type: none"> • витрати на реалізацію є набагато вищими у порівнянні з іншими методами; • результати перевірки в певній мірі є суб'єктивними, тому що висновок робить експерт
Тестування ПЗ за методом «білої скриньки»	<ul style="list-style-type: none"> • береться до уваги структура всієї програми, що полегшує виявлення помилок навіть у тому випадку, коли специфікації програмного забезпечення неповні; • забезпечує перевірку стійкості і працездатності системи 	<ul style="list-style-type: none"> • є громіздким і вимагає значних розумових та фінансових затрат; • результати перевірки в певній мірі є суб'єктивними, тому що висновок робить тестувальник
Тестування ПЗ за методом «чорної скриньки»	<ul style="list-style-type: none"> • простота реалізації; • невисока вартість реалізації 	<ul style="list-style-type: none"> • складність створення підмножин наборів даних, які дозволяють з найвищою імовірністю виявити помилки ПЗ

На основі аналізу переваг та недоліків вищерозглянутих методів можна зробити висновок, що для реалізації метрологічної перевірки ПЗ ЗВ

доцільним є використання двох методів: методу «чорної скриньки» та методу генерування «еталонних» даних. Поєднавши ці методи, запропоновано метод метрологічної перевірки програмного забезпечення, який не потребує знання внутрішньої структури програми. Для реалізації цього методу не потрібно створювати зразкове ПЗ, проводити аналіз алгоритму та структури ПЗ, не потрібно залучати великі фінансові затрати та спеціалістів у сферах метрології та програмування і т.д., що в свою чергу дозволяє суттєво спростити та зменшити тривалість метрологічної перевірки ПЗ ЗВ, а також уніфікувати процес метрологічної перевірки ПЗ ЗВ, яке приймає участь у розрахунку результату вимірювання.

Висновки до розділу 1

1. Аналіз існуючих процедур та методів перевірки програмного забезпечення показав, що вони не надають оцінки значення похибки програмного забезпечення, а також не забезпечують уніфікованого підходу до метрологічної перевірки програмного забезпечення.
2. Метрологічна перевірка програмного забезпечення засобів вимірювання переважно потрібна лише на етапі проектування та розробки. Метрологічна перевірка програмного забезпечення кіберфізичних систем є актуальною і під час експлуатації, оскільки такі системи можуть змінювати своє програмне забезпечення.
3. Для реалізації метрологічної перевірки програмного забезпечення засобів вимірювання доцільним є поєднання двох методів: методу «чорної скриньки» та методу генерування «еталонних» даних, що дозволить уніфікувати та спростити метрологічну перевірку програмного забезпечення засобів вимірювання.

РОЗДІЛ 2

ОСОБЛИВОСТІ МЕТРОЛОГІЧНОЇ ПЕРЕВІРКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАННЯ ТА КІБЕР-ФІЗИЧНИХ СИСТЕМ

У другому розділі запропоновано алгоритм метрологічної перевірки програмного забезпечення засобів вимірювання, розглянуто кібер-фізичні системи, розроблено алгоритми для реалізації метрологічної перевірки програмного забезпечення кібер-фізичних систем на різних рівнях функціонування.

2.1 Особливості метрологічної перевірки програмного забезпечення засобів вимірювання

Метрологічна перевірка ПЗ ЗВ може бути реалізована тільки при наявності доступу до ПЗ. У іншому випадку можливо провести тільки метрологічну перевірку ЗВ в цілому. Для автономних ЗВ метрологічна перевірка ПЗ може бути реалізована через додатковий вхід (рисунок 1.2). Формат даних, які поступають на ПЗ має бути таким, як і дані з АЦП.

Для засобів вимірювання, програмна частина яких знаходиться на комп'ютері, передавання даних може бути реалізоване таким чином:

1. із використанням оперативної пам'яті (рисунок 2.1);
2. за допомогою файлової системи (рисунок 2.2).

У першому випадку можливі два способи використання даних для обчислення результатів вимірювання. Для першого способу дані з АЦП знаходяться в одному блоці пам'яті, а тестові дані – у іншому. У такому

випадку у робочому стані програма працює з першим блоком даних, при перевірці – з другим. Для другого способу всі дані розташовуються в одному блоці пам'яті. Під час роботи в цей блок надходять дані з АЦП, під час перевірки – тестові дані.



Рисунок 2.1 – Передавання цифрових тестових послідовностей сигналу на вхід ПЗ через оперативну пам'ять, де ОП – блок оперативної пам'яті

У другому випадку тестові послідовності (як і вимірювані сигнали) записуються у файл. Дані завантажуються програмою із файлу після чого відбувається подальше опрацювання отриманої інформації.

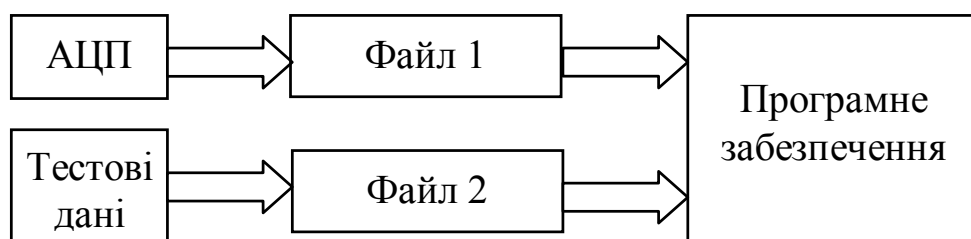


Рисунок 2.2 – Передавання цифрових тестових послідовностей сигналу на вхід ПЗ через файл

Будь-який із розглянутих вище способів передавання даних не впливає на алгоритм метрологічної перевірки ПЗ ЗВ.

На рисунку 2.3 представлено структурну схему, а на рисунку 2.4 загальний алгоритм метрологічної перевірки ПЗ ЗВ.



Рисунок 2.3 – Структурна схема метрологічної перевірки ПЗ ЗВ

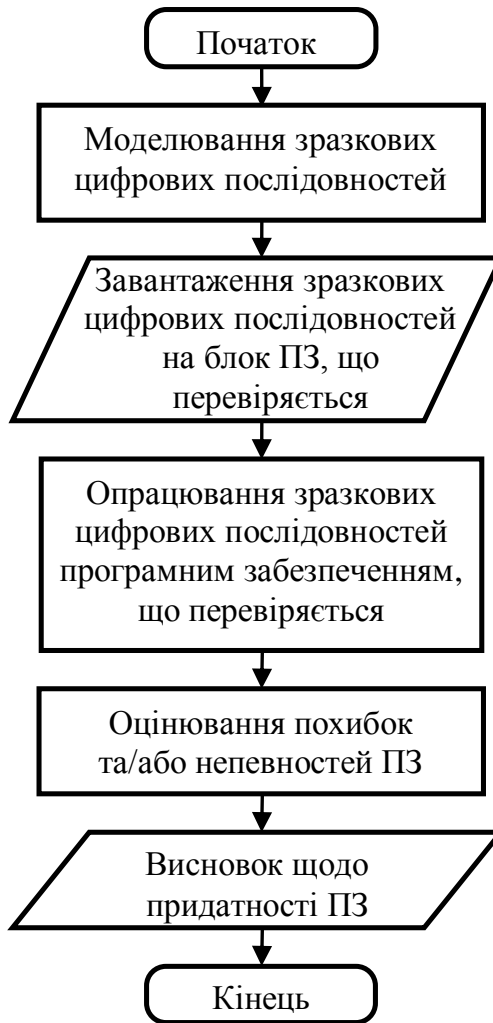


Рисунок 2.4 – Алгоритм метрологічної перевірки ПЗ ЗВ

Для метрологічної перевірки ПЗ необхідно провести формування зразкових цифрових послідовностей, які будуть подаватися на програмне забезпечення ЗВ. Ці цифрові послідовності повинні враховувати діапазон вхідних даних ПЗ. Даний метод забезпечує детальнішу перевірку ПЗ у порівнянні із стандартною метрологічною перевіркою ЗВ, оскільки аналоговими тестовими сигналами на вході ЗВ складно відтворити усі можливі варіанти вхідної величини для ПЗ. Крім того, при використанні зразкових цифрових послідовностей можна суттєво скоротити тривалість метрологічної перевірки ПЗ.

Процесом метрологічної перевірки ПЗ ЗВ керує комп'ютер (рисунок 2.3). Зразкові цифрові послідовності подаються на ПЗ ЗВ. Вихідні дані ПЗ передаються на комп'ютер, де порівнюються із очікуваними результатами. Проводиться обчислення похибок ПЗ і робиться висновок щодо придатності ПЗ ЗВ. Розглянутий вище алгоритм може бути застосований для перевірки ПЗ ЗВ як на етапі розроблення так і під час експлуатації.

2.2 Особливості метрологічної перевірки програмного забезпечення компонентів кібер-фізичних систем

КФС, як правило, є складними системи, компоненти яких можуть знаходитися на великій відстані одна від одної та з'єднані між собою із використанням глобальних мереж [20-22, 33-37]. Враховуючи це, є можливість проведення віддаленої метрологічної перевірки ПЗ КФС.

Інфраструктура КФС в основному складається з підсистем, електронні компоненти яких реалізуються через ВСК [23], що отримують інформацію про навколишнє середовище за допомогою інтелектуальних первинних

перетворювачів (ІПП) та ЗВ і можуть впливати на нього через виконавчі механізми (рисунок 2.5).



Рисунок 2.5 – Структурна схема підсистеми КФС

Інтелектуальні первинні перетворювачі включають в себе мікроконтролер з ПЗ, за рахунок чого можуть автоматично обирати діапазон вимірювання, проводити алгоритмічну корекцію результатів вимірювання, передавати результати вимірювання в цифровій формі тощо. ІПП можуть проводити самоналаштування, самодіагностику та самоперевірку [84, 85]. Така структура ІПП забезпечує набагато простішу метрологічну перевірку ІПП у порівнянні з типовими первинними перетворювачами та засобами вимірювання [4, 25, 86, 87]. Метрологічну перевірку ПЗ ІПП можна проводити за алгоритмом, який представлений на рисунку 2.4.

Якщо вимірювальна інформація відображає характеристики об'єкта зовнішнього середовища з похибкою, що перевищує допустимі значення, то дія виконавчих механізмів на об'єкт може бути неправильною. Це може призвести до некоректного функціонування КФС та небажаних наслідків, а інколи і до небезпеки життю людини. Тому для забезпечення правильного функціонування компонентів КФС актуальним є розроблення методів, методик та засобів метрологічної перевірки програмного забезпечення.

Для віддаленої перевірки ПЗ кібер-фізичні системи підходять якнайкраще, оскільки вони з'єднані з Інтернетом і доступ до них, а отже і до ПЗ, що їх обслуговує, є наявним. Потрібно лише забезпечити певний програмний сервіс, який дозволяє запустити процес перевірки.

Структури кібер-фізичних систем. Організація структури КФС може відбуватися двома способами [25, 87]:

- 1) всі підсистеми КФС є рівноправними, здатні самостійно формувати задачі і разом з іншими компонентами брати участь у їх вирішенні, проводять самоперевірку, самодіагностику, самоналаштування, тощо (рисунок 2.6).
- 2) у КФС є основна підсистема, яка здійснює контроль та перевірку всіх інших підсистем, засобів вимірювання та первинних перетворювачів (рисунок 2.7).

При організації роботи **КФС з рівноправними підсистемами** одна із підсистем, залежно від поставленого завдання, бере на себе функцію головної (стає керівником і виконавцем одночасно). Переважно це підсистема, до якої звернувся користувач. Всі інші підсистеми у цей час є виконавцями. Такі системи є складними, оскільки кожна із підсистем може бути головною, тобто в усіх підсистем є функції і виконавця і керівника, що ускладнює їх програмну частину. У КФС з рівноправними підсистемами звітність може йти до кожної з підсистем. Користувач може здійснювати управління КФС та отримувати всю інформацію про неї, приєднавшись до

будь-якої з підсистем КФС. Перевагою КФС такого типу є її універсальність. Недоліком КФС з рівноправними компонентами є складність ПЗ та, відповідно, високі витрати на розробку такого ПЗ.

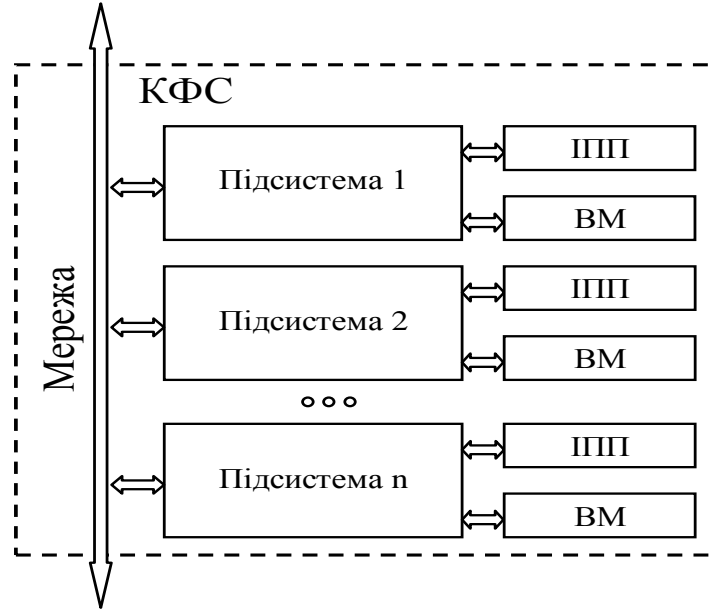


Рисунок 2.6 – КФС з рівноправними підсистемами, де ІПП – інтелектуальні первинні перетворювачі, ВМ – виконавчі механізми

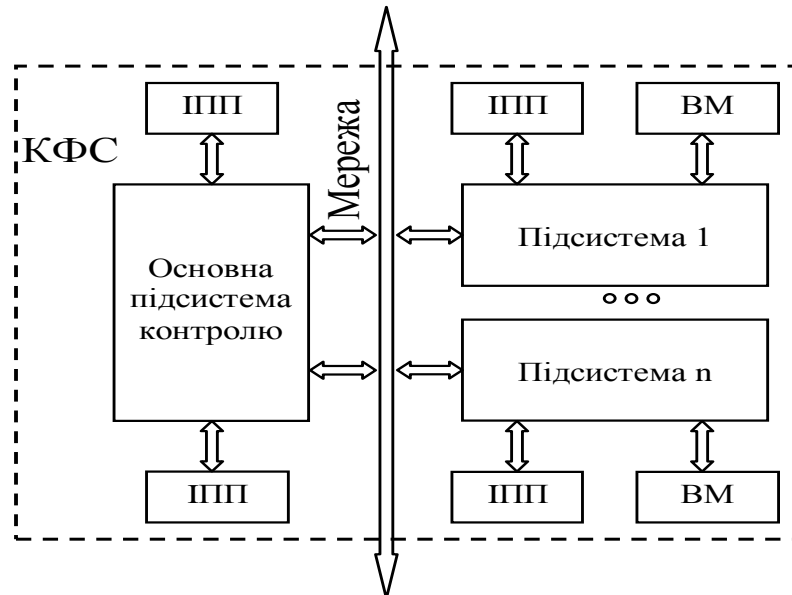


Рисунок 2.7 – КФС з основною підсистемою контролю, де ІПП – інтелектуальні первинні перетворювачі, ВМ – виконавчі механізми

При організації роботи **КФС з основною підсистемою контролю** тільки одна із підсистем має функції керівника та контролює роботу системи в цілому. Вона є основною підсистемою і проводить розподіл та організацію роботи між іншими підсистемами. Всі інші підсистеми є лише виконавцями. Контакт людини і КФС відбувається через підсистему контролю.

Перевагою КФС з основною підсистемою контролю є простіша реалізація ПЗ підсистем-виконавців у порівнянні із компонентами КФС з рівноправними підсистемами. Недоліком такої організації роботи є те, що підсистеми-виконавці не можуть самостійно розпочати роботу над новим проектом, що суттєво обмежує можливості таких КФС.

У обох варіантах реалізації КФС має бути врахована можливість ініціювання перевірки ПЗ усіх компонентів КФС людиною.

Багаторівнева метрологічна перевірка програмного забезпечення кібер-фізичних систем. Метрологічна перевірка ПЗ КФС є достатньо складною процедурою, тому що КФС переважно складається з великої кількості підсистем. Враховуючи таку специфіку КФС, як «засобу вимірювання» запропоновано багаторівневу метрологічну перевірку ПЗ КФС. Метрологічна перевірка ПЗ КФС має відбуватися на всіх етапах функціонування, розпочинаючи з самоперевірки ПЗ ІПП та ЗВ і закінчуючи загальною метрологічною перевіркою ПЗ КФС. Метрологічна перевірка може бути ініційована основною підсистемою контролю або обслуговуючим персоналом КФС, якщо виникли сумніви щодо коректності функціонування КФС або прийшов час планової перевірки. Ієрархія ініціювання метрологічної перевірки ПЗ КФС представлена на рисунку 2.8. Пріоритет ініціювання метрологічної перевірки ПЗ КФС має бути таким [3, 4, 25]:

- найвищий рівень: людина;
- другий рівень: основна підсистема контролю;
- третій рівень: підсистеми;
- четвертий рівень: ІПП та ЗВ.

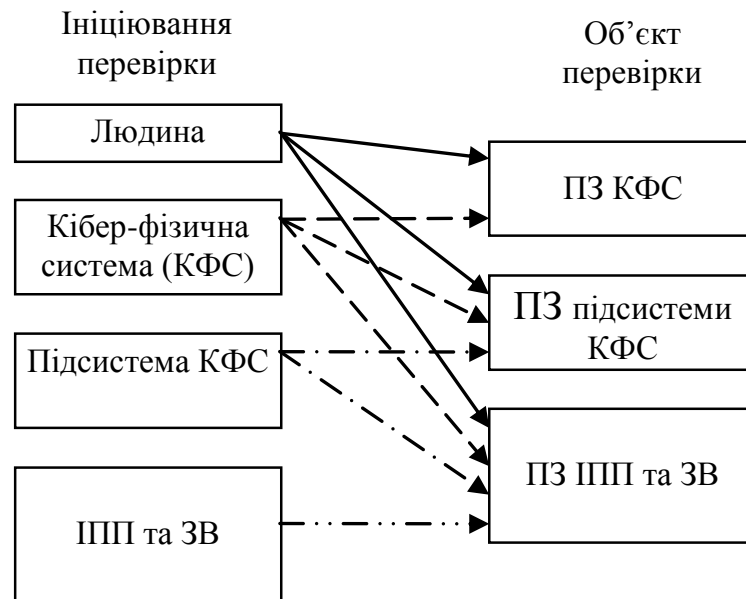


Рисунок 2.8 – Ієрархія ініціювання метрологічної перевірки КФС

Алгоритми метрологічної перевірки програмного забезпечення кібер-фізичних систем. Метрологічна перевірка ПЗ КФС може відбуватися на рівні системи, підсистеми та ІПП або ЗВ [3, 4, 25].

На рисунку 2.9 представлено алгоритм перевірки ПЗ КФС на рівні системи. Ініціювання метрологічної перевірки ПЗ здійснюватиме основна підсистема контролю. Якщо в цей час триває метрологічна перевірка ПЗ підсистеми, або ІПП чи ЗВ, то вона буде призупинена. Якщо при метрологічній перевірці ПЗ системи похибки не перевищують допустимих значень, то система проводитиме і далі роботу в звичайному режимі до наступної перевірки. Якщо в результаті перевірки виявлено похибки ПЗ, які перевищують допустимі значення, або збої у ПЗ системи, то система повинна замінити непрацездатну частину КФС, провести метрологічну перевірку ПЗ цієї частини та продовжити роботу в штатному режимі. У іншому випадку, коли системі самостійно не вдасться відновити роботу КФС, інформація про це буде передана до інтерфейсу користувача, КФС буде переведена в

безпечний режим роботи і відновленням роботи КФС займається обслуговуючий персонал.

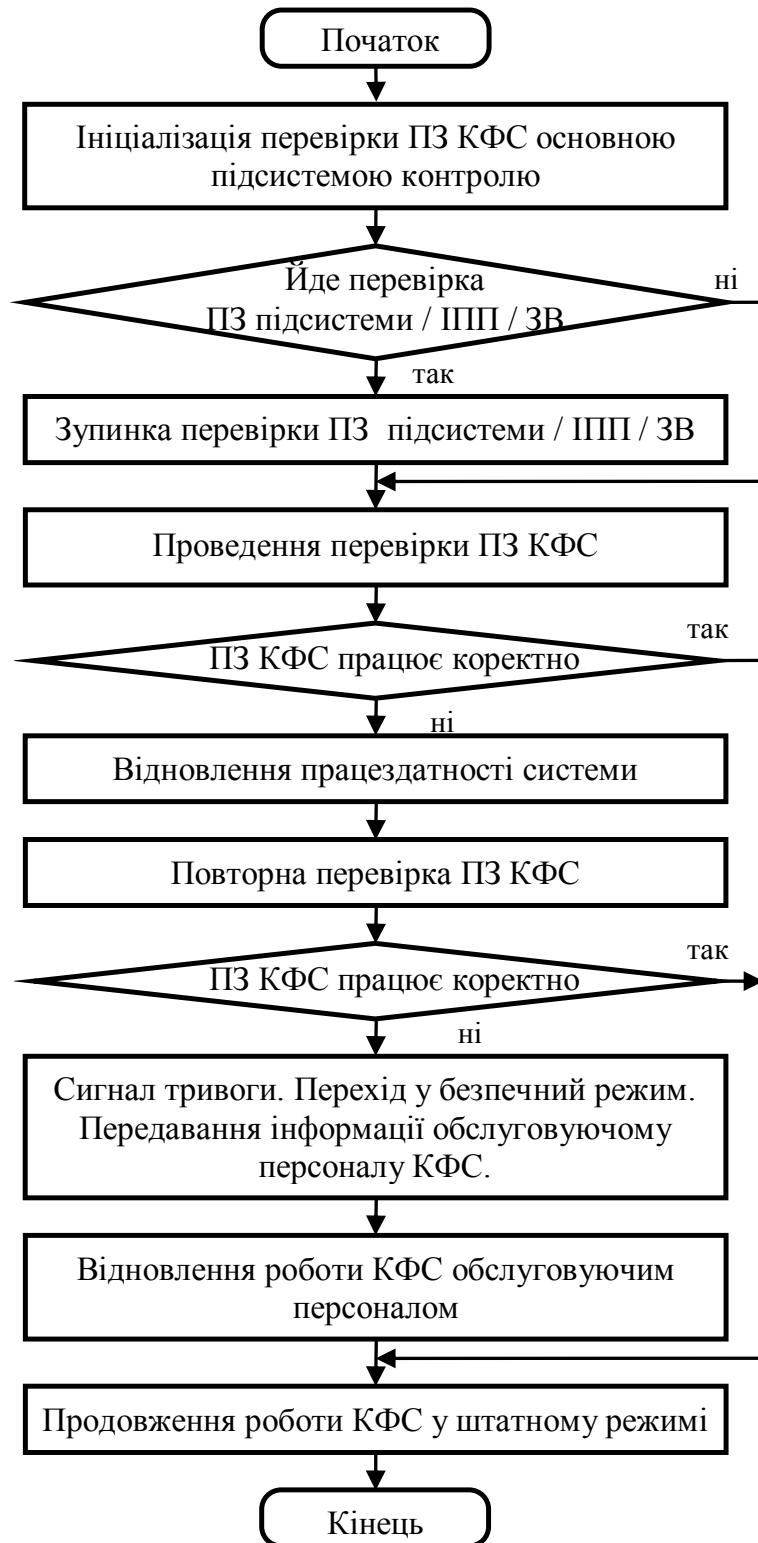


Рисунок 2.9 – Алгоритм перевірки ПЗ КФС

На рисунку 2.10 представлено алгоритм метрологічної перевірки ПЗ підсистеми КФС. Метрологічна перевірка ПЗ підсистеми може ініціюватися як основною підсистемою контролю так і самою підсистемою. Якщо в цей час відбувається метрологічна перевірка ПЗ ІПП або ЗВ, то вона буде призупинена. Якщо під час ініціалізації метрологічної перевірки ПЗ основною підсистемою вже відбувається метрологічна перевірка, то достатньо дочекатися її завершення без проведення додаткової перевірки. Подальший алгоритм роботи аналогічний перевірці ПЗ на системному рівні.

На рисунку 2.11 наведено алгоритм метрологічної перевірки ПЗ ІПП та ЗВ у КФС. Метрологічна перевірка ПЗ ІПП та ЗВ може ініціюватися основною підсистемою контролю, підсистемою, ІПП або ЗВ.

Якщо вже відбувається метрологічна перевірка ПЗ ІПП або ЗВ, то достатньо дочекатися її завершення без проведення повторної перевірки. Подальший алгоритм метрологічної перевірки аналогічний метрологічній перевірці ПЗ на системному та підсистемному рівнях.

Запропоновані алгоритми описують можливий варіант метрологічної перевірки ПЗ КФС на різних рівнях. В ідеалі людський фактор буде виключений із процесу метрологічної перевірки ПЗ КФС. Людина братиме участь у метрологічній перевірці ПЗ КФС тільки тоді, коли буде виявлено збої в програмі, з якими система не зможе справитися самостійно.

Розроблені алгоритми метрологічної перевірки ПЗ ІПП, ЗВ та КФС на різних рівнях є достатньо подібними. Тому доцільним є використання тестових цифрових сигналів для метрологічної перевірки їх ПЗ.

Враховуючи це, необхідно розробити універсальний засіб формування тестових цифрових послідовностей, який би дозволив реалізувати запропоновану метрологічну перевірку ПЗ ІПП, ЗВ та КФС.

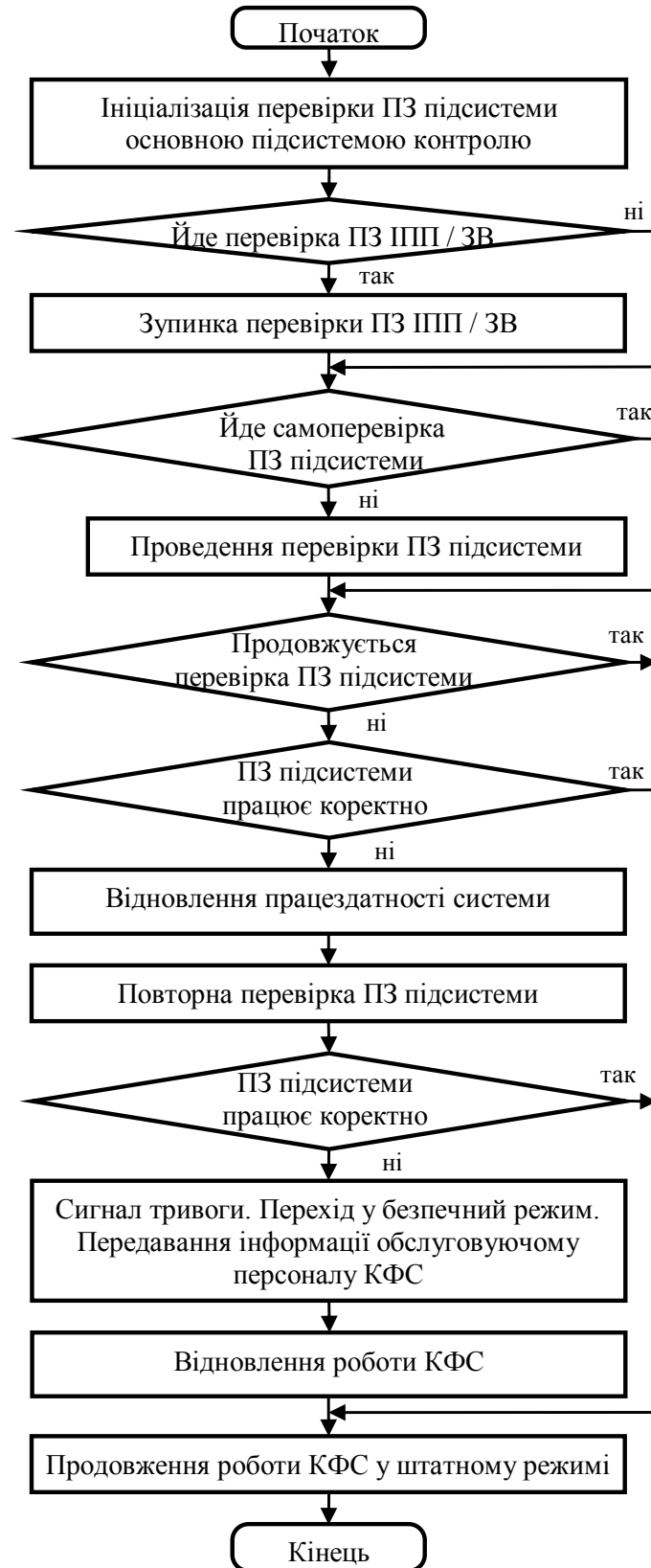


Рисунок 2.10 – Алгоритм перевірки ПЗ підсистеми

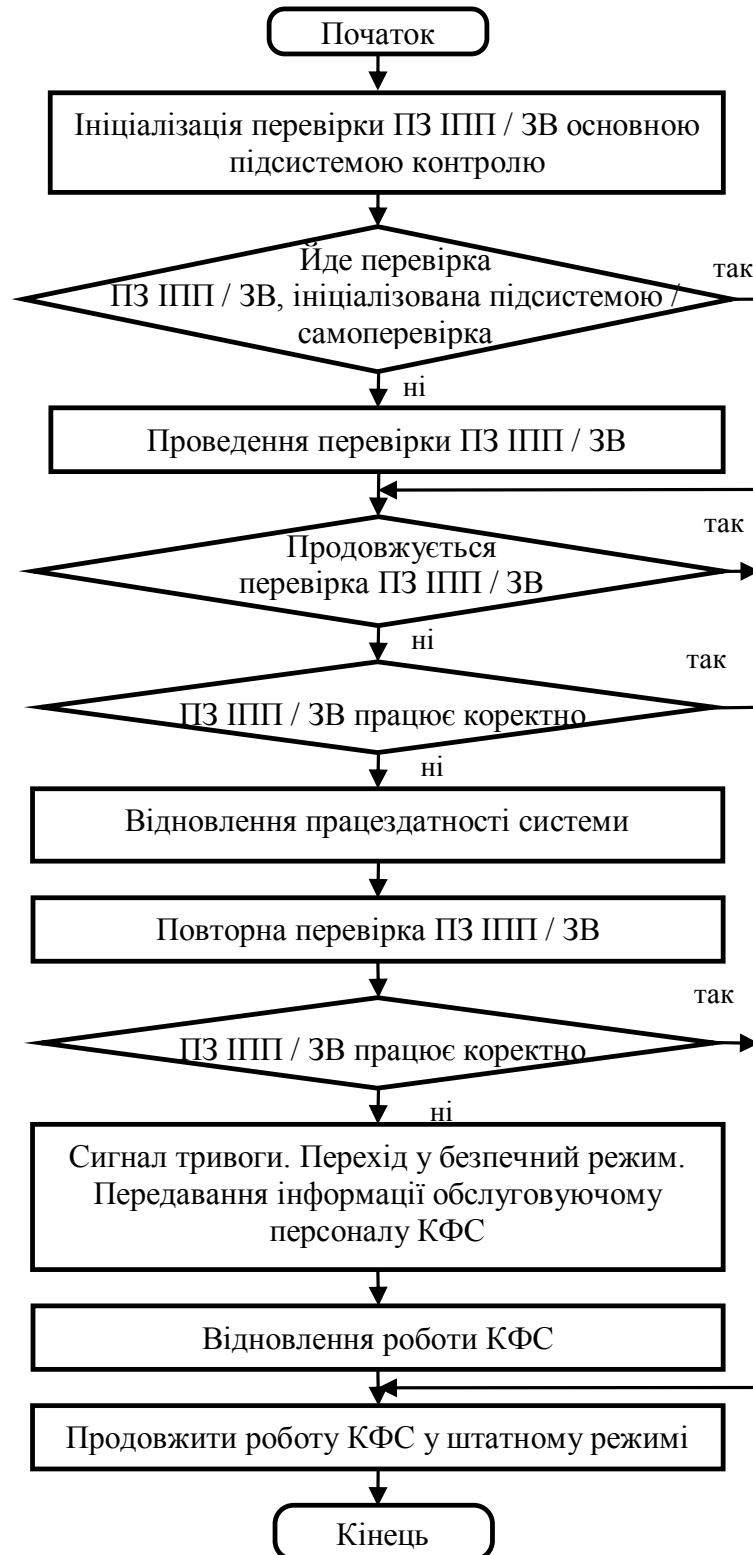


Рисунок 2.11 – Алгоритм перевірки ПЗ ІПП та ЗВ у КФС

Висновки до розділу 2

1. Розроблено алгоритми метрологічної перевірки програмного забезпечення інтелектуальних первинних перетворювачів, засобів вимірювання та кібер-фізичних систем.
2. Запропоновано багаторівневу метрологічну перевірку ПЗ компонентів кібер-фізичних систем, що дозволяє покращити безпеку функціонування кібер-фізичних систем та полегшити процес метрологічної перевірки їх ПЗ.
3. Для реалізації розроблених алгоритмів метрологічної перевірки програмного забезпечення засобів вимірювань необхідно розробити генератор зразкових цифрових послідовностей.

РОЗДІЛ 3

ГЕНЕРАТОР ЗРАЗКОВИХ ЦИФРОВИХ ПОСЛІДОВНОСТЕЙ

У третьому розділі описано генератор зразкових цифрових послідовностей, що призначений для формування зразкових цифрових послідовностей, які використовуються для метрологічної перевірки ПЗ ЗВ. Крім того, генератор зразкових цифрових послідовностей може використовуватись для відлагоджування ПЗ ЗВ на етапі проектування та розроблення.

Метрологічна перевірка ПЗ ЗВ реалізується за допомогою зразкових цифрових послідовностей, які відтворюють цифровий сигнал аналогової частини (наприклад від АЦП), що надходить на ПЗ ЗВ.

Реалізовано два режими роботи генератора зразкових цифрових послідовностей:

- метрологічна перевірка;
- відлагоджування програми.

Якщо генератор зразкових цифрових послідовностей використовується для метрологічної перевірки ПЗ ЗВ, то при формуванні зразкових цифрових послідовностей враховуються параметри вимірюваного сигналу:

- форма сигналу;
- частота сигналу,
- амплітуда сигналу;
- фаза сигналу;

та параметри АЦП [88-90]:

- розрядність;
- частота дискретизації;
- діапазон вхідної напруги,

а також час вимірювання.

Якщо генератор зразкових цифрових послідовностей використовується для відлагоджування ПЗ ЗВ, то є можливість враховувати також похибки АЦП:

- похибка зміщення (адитивна складова);
- похибка перетворення (мультиплікативна складова);
- інтегральна та диференціальна нелінійність (нелінійна складова);
- відношення сигнал/шум (випадкова складова);

та похибки аналогової частини.

3.1 Математична модель генератора зразкових цифрових послідовностей

На рисунку 3.1 представлено спрощений алгоритм роботи генератора зразкових цифрових послідовностей. У першу чергу проводиться введення усіх необхідних даних вимірювального сигналу та АЦП. Наступним кроком є розрахунок параметрів, необхідних для формування зразкових цифрових послідовностей. Під час формування зразкових цифрових послідовностей враховується:

- форма сигналу;
- похибки аналогової частини ЗВ;
- дискретизація за часом;
- квантування за рівнем.

Якщо рівень вхідної напруги сигналу перевищує максимальне значення діапазону вхідних значень АЦП, то можливі два способи формування сигналу:

- обмеження сигналу за рівнем;
- масштабування сигналу.



Рисунок 3.1 – Спрощений алгоритм роботи генератора зразкових цифрових послідовностей

Під час роботи генератора зразкових цифрових послідовностей в режимі метрологічної перевірки формується лише одна зразкова цифрова послідовність, яка в подальшому використовується для метрологічної перевірки ПЗ ЗВ.

Під час роботи генератора зразкових цифрових послідовностей в режимі відлагоджування програми можливим є формування n зразкових цифрових послідовностей із врахуванням похибок аналогової частини ЗВ. Для кожної послідовності значення похибок змінюються випадковим чином у певному заданому діапазоні. Це дозволяє провести дослідження впливу

похибок аналогової частини ЗВ на результат вимірювання, розрахований із використанням ПЗ, методом Монте-Карло.

Введення необхідних даних вимірювального сигналу та АЦП. При виконанні цього блоку алгоритму (рисунок 3.1) відбувається введення основних параметрів АЦП та вимірювального сигналу, таких як тактова частота АЦП f_{adc} , Гц; розрядність АЦП N_{adc} , бит.; час перетворення АЦП t , с; номінальна вхідна напруга АЦП U_{Nadc} , В; реальна вхідна напруга АЦП U_{Iadc} , В, що відповідає амплітуді сигналу A_s , кВ; частота сигналу f_s , Гц; фаза сигналу φ_s , град.

Розрахунок необхідних параметрів. Виходячи з часу перетворення t та тактової частоти АЦП f_{adc} розраховується загальна кількість даних зразкової цифрової послідовності:

$$n = f_{adc} \cdot t. \quad (3.1)$$

Коефіцієнт вхідної напруги розраховується за формулою:

$$k_{U_{vx}} = U_{Iadc} / U_{Nadc}. \quad (3.2)$$

Коефіцієнт вхідної напруги характеризує відношення напруги на вході АЦП до номінального значення напруги. Даний коефіцієнт використовується для формування цифрового квантового сигналу та для обмеження або масштабування сигналу за рівнем.

Крок дискретизації визначається за наступним виразом:

$$h_d = 1 / f_{adc}. \quad (3.3)$$

Амплітуда сигналу у квантах, враховуючи розрядність АЦП та рівень вхідної напруги обчислюється за формулою:

$$A_s = k_{U_{vx}} \cdot (n_{kv} / 2 - 1), \quad (3.4)$$

де $n_{kv} = 2^{N_{adc}}$ - кількість квантових рівнів за напругою.

Формування сигналу. У генераторі зразкових цифрових послідовностей реалізовано можливість формувати зразкові цифрові послідовності таких форм:

- синусоїдальної;
- типу меандр;
- трапецієподібної;
- трикутної;
- пилкоподібної.

При формуванні зразкових цифрових послідовностей враховуються характеристики та похибки АЦП. Крім того можна задати тип квантування до ближчого, більшого або меншого рівня.

Математична модель сигналу синусоїдальної форми. Формування зразкових цифрових послідовностей сигналу синусоїдальної форми відбувається в циклі за формулою:

$$Ys_i = A_s \cdot \sin(2 \cdot \pi \cdot f_s \cdot t_i + \phi_s), \quad (3.5)$$

де $i = 0 \dots n - 1$. Біжуче значення часу t_i розраховується за виразом:

$$t_i = h_d \cdot i. \quad (3.6)$$

Математична модель сигналу типу меандр. Формування зразкових цифрових послідовностей сигналу типу меандр відбувається із використанням даних Ys_i (3.5) за наступною формулою:

$$Ym_i = \begin{cases} A_s, & \text{якщо } Ys_i \geq 0 ; \\ -A_s, & \text{якщо } Ys_i < 0 . \end{cases} \quad (3.7)$$

Математична модель сигналу типу трапеція. Формування зразкових цифрових послідовностей трапецієподібної форми відбувається також із використанням даних Ys_i (3.5) за наступною формулою:

$$Ytr_i = \begin{cases} A_s \cdot k_{tr}, & \text{якщо } Ys_i > A_s \cdot k_{tr} ; \\ -A_s \cdot k_{tr}, & \text{якщо } Ys_i < -A_s \cdot k_{tr} ; \\ Ys_i, & \text{якщо } -A_s \cdot k_{tr} \leq Ys_i \leq A_s \cdot k_{tr} , \end{cases} \quad (3.8)$$

де $k_{tr} = 0,1...0,4$ – коефіцієнт трапеції, $A_s \cdot k_{tr}$ – рівень обмеження зразкових цифрових послідовностей синусоїдальної форми.

Для приведення зразкових цифрових послідовностей трапецієподібної форми до граничних значень АЦП необхідно врахувати цей коефіцієнт:

$$Ytr_i = Ytr_i / k_{tr} . \quad (3.9)$$

Математична модель сигналу трикутної форми. Розроблено алгоритм формування зразкових цифрових послідовностей сигналу трикутної форми.

На рисунку 3.2 представлено сигнал трикутної форми, частота якого $f_s = 1$ Гц, час вимірювання $t = 1$ с, амплітуда сигналу $A_s = 31$ кв. Кількість періодів зразкових цифрових послідовностей сигналу трикутної форми

залежить від частоти сигналу та часу вимірювання. Період сигналу розбивається на 4 рівні проміжки (рисунок 3.2), в кожному з яких формується пряма з координатами $(t_1, y_1; t_2, y_2)$, на першому проміжку пряма має координати $(0, 0; 1/4T, A_s)$, на другому проміжку – $(1/4T, A_s; 1/2T, 0)$, на третьому – $(1/2T, 0; 3/4T, -A_s)$, на четвертому – $(3/4T, -A_s; T, 0)$, де T – період сигналу, A_s – амплітуда сигналу. Тобто, кінцеві координати прямої є початковими координатами наступної прямої.

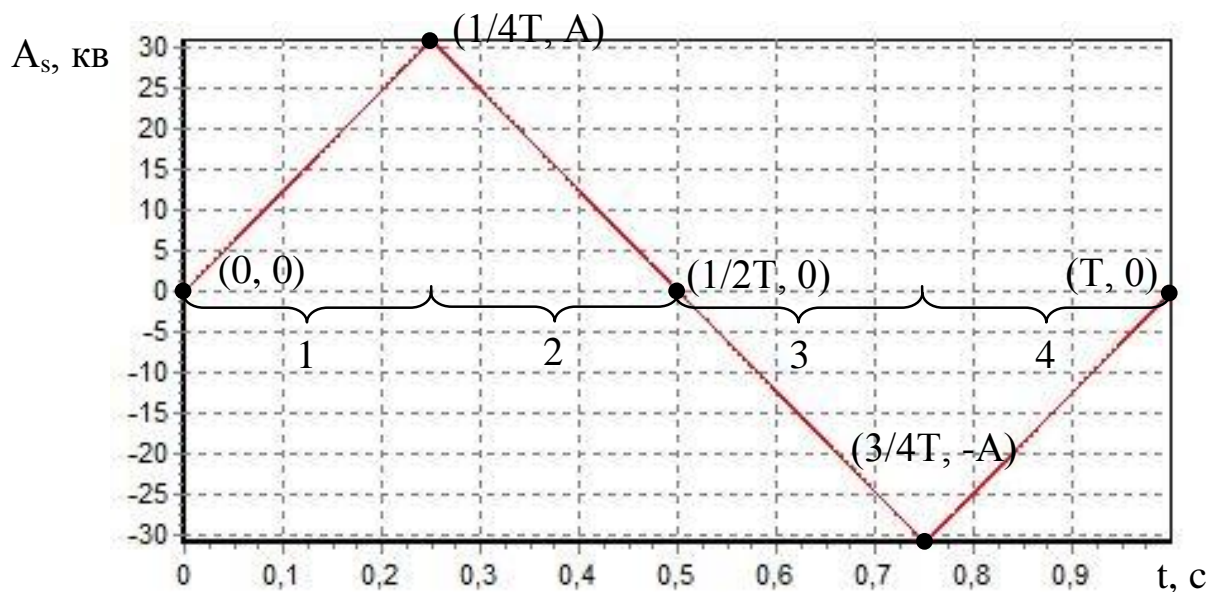


Рисунок 3.2 – Сигнал трикутної форми

Кожну із прямих можна описати рівнянням прямої [108]:

$$Y_i = \frac{(t_i - t_1) \cdot (y_2 - y_1)}{t_2 - t_1} + y_1, \quad (3.10)$$

де t_i – біжуче значення часу.

Кількість даних, які припадають на один період сигналу, визначається за формулою:

$$n_T = f_{adc} / f_s. \quad (3.11)$$

Кількість даних, які припадають на одну із прямих, що формують зразкові цифрові послідовності сигналу трикутної форми:

$$n_1 = n_T / 4. \quad (3.12)$$

На рисунку 3.3 подано графічне представлення кількості даних, які припадають на період та на $1/4$ періоду сигналу трикутної форми.

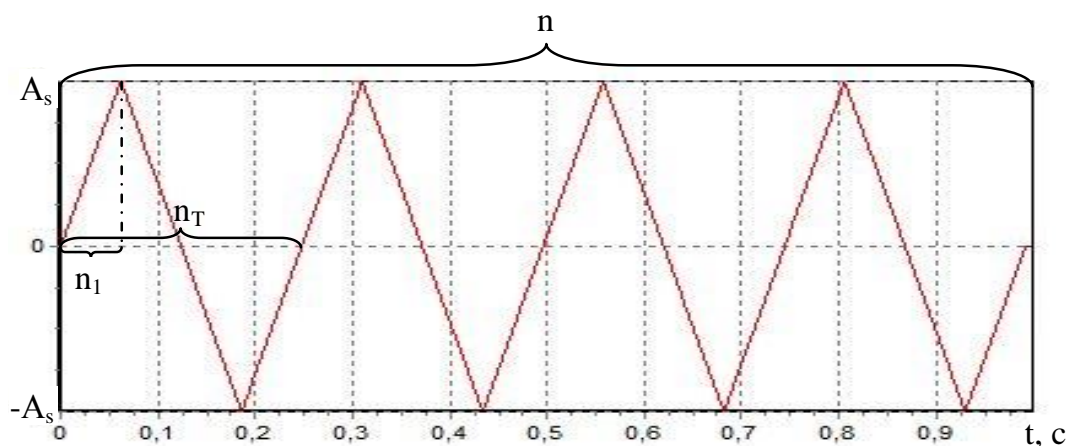


Рисунок 3.3 – Графічне представлення кількості даних, які припадають на період та на $1/4$ періоду сигналу трикутної форми

Отже, кількість координат, за якими формуються зразкові цифрові послідовності сигналу трикутної форми обчислюється для $m=0\dots n/n_1$ з кроком 1. Тоді координати прямих t_m та t_{m+1} , які відповідають координатам t_1 та t_2 виразу (3.10), розраховуються за формулами:

$$\begin{aligned} t_m &= n_1 \cdot m \cdot h_d ; \\ t_{m+1} &= n_1 \cdot (m+1) \cdot h_d . \end{aligned} \quad (3.13)$$

Координати Y_m та Y_{m+1} відповідають координатам y_1 , y_2 та можуть приймати чотири значення 0 , A_s , 0 , $-A_s$, які циклічно змінюються в залежності від номера прямої (рисунок 3.2). Для формування чотирьох ліній у періоді сигналу трикутної форми використовується змінна s , яка змінюється від 0 до 4 та використовується для розрахунку координат ліній Y_m та Y_{m+1} за формулами:

$$Y_m = \begin{cases} 0, & \text{якщо } s = 0 ; \\ A_s, & \text{якщо } s = 1 ; \\ 0, & \text{якщо } s = 2 ; \\ -A_s, & \text{якщо } s = 3 ; \end{cases} \quad (3.14)$$

$$Y_{m+1} = \begin{cases} A_s, & \text{якщо } s = 0 ; \\ 0, & \text{якщо } s = 1 ; \\ -A_s, & \text{якщо } s = 2 ; \\ 0, & \text{якщо } s = 3 . \end{cases}$$

Формування зразкових цифрових послідовностей сигналу трикутної форми відбувається у циклі для значень $i=0\dots n-1$. Змінна j використовується при формуванні прямої (рисунок 3.2). Початкове значення j рівне нулеві. Змінна j змінюється від 0 до n_1 , з кроком 1 . При досягненні змінною j значення n_1 , змінній j присвоюється значення 0 і починається формування наступної лінії. Змінна k використовується для зміни координат t_k , t_{k+1} , Y_k та Y_{k+1} , які вибираються із попередньо сформованого масиву Y_m (3.14). Початкове значення k рівне нулеві, та збільшується на 1 тоді, коли змінній j присвоюється нульове значення. Кожен раз, коли k збільшується на 1 ,

змінюються значення координат t_k , t_{k+1} , Y_k та Y_{k+1} . Таким чином при формуванні першої лінії будуть використовуватися координати t_0 , t_1 , та Y_0 , Y_1 , другої лінії – t_1 , t_2 , та Y_1 , Y_2 і т.д., доки не завершиться формування зразкової цифрової послідовності сигналу трикутної форми.

Математична модель сигналу пилкоподібної форми. Формування зразкових цифрових послідовностей сигналу пилкоподібної форми аналогічне формуванню зразкових цифрових послідовностей сигналу трикутної форми. На відміну від сигналу трикутної форми, один період якого описується чотирма рівняннями прямої, період сигналу пилкоподібної форми описується одним рівнянням прямої, що спрощує алгоритм моделювання.

Для одного періоду сигналу формується пряма з координатами $(0, 0; T, A_s)$, яку можна описати рівнянням прямої (3.10). Кількість даних, які припадають на один період сигналу, визначається за формулою (3.11). Кількість координат, за якими формуються зразкові цифрові послідовності сигналу пилкоподібної форми, обчислюється для $m = 0 \dots n/n_T$ з кроком 1. Тоді координати прямих сигналу пилкоподібної форми t_m та t_{m+1} , які відповідають координатам t_1 та t_2 виразу (3.10), розраховуються за формулами:

$$\begin{aligned} t_m &= n_T \cdot m \cdot h_d ; \\ t_{m+1} &= n_T \cdot (m + 1) \cdot h_d . \end{aligned} \quad (3.15)$$

Координати осі Y для усіх періодів зразкових цифрових послідовностей сигналу пилкоподібної форми приймають значення $y_1 = 0$, $y_2 = A_s$.

Формування зразкових цифрових послідовностей сигналу пилкоподібної форми відбувається у циклі для значень $i = 0 \dots n-1$. Змінна j використовується при формуванні прямої. Початкове значення j рівне нулеві. Змінна j змінюється від 0 до n_T , з кроком 1. При досягненні змінною j значення n_T , змінній j присвоюється значення 0 і починається формування наступної лінії. Змінна k використовується для зміни координат t_k та t_{k+1} , які

вибираються із попередньо сформованого масиву (3.15). Початкове значення k рівне нулеві, та збільшується на 1 тоді, коли змінній j присвоюється нульове значення. Кожен раз, коли k збільшується на 1, змінюються значення координат t_k та t_{k+1} . Таким чином при формуванні першої лінії будуть використовуватися координати t_0, t_1 , другої лінії – t_1, t_2 , і т.д., доки не завершиться моделювання зразкової цифрової послідовності сигналу пилкоподібної форми. Координати по осі Y є однаковими для усіх прямих.

Розроблений алгоритм формування зразкових цифрових послідовностей трикутної та пилкоподібної форми можна використовувати для формування сигналів будь-якої форми.

Проблема некратності періоду сигналу та часу вимірювання. У процесі розроблення алгоритму для моделювання зразкових цифрових послідовностей сигналів різних форм (на прикладі сигналів трикутної та пилкоподібної форми) виникає проблема некратності частоти сигналу та часу вимірювання (рисунок 3.4).

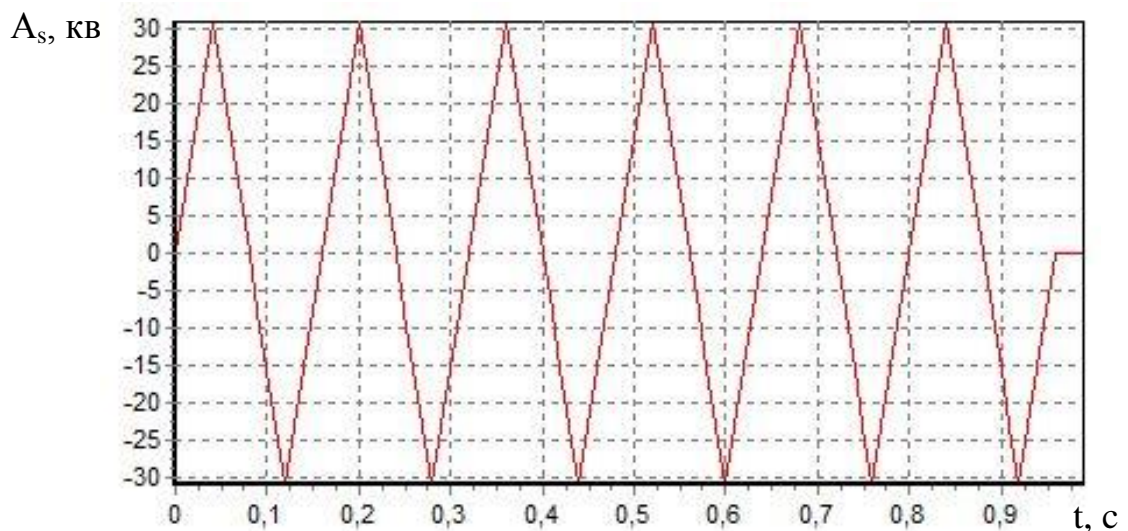


Рисунок 3.4 – Проблема некратності періоду

Розглянемо приклад моделювання зразкових цифрових послідовностей сигналу трикутної форми. Нехай $f_s = 6 \text{ Гц}$, тоді за частоти дискретизації АЦП, наприклад $f_{adc} = 100 \text{ Гц}$; кількість даних за період сигналу $n_T = f_{adc} / f_s = 100 / 6 = 16.66(6)$ є дійсним числом. Тоді, заокругливши n_T до 16, кількість точок дискретизації за 6 періодів сигналу буде рівною 96. Тобто 4 точки при моделюванні не будуть заповнені (рисунок 3.4).

Для вирішення цієї проблеми достатньо сформувати ще одну лінію із координатами значенні $t_m, Y_m, t_{m+1}, Y_{m+1}$ (рисунок 3.5), за якою розраховуються останні 4 значення сигналу.

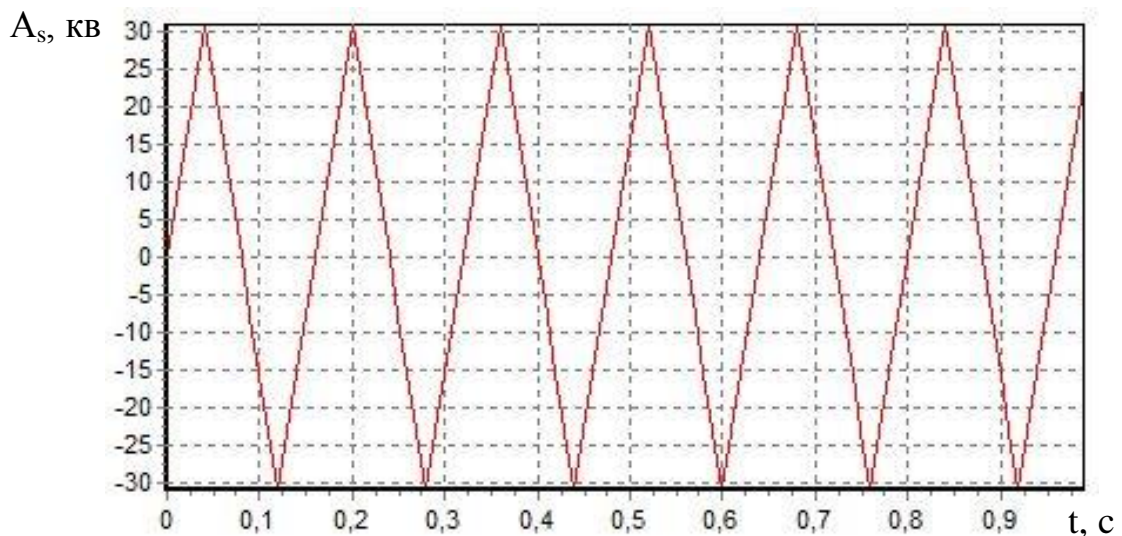


Рисунок 3.5 – Вирішення проблеми некратності періоду

Квантування сигналу. У генераторі зразкових цифрових послідовностей квантування здійснюється за допомогою функції $\text{floor}(x)$, яка повертає найбільше ціле значення рівне або менше за x [108].

Реалізовано такі види квантування:

- до ближчого рівня;
- до більшого рівня;
- до меншого рівня.

Отже квантування до більшого рівня (рисунок 3.6) здійснюється за формулою:

$$Y_{kvant_i} = floor(Y_i + 1). \quad (3.16)$$

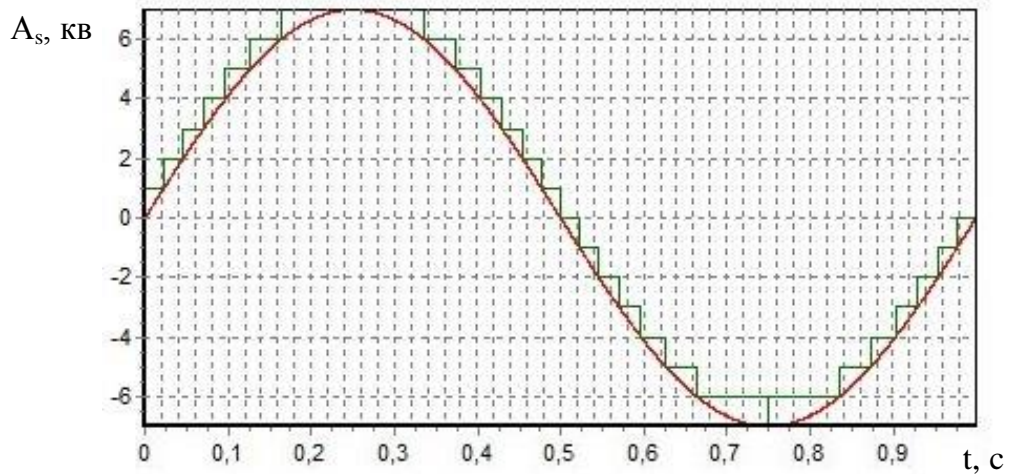


Рисунок 3.6 – Квантування до більшого рівня

Квантування до меншого рівня (рисунок 3.7) відбувається за формулою:

$$Y_{kvant_i} = floor(Y_i). \quad (3.17)$$

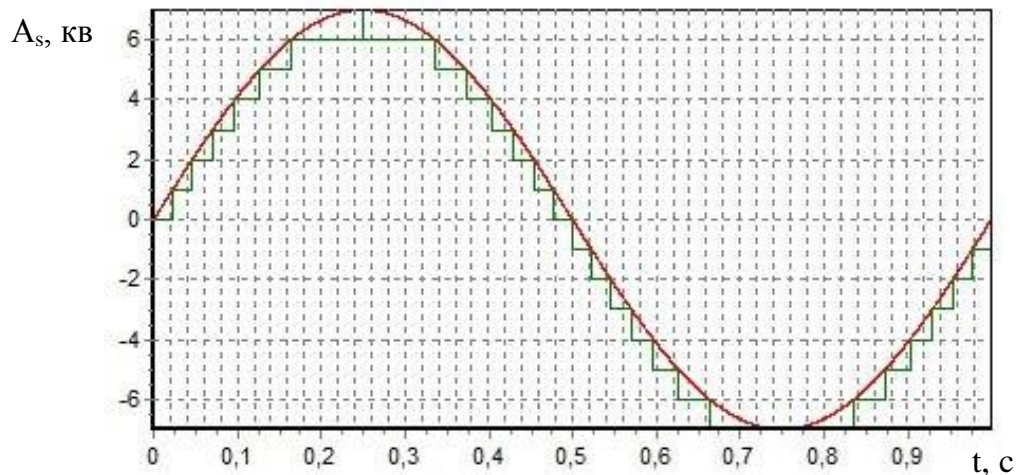


Рисунок 3.7 – Квантування до меншого рівня

Квантування до ближчого рівня (рисунок 3.8) відбувається за формулою:

$$Y_{kvant_i} = floor(Y_i + 0.5). \quad (3.18)$$

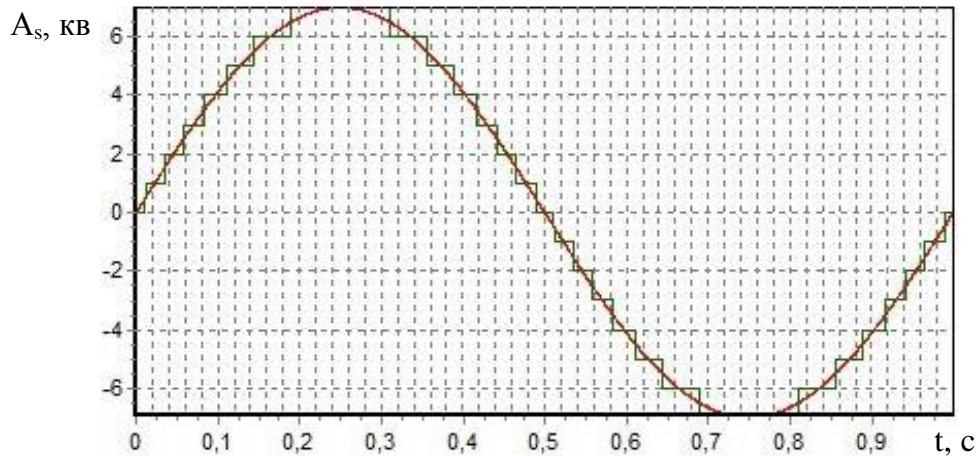


Рисунок 3.8 – Квантування до ближчого рівня

Обмеження сигналу відповідно до діапазону АЦП. Однією із характеристик АЦП є діапазон вхідних значень напруги (наприклад від 0 В до 10 В; від -10 В до +10 В, тощо). Якщо рівень вхідної напруги сигналу перевищує максимальне значення діапазону вхідних значень АЦП, то у генераторі зразкових цифрових послідовностей можливі два способи формування сигналу:

- обмеження сигналу за рівнем;
- масштабування сигналу.

Коефіцієнт $k_{U_{vx}}$ описує відношення реальної напруги на вході АЦП U_{ladc} до номінальної вхідної напруги U_{Nadc} . Якщо $U_{ladc} > U_{Nadc}$, то $k_{U_{vx}} > 1$. У такому випадку амплітуда сигналу у квантах перевищуватиме допустиме значення, тому потрібно програмно обмежити її значення за рівнем, або провести масштабування сигналу.

Для обмеження сигналу за рівнем використано максимальне та мінімальне значення цифрового сигналу:

$$Y_{\max} = n_{kv} / 2 - 1; \quad (3.19)$$

$$Y_{\min} = -(n_{kv} / 2 - 1). \quad (3.20)$$

Виходячи з (3.19), (3.20), обмеження сигналу відбувається за формулами (рисунок 3.9 (а)):

$$Y_i = \begin{cases} Y_{\max}, & \text{якщо } Y_i > Y_{\max} ; \\ Y_{\min}, & \text{якщо } Y_i < Y_{\min} . \end{cases} \quad (3.21)$$

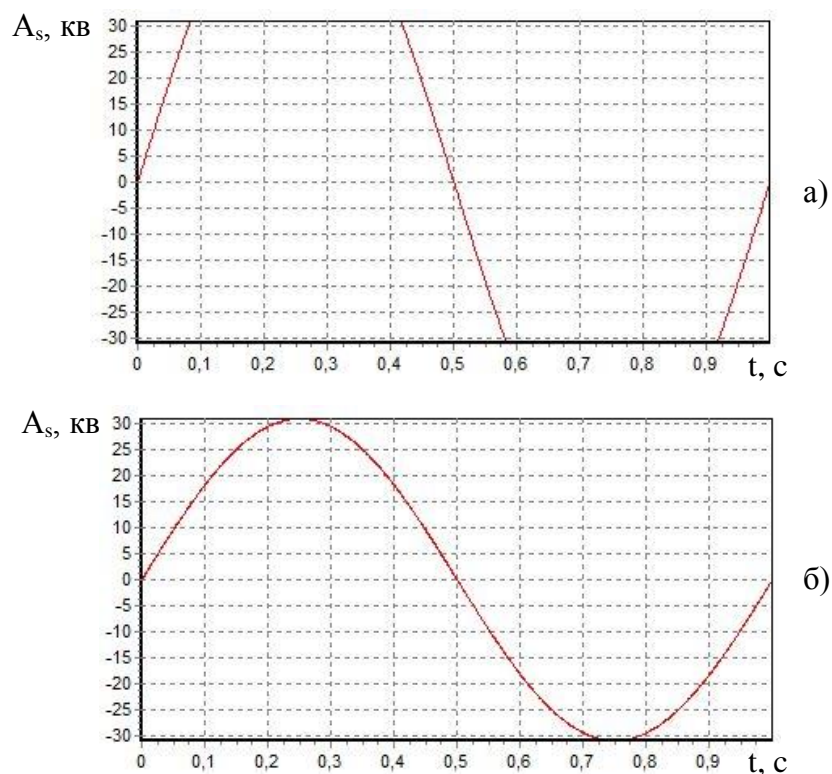


Рисунок 3.9 – Вплив рівня вхідної напруги на сигнал: а) сигнал обмежений за рівнем; б) масштабований сигнал

Масштабування сигналу реалізується за наступною формулою:

$$Y_i = Y_i / k_{U_{vx}}. \quad (3.22)$$

Рівень сигналу буде зменшений до граничного рівня АЦП, якщо сигнал виходить за межі цього рівня. Сигнал у цьому випадку буде такий, як на рисунку 3.9 (б).

Моделі похибок аналогової частини ЗВ. Основними нормованими складовими похибки АЦП є адитивна, мультиплікативна, нелінійна та випадкова складові, які переважно задаються в квантах. При реалізації генератора зразкових цифрових послідовностей використано такі параметри:

- k_{add} – значення адитивної похибки;
- k_{mult} – значення мультиплікативної похибки;
- k_{nl} – значення нелінійної похибки;
- k_{noise} – максимальне значення амплітуди випадкової похибки.

Адитивна складова похибки є сталою величиною. Вона визначається тільки параметром k_{add} :

$$\Delta add_i = k_{add}. \quad (3.23)$$

На рисунку 3.10 представлено графік, побудований за значеннями зразкової цифрової послідовності синусоїдального сигналу, амплітуда якого 30 квантів з адитивною похибкою 5 квантів.

Мультиплікативна складова похибки формується за лінійною залежністю із координатами $(Y_{min}, -k_{mult}; Y_{max}, k_{mult})$:

$$\Delta mult_i = (2 \cdot k_{mult} \cdot (Y_i - Y_{min}) / (Y_{max} - Y_{min}) - k_{mult}). \quad (3.24)$$

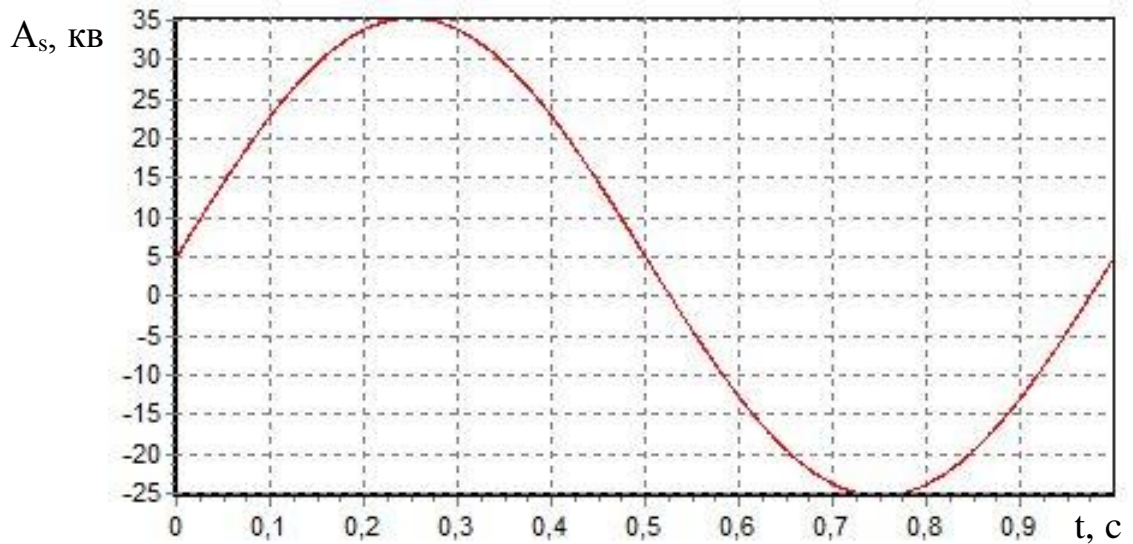


Рисунок 3.10 – Зразкова цифрова послідовність з урахуванням адитивної складової похибки

На рисунку 3.11 представлено графік, побудований за значеннями зразкової цифрової послідовності синусоїдального сигналу, амплітуда якого 30 квантів з мультиплікативною похибкою 5 квантів.

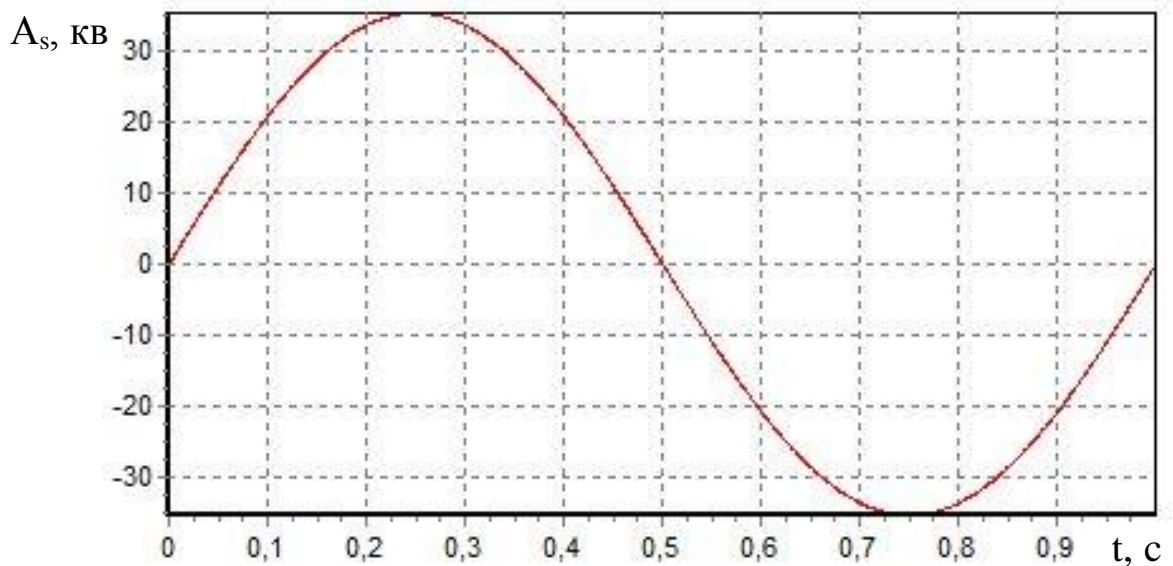


Рисунок 3.11 – Зразкова цифрова послідовність з урахуванням мультиплікативної складової похибки

Нелінійна складова похибки є квадратичною функцією і її можна описати рівнянням:

$$\Delta n l_i = k_{nl} \cdot (1 - Y_i^2 / Y_{\max}^2). \quad (3.25)$$

На рисунку 3.12 представлено графік, побудований за значеннями зразкової цифрової послідовності синусоїдального сигналу, амплітуда якого 30 квантів з нелінійною похибкою 10 квантів.

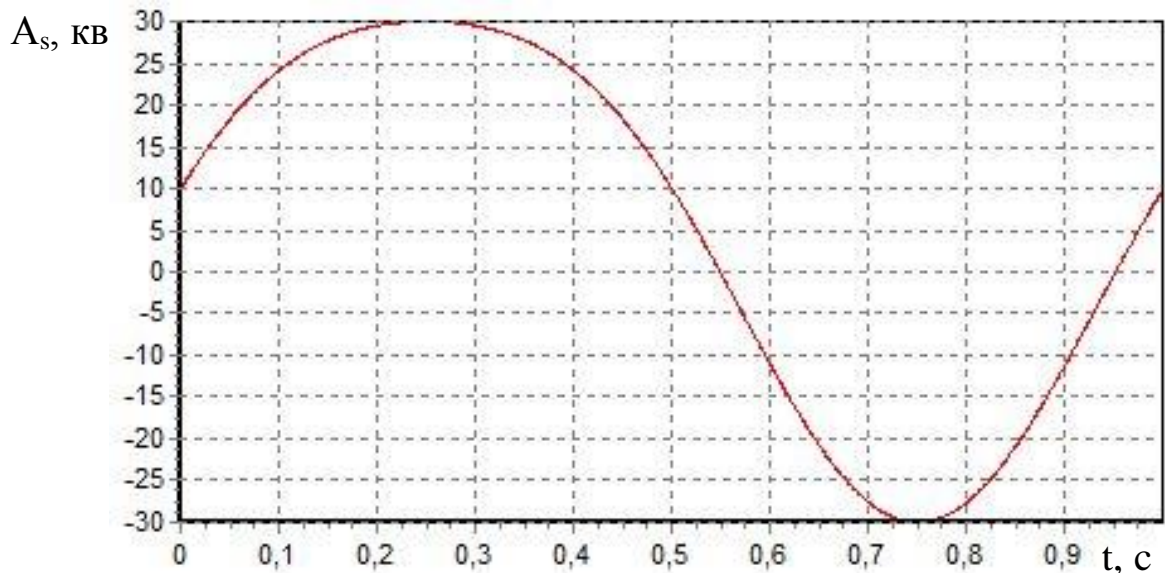


Рисунок 3.12 – Зразкова цифрова послідовність з урахуванням нелінійної складової похибки

Випадкова складова похибки є послідовністю випадкових чисел з середнім значенням близьким до нуля та з рівномірним законом розподілу:

$$\Delta noise_i = 2 \cdot k_{noise} \cdot (rand() / 32767 - 0.5), \quad (3.26)$$

де $rand()$ – функція, що генерує випадкові числа у діапазоні від 0 до 32767.

На рисунку 3.13 представлено графік, побудований за значеннями зразкової цифрової послідовності синусоїдального сигналу, амплітуда якого 30 квантів з випадковою похибкою 3 кванти.

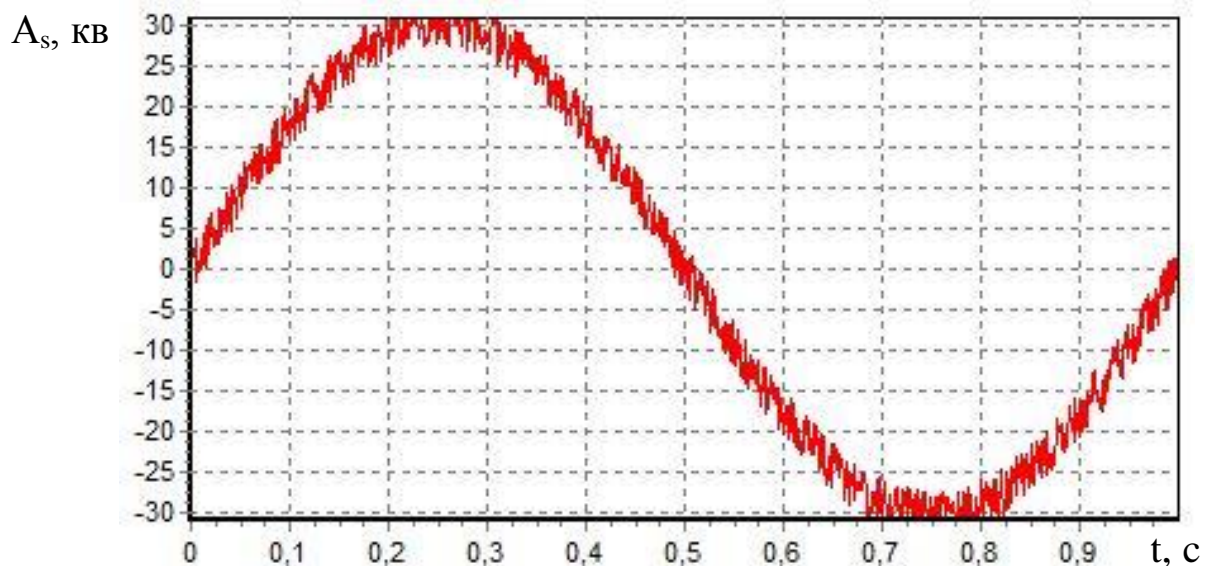


Рисунок 3.13 – Зразкова цифрова послідовність з урахуванням випадкової складової похибки

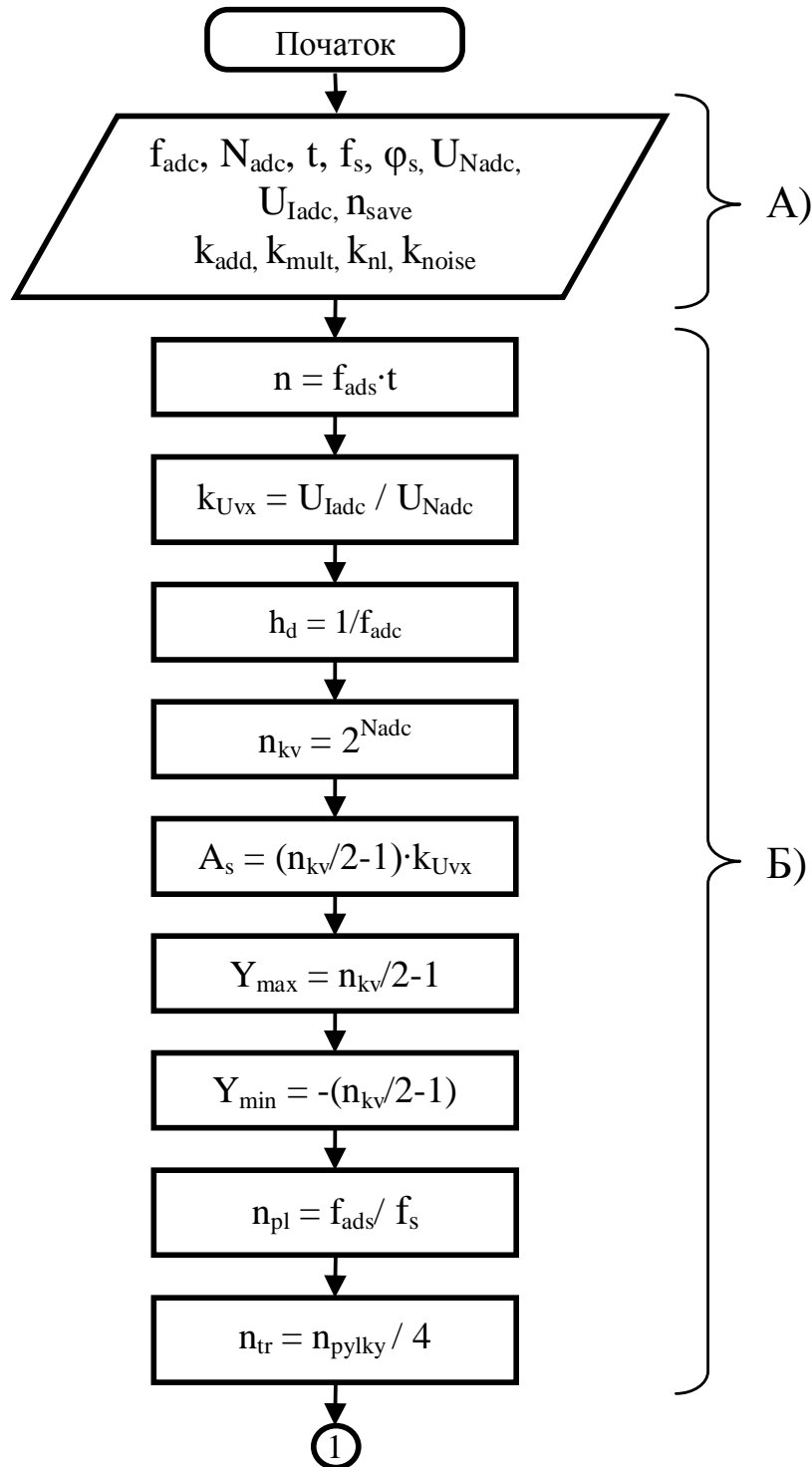
Для формування сигналу врахуванням усіх складових похибки достатньо додати значення цих похибок до значення сигналу:

$$Y_i = Y_i + \Delta add_i + \Delta mult_i + \Delta nl_i + \Delta noise_i . \quad (3.27)$$

Із використанням математичної моделі, описаної вище, розроблено генератор зразкових цифрових послідовностей.

3.2 Алгоритм роботи генератора зразкових цифрових послідовностей

На основі математичної моделі розроблено алгоритм роботи генератора зразкових цифрових послідовностей (рисунок 3.14, 3.15).



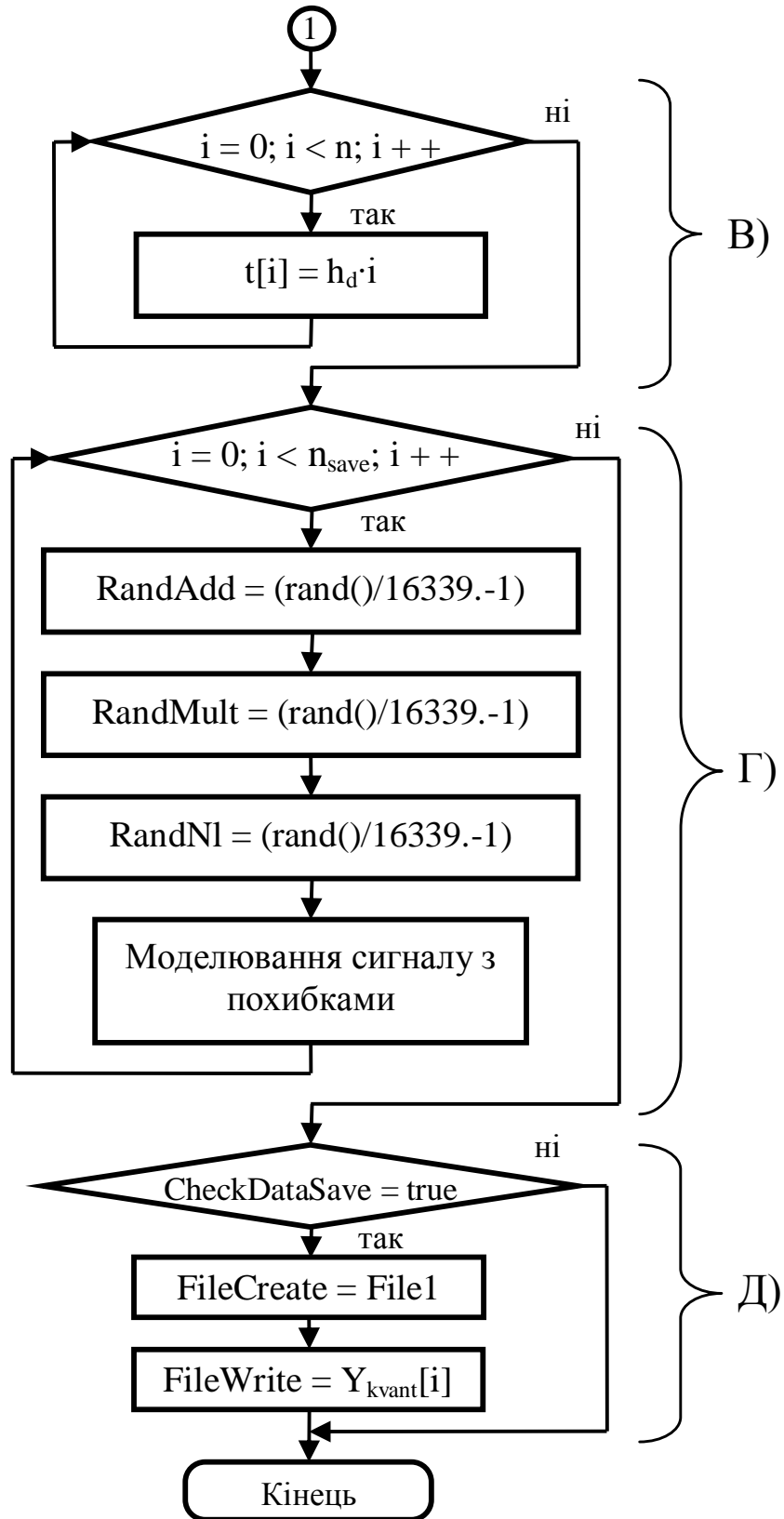


Рисунок 3.14 – Алгоритм роботи генератора зразкових цифрових послідовностей

Для кращого розуміння алгоритм поділено на окремі блоки, які описані нижче.

Блок А. Введення параметрів вимірювального сигналу та АЦП. Даний блок включає в себе введення основних параметрів АЦП, та зразкового сигналу, включаючи коефіцієнти похибок аналогової частини АЦП.

Блок Б. Обчислення необхідних параметрів на основі введених даних. У даному блоці обчислюються параметри, необхідні для формування зразкових тестових послідовностей. За формулою (3.1) обчислюється загальна кількість даних зразкової цифрової послідовності, (3.2) – коефіцієнт вхідної напруги; (3.3) – крок дискретизації, (3.4) - амплітуда сигналу, (3.19) та (3.20) - максимальне та мінімальне значення цифрового сигналу тощо.

Блок В. Моделювання значень біжучих часових координат. За формулою (3.6) формується масив значень часу t_i , який використовується для формування зразкових цифрових послідовностей.

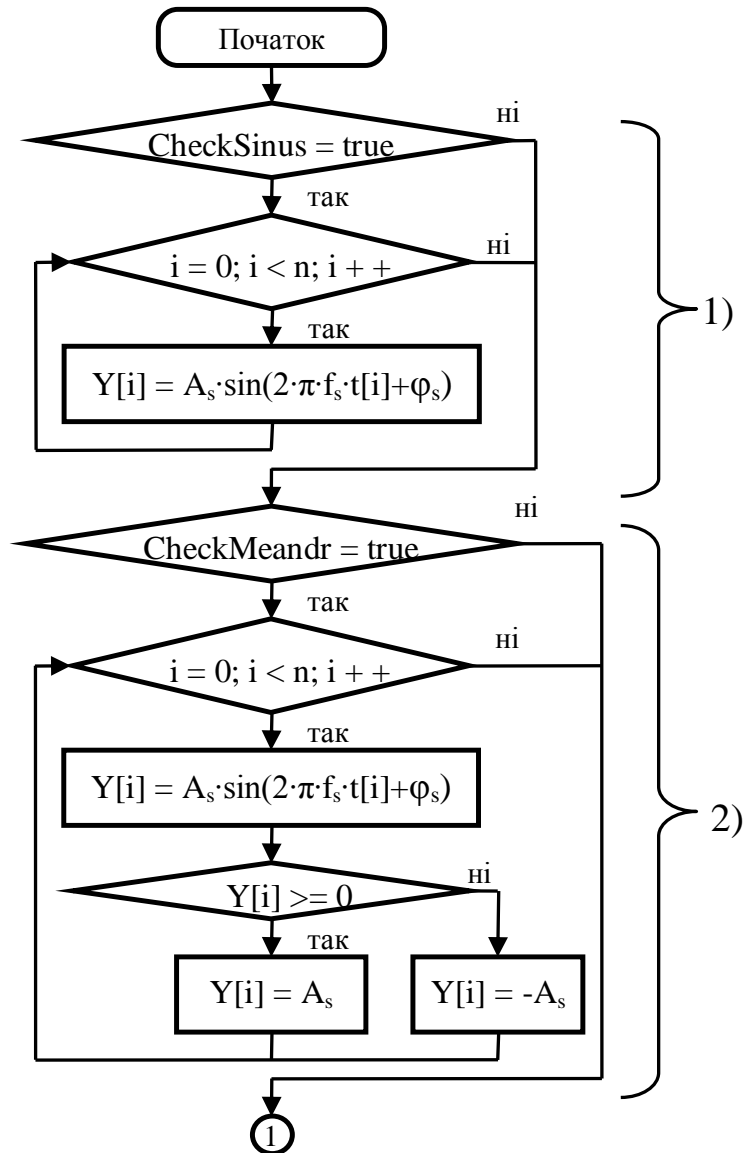
Блок Г. Моделювання зразкових цифрових послідовностей сигналів. Даний блок включає в себе формування зразкових цифрових послідовностей сигналів різних форм, моделювання похибок, обмеження сигналу за рівнем, масштабування та квантування сигналу. В режимі метрологічної перевірки враховується лише квантування сигналу.

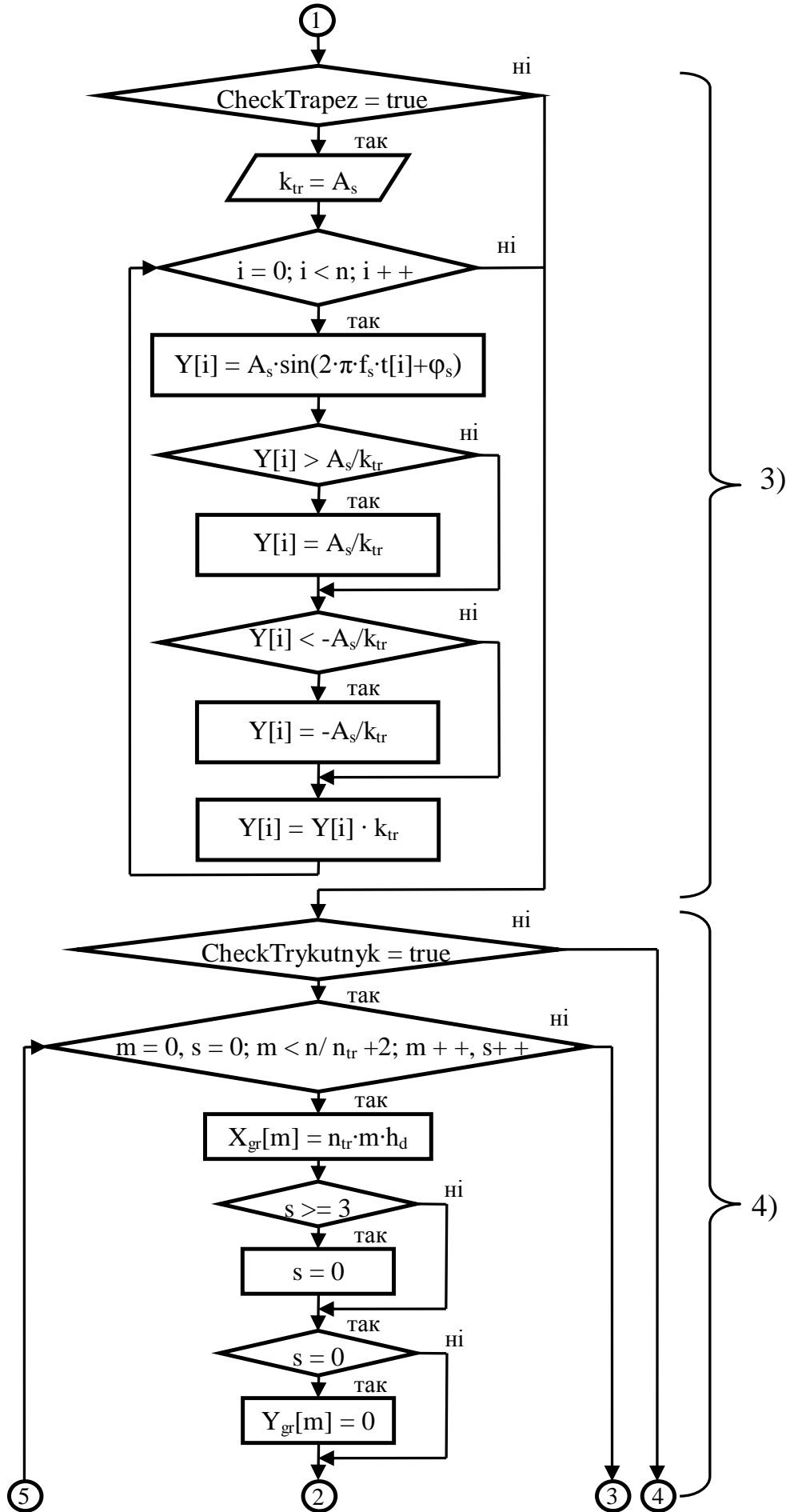
В режимі відлагоджування програм є можливість враховувати окремо адитивну, мультиплікативну, нелінійну та випадкову похибки аналогової частини, зокрема АЦП, а також суму усіх складових похибки одночасно або суму деяких з них. Забезпечено можливість формування однієї або декількох (параметр n_{save}) зразкових цифрових послідовностей, для кожної яких випадковим чином розраховується значення похибки в заданому діапазоні.

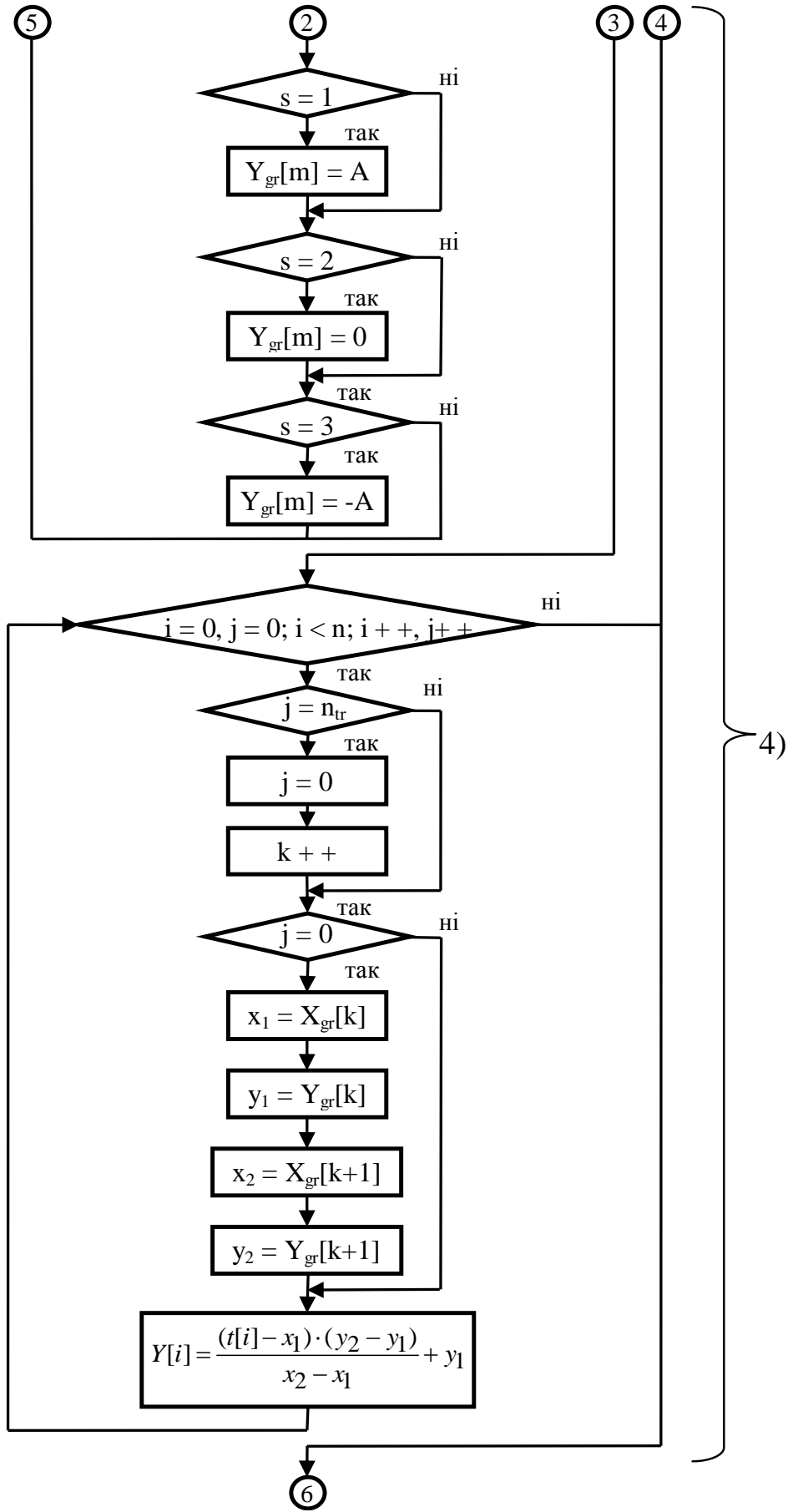
Блок Д. Запис в файл. Запис даних у файл відбувається при натисканні кнопки «Зберегти файл» (рисунок 3.17). Результатом роботи

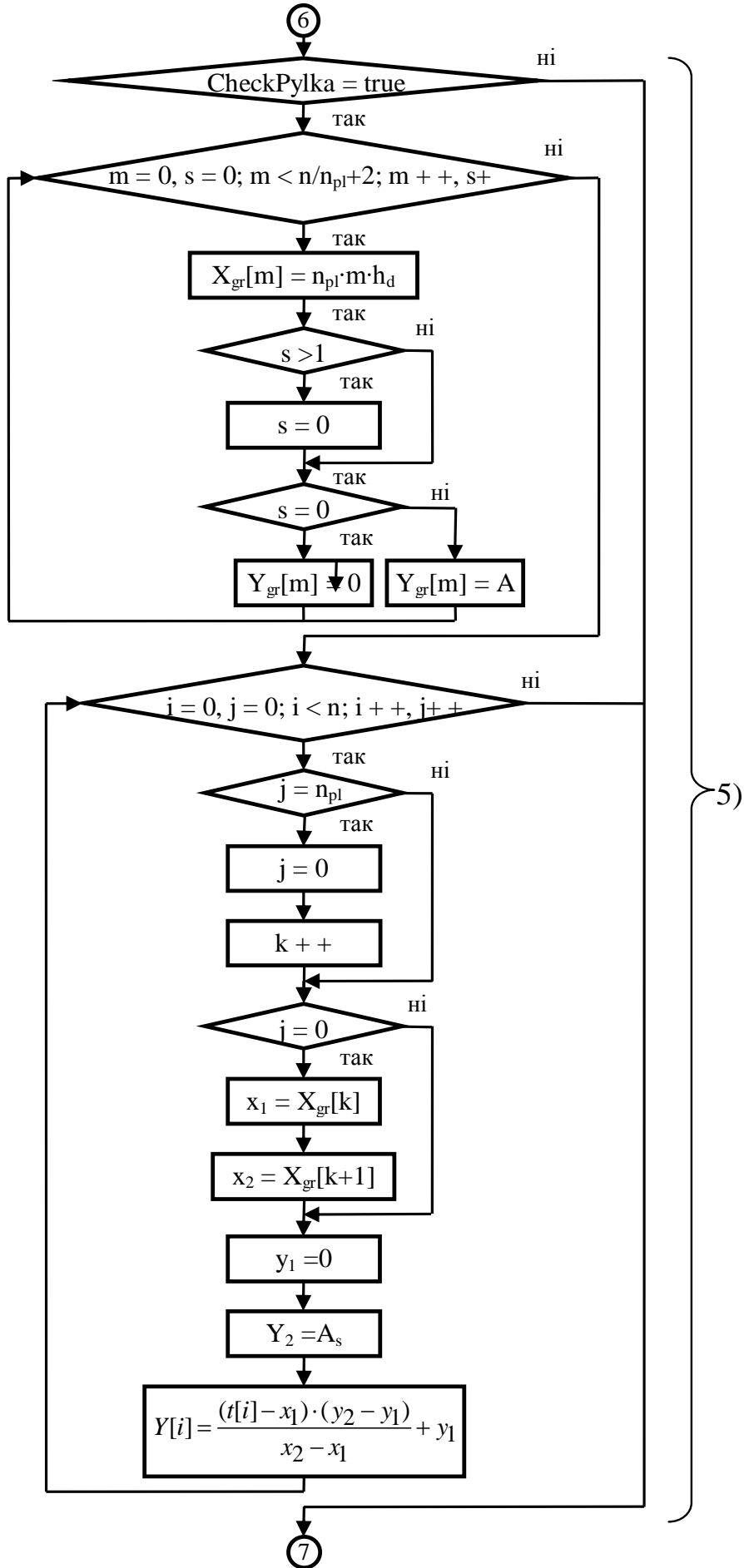
генератора зразкових цифрових послідовностей є файл, який містить одну або декілька зразкових цифрових послідовностей.

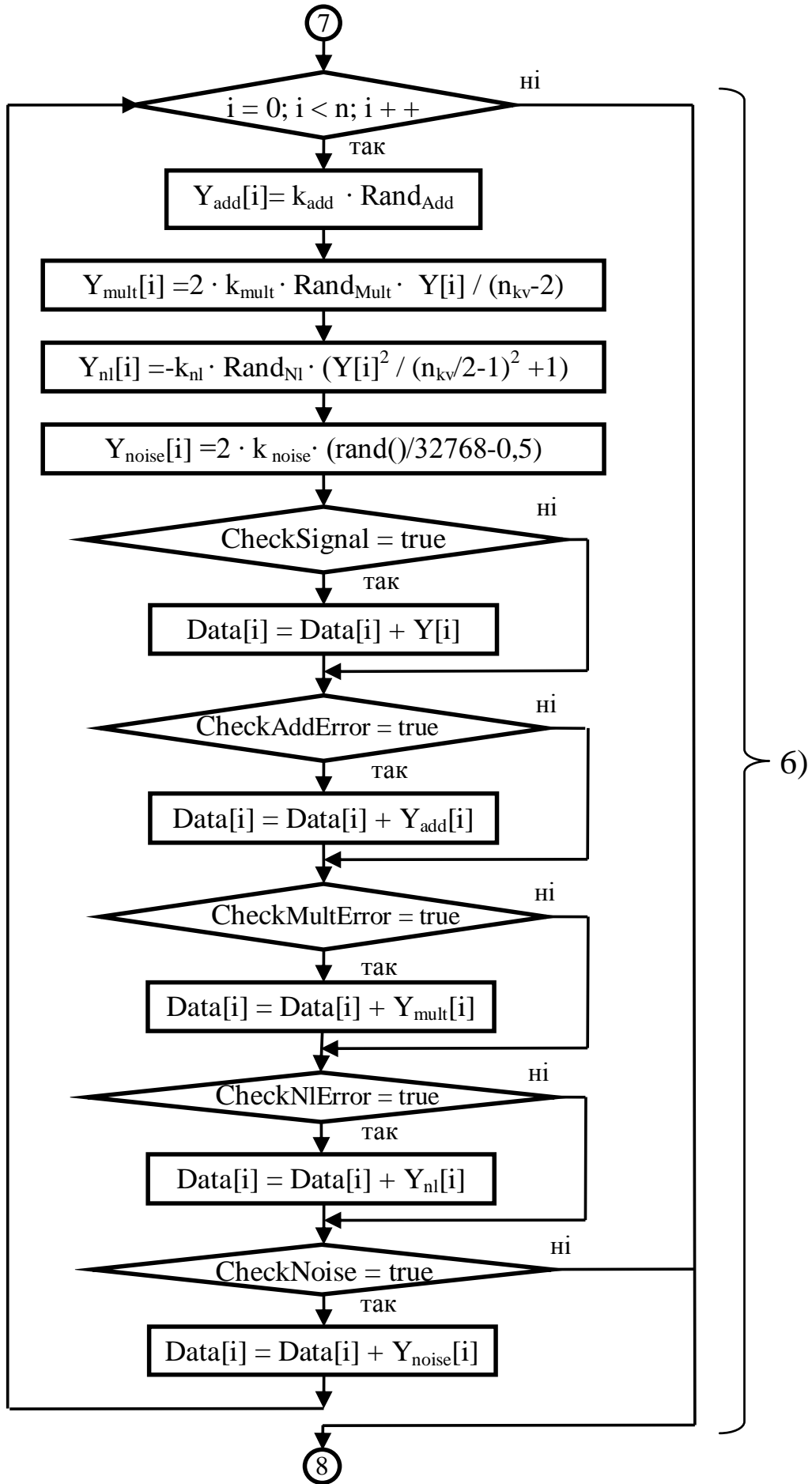
На рисунку 3.15 представлено детальний алгоритм формування зразкових цифрових послідовностей. Алгоритм поділено на окремі блоки, які описані нижче.

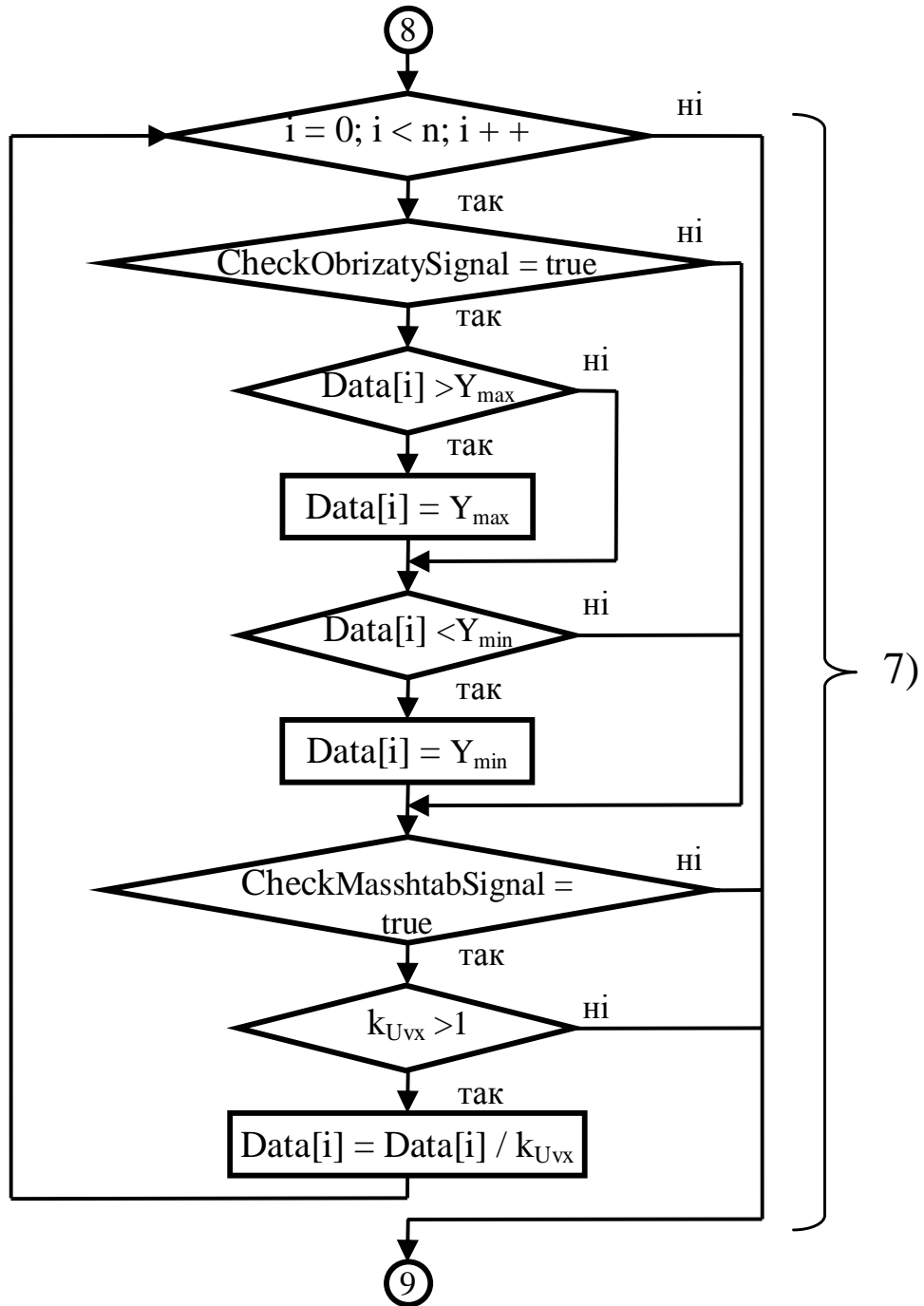












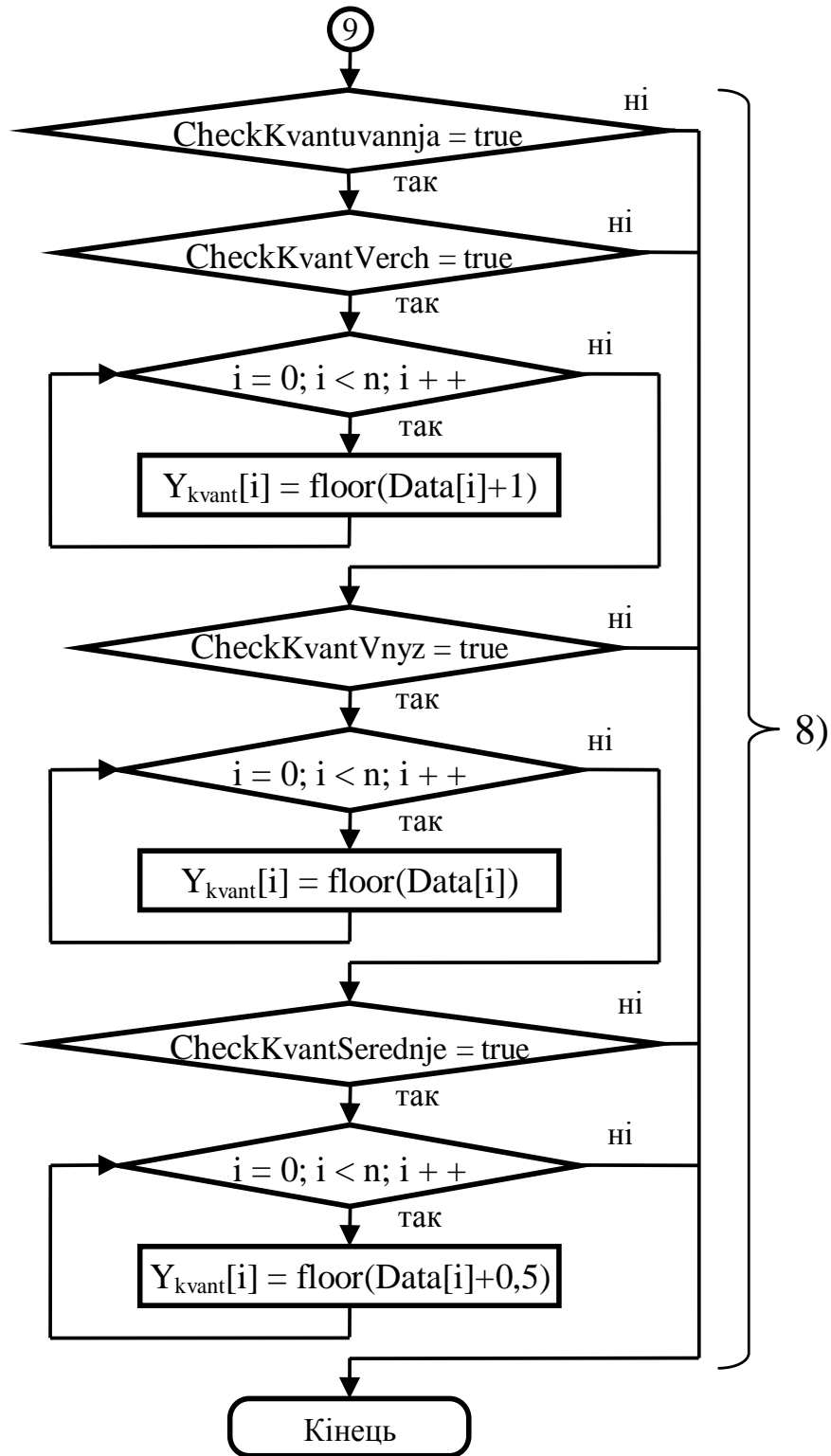


Рисунок 3.15 – Алгоритм формування зразкових цифрових послідовностей

Блок 1. Формування зразкових цифрових послідовностей сигналу синусоїдальної форми. На основі введених даних та проведених обчислень за формулою (3.5) відбувається моделювання зразкових цифрових послідовностей сигналу синусоїдальної форми. При цьому враховується значення реальної вхідної напруги АЦП U_{Iadc} , номінальна вхідна напруга АЦП U_{Nadc} , розрядність АЦП N_{adc} , час вимірювання t , частота f_{adc} та фаза сигналу φ_s .

Блок 2. Формування зразкових цифрових послідовностей сигналу типу меандр. Моделювання зразкових цифрових послідовностей сигналу типу меандр базується на онові синусоїдального сигналу. Розрахунок значень зразкової послідовності сигналу типу меандр відбувається за формулою (3.7).

Блок 3. Формування зразкових цифрових послідовностей сигналу трапецієподібної форми. Моделювання зразкових цифрових послідовностей сигналу трапецієподібної форми базується також на онові синусоїдального сигналу. Сформовані дані обмежуються на певному рівні за формулою (3.8). Рівень обмеження зразкових цифрових послідовностей сигналу трапецієподібної форми задається коефіцієнтом трапеції k_t , який також використовується для приведення значень зразкової цифрової послідовності трапецієподібної форми до значення реальної вхідної напруги АЦП U_{Iadc} за формулою (3.9).

Блок 4. Формування зразкових цифрових послідовностей сигналу трикутної форми. Формування зразкових цифрових послідовностей сигналу трикутної форми відбувається із використанням рівняння прямої. У даному блоці за формулами (3.10), (3.11), (3.12), (3.13), (3.14) проводиться розрахунок значень зразкових цифрових послідовностей сигналу трикутної форми.

Блок 5. Формування зразкових цифрових послідовностей сигналу пилкоподібної форми. Формування зразкових цифрових послідовностей сигналу пилкоподібної форми відбувається також із використанням рівняння

прямої. Алгоритм моделювання сигналу пилкоподібної форми, який представлений у даному блоці, є спрощеним варіантом алгоритму моделювання зразкових цифрових послідовностей сигналу трикутної форми. Розрахунок значень зразкових цифрових послідовностей сигналу пилкоподібної форми відбувається за формулами (3.10), (3.11), (3.15).

Блок 6. Моделювання похибок. Блок 6 виконується лише у режимі відлагоджування програми. У даному блоці алгоритму реалізовано моделювання адитивної – формула (3.23); мультиплікативної – формула (3.24), нелінійної – формула (3.25) та випадкової – формула (3.26) похибок. До сигналу можуть бути додані всі похибки, деякі з них, або жодної.

Блок 7. Обмеження за рівнем / масштабування сигналу. У даному блоці, в залежності від вибору оператора, відбувається або обмеження за рівнем сигналу – формули (3.19), (3.20), (3.21), або масштабування сигналу – формула (3.22). Даний блок виконується лише у випадку, якщо реальна напруга на вході АЦП перевищує номінальну напругу АЦП.

Блок 8. Квантування сигналу. У даному блоці відбувається квантування аналогового сигналу в залежності від розрядності АЦП. Реалізовано три способи квантування: до більшого рівня – за формулою (3.16); до меншого рівня – за формулою (3.17); до ближчого рівня – за формулою (3.18).

3.3 Інтерфейс користувача генератора зразкових цифрових послідовностей

Описаний вище генератор зразкових цифрових послідовностей може бути реалізований у будь якому програмному середовищі, наприклад, MATLAB, Mathcad, LabVIEW, C++Builder тощо. В роботі представлено

реалізацію генератора зразкових цифрових послідовностей у об'єктно-орієнтованому програмному середовищі C++Builder.

Розроблено інтерфейс користувача генератора зразкових цифрових послідовностей, який складається із п'яти вікон:

1. вікно «Режим роботи»;
2. вікно «Формування зразкових цифрових послідовностей»;
3. вікно «Параметри АЦП»;
4. вікно «Параметри сигналу»;
5. вікно «Обмеження / масштабування сигналу».

При запуску програми відкривається вікно «Режим роботи», у якому можна вибрати два режими: «Метрологічна перевірка» або «Відлагоджування програмного забезпечення» (рисунок. 3.16).

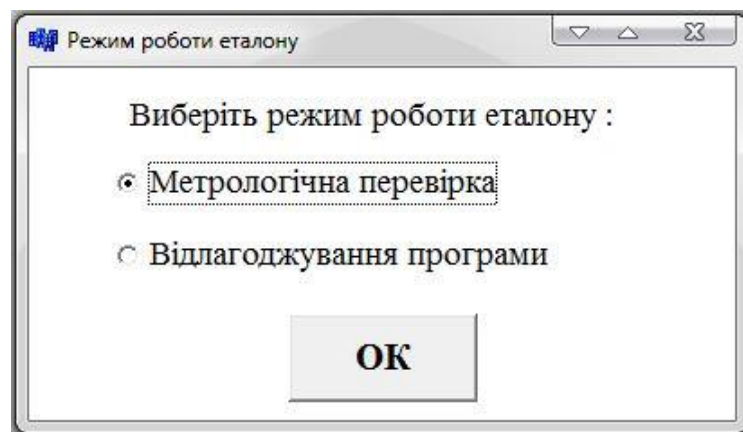


Рисунок 3.16 – Вікно «Режим роботи»

Після вибору потрібного режиму роботи при натисканні кнопки «ОК» відбувається перехід до вікна «Формування зразкових цифрових послідовностей» (рисунок 3.17). Кнопка «Старт» запускає моделювання зразкових цифрових послідовностей сигналу вибраної форми. За замовчуванням вибрано сигнал синусоїдальної форми.

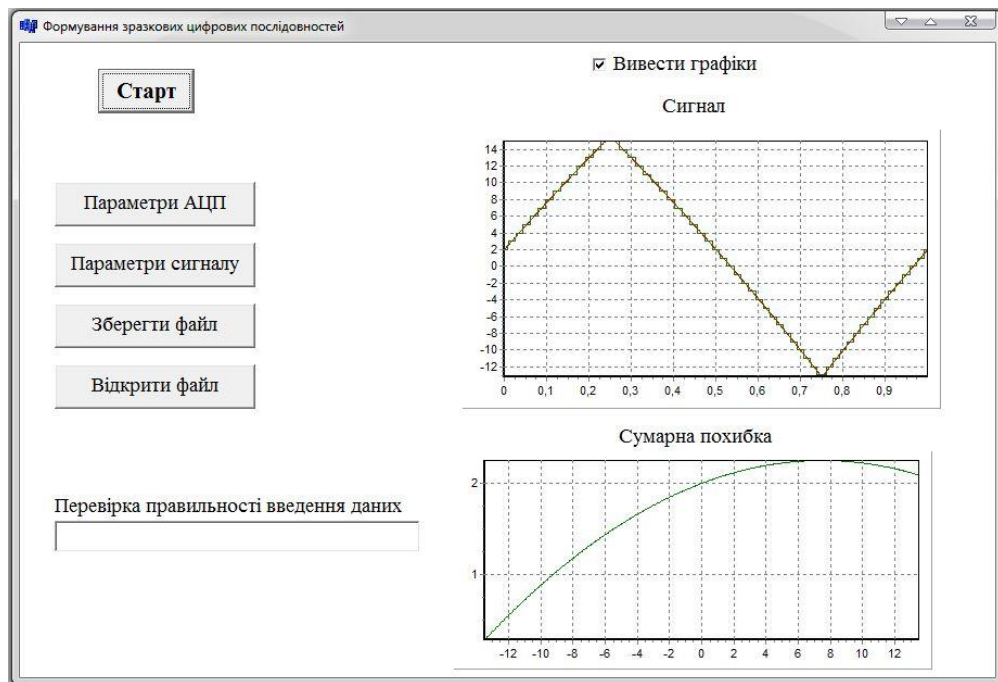


Рисунок 3.17 – Вікно «Формування зразкових цифрових послідовностей»

Вікно «Формування зразкових цифрових послідовностей» має декілька кнопок керування:

- кнопка «Параметри АЦП» – дозволяє перейти до вікна введення параметрів АЦП;
- кнопка «Параметри сигналу» – дозволяє перейти до вікна, у якому можна задати параметри сигналу;
- кнопка «Зберегти файл» – дозволяє записати зразкові цифрові послідовності у файл. Можна задати назву та місце запису файлу;
- кнопка «Відкрити файл» використовується для завантаження попередньо сформованих і збережених файлів зразкових цифрових послідовностей та їх візуалізації у графічному представленні.

Під час формування зразкових цифрових послідовностей реалізована можливість візуалізації сигналу та сумарної похибки з якою формується сигнал. Візуалізація відбувається лише тоді, коли вибрано прапорець «Вивести графіки». Графік похибок виводиться лише у режимі

«Відлагоджування програми». На графіку «Сигнал» виводиться як аналоговий, так і дискретний сигнал. При одночасному формуванні декількох зразкових цифрових послідовностей (у режимі відлагоджування програми), а також при великій кількості даних не рекомендується використовувати візуалізацію, оскільки це збільшить час формування зразкових цифрових послідовностей.

У вікні «Параметри АЦП» (рисунок 3.18) можна задати розрядність, частоту дискретизації, час перетворення, номінальну та реальну вхідну напругу АЦП.

Параметри АЦП

OK

Розрядність, біт
16

Частота дискретизації, Гц
10000

Час перетворення, с
1

Номінальна вхідна напруга, В
10

Реальна вхідна напруга, В
10

Тип квантування

- Ввімкнути квантування
- До більшого рівня
- До меншого рівня
- До ближчого рівня

Похибки АЦП

Складові похибки	Значення похибки	
	кВ	мВ
<input type="checkbox"/> Адитивна	1	0
<input type="checkbox"/> Мультиплікативна	1	0
<input type="checkbox"/> Нелінійна	1	0
<input type="checkbox"/> Випадкова	3	0

Додаткові похибки

Ввімкнути похибки

Складові похибки	Значення похибки	
	кВ	мВ
<input type="checkbox"/> Адитивна	0	0
<input type="checkbox"/> Мультиплікативна	0	0
<input type="checkbox"/> Нелінійна	0	0
<input type="checkbox"/> Випадкова	0	0

Рисунок 3.18 – Вікно «Параметри АЦП»

Також вікно «Параметри АЦП» містить три додаткових поля: «Тип квантування», «Похибки АЦП» та «Додаткові похибки». Прапорець «Ввімкнути квантування» активує опцію квантування сигналу, яке може бути до більшого, меншого або ближчого рівня. Поля «Похибки АЦП» та «Додаткові похибки» активні тільки у режимі «Відлагоджування програми». У полі «Похибки АЦП» можна вибрати складові похибки АЦП, що будуть враховані при формуванні зразкових цифрових послідовностей, та їх числові значення.

Числові значення похибки можна задати в квантах або в мілівольтах. Якщо користувач задав значення складової похибки в квантах, поле введення значення цієї складової похибки у мілівольтах стає неактивним, і навпаки. У полі «Додаткові похибки» можна ввести значення інших похибок аналогової частини ЗВ. Значення похибок у полі «Додаткові похибки» можна вводити тоді, коли вибрано прапорець «Ввімкнути похибки». Підтвердження введення параметрів відбувається при натисканні кнопки «ОК». За кнопкою «ОК» також здійснюється обчислення значення похибок. Якщо похибка задана у квантах, то вона обчислюється у мілівольтах, а якщо задана у мілівольтах, то обчислюється в квантах, і заповнюються відповідні поля «Похибки АЦП» та «Додаткові похибки». Після цього здійснюється перехід до вікна «Формування зразкових цифрових послідовностей». Для моделювання сигналу з новими параметрами АЦП потрібно натиснути кнопку «Старт» (рисунок 3.17).

У вікні «Параметри сигналу» (рисунок 3.19) можна задати числові значення фази і частоти сигналу, а також вибрати форму сигналу. Підтвердження введення параметрів та перехід до вікна «Формування зразкових цифрових послідовностей» реалізується натисканням кнопки «ОК».

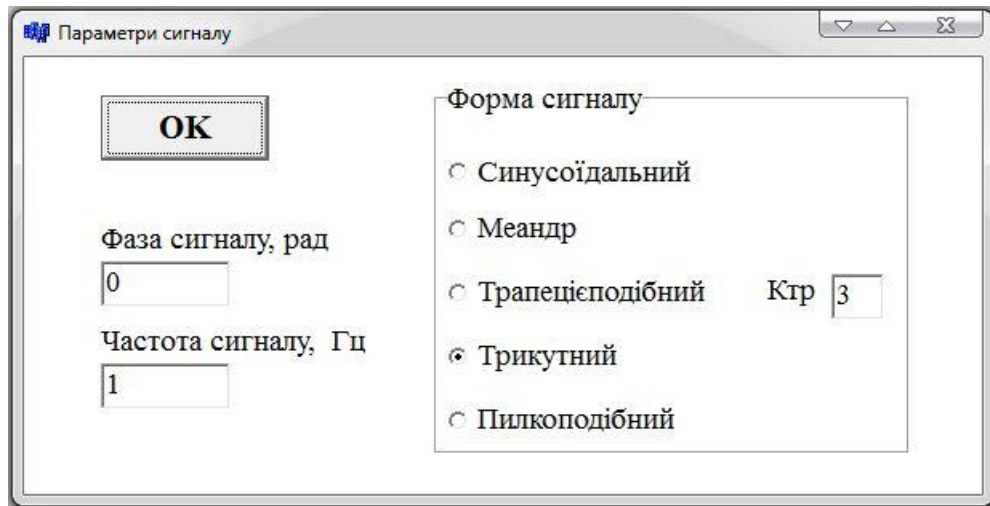


Рисунок 3.19 – Вікно «Параметри сигналу» генератора зразкових цифрових послідовностей

Відкриття вікна «Обмеження / масштабування сигналу» (рисунок 3.20) відбувається тільки у тому випадку, коли реальна напруга на вході АЦП перевищує номінальну. У генераторі зразкових цифрових послідовностей реалізовано два способи формування сигналу: з обмеженням (рисунок 3.9, а) або масштабуванням сигналу (рисунок 3.9, б).

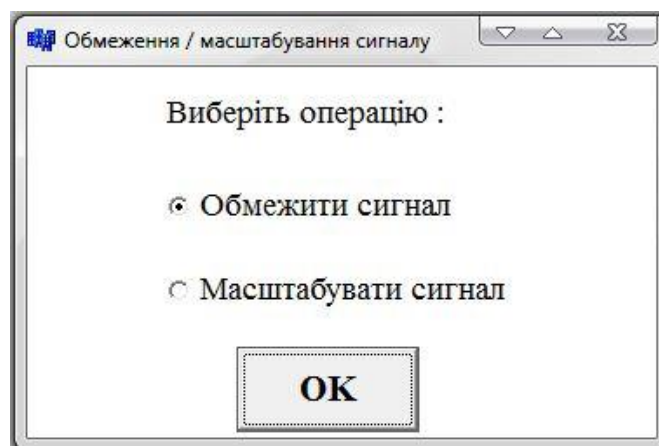


Рисунок 3.20 – Вікно «Обмеження / масштабування сигналу»

Розроблений інтерфейс користувача дає змогу у зручній формі задавати параметри сигналів та АЦП для формування зразкових цифрових послідовностей.

Генератор зразкових цифрових послідовностей можна інтегрувати у будь-який програмний пакет сучасних засобів вимірювання (типу MATLAB, LabVIEW, тощо) як окремий блок для метрологічної перевірки програмного забезпечення, а також як засіб відлагоджування програмного забезпечення при розробці засобу вимірювання. Запропонований алгоритм роботи генератора зразкових цифрових послідовностей реалізований у програмних середовищах Builder C++, MATLAB і Mathcad.

Висновки до розділу 3

1. Розроблено генератор зразкових цифрових послідовностей, зокрема, його математичну модель та алгоритм роботи, що дозволяє формувати зразкові цифрові послідовності з врахуванням параметрів вимірюваного сигналу: форми, частоти, амплітуди, фази; параметрів АЦП: розрядності, частоти дискретизації, часу перетворення, адитивної, мультиплікативної, нелінійної та випадкової похибок.
2. Генератор зразкових цифрових послідовностей може працювати у режимі метрологічної перевірки або режимі відлагоджування програми.
3. Розроблено алгоритм формування зразкових цифрових послідовностей, сигналів трикутної та пілкоподібної форми, який може використовуватися для моделювання сигналів складної форми.
4. Розроблено інтерфейс користувача, який дозволяє у зручній формі задавати параметри сигналів та АЦП для формування зразкових цифрових послідовностей.
5. Генератор зразкових цифрових послідовностей можна інтегрувати у будь-який програмний пакет сучасних засобів вимірювання (типу MATLAB, LabVIEW, тощо) як окремий блок для метрологічної перевірки програмного забезпечення, а також як засіб відлагоджування програмного забезпечення при розробці засобу вимірювання.

РОЗДІЛ 4

ДОСЛІДЖЕННЯ ПОХИБОК ТА НЕПЕВНОСТЕЙ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАННЯ НА ПРИКЛАДІ ФУНКЦІЇ ПЕРЕТВОРЕННЯ ФУР'Є

У четвертому розділі з використанням розробленого генератора зразкових цифрових послідовностей проведено метрологічну перевірку ПЗ на прикладі функції перетворення Фур'є. Дана функція вибрана як предмет дослідження тому, що вона є достатньо складною в алгоритмічному та математичному плані, має багато варіантів реалізації у різних програмних продуктах [93 - 95]. Дослідження проведено у програмних пакетах MATLAB, та Mathcad. Обчислювалася похибка визначення гармонік функцією перетворення Фур'є (далі похибка функції `fft()`).

Комп'ютерні обчислення проводяться з цифровими даними, що призводить до виникнення похибки результату обчислень за рахунок дискретизації даних та округлення результатів. Похибка результату обчислень залежить від типу використовуваних даних (`int`, `float`, `double`, тощо). Крім того сам алгоритм опрацювання даних може вносити додаткову похибку [96-100]. Дослідження похибок програмного забезпечення проводилося для двох типів даних:

- типу `int`;
- типу `double`.

`Double` – тип даних з плаваючою комою, який використовують для вирішення завдань з високою точністю обчислень. `Int` – тип даних, що представляє ціле число [101-103]. Тип `double` у порівнянні з типом `int` може вважатися зразковим. Дані типу `int` відповідають вихідним даним АЦП.

Сучасна метрологія для оцінювання точності результатів вимірювань використовує два підходи: похибок та непевностей [24, 104-110]. Тому в роботі для оцінювання метрологічних характеристик ПЗ ЗВ застосовано обидва підходи.

Із використанням розробленого генератора зразкових цифрових послідовностей запропоновано засіб метрологічної перевірки ПЗ ЗВ (рисунок 4.1), який складається з двох блоків: генератора зразкових цифрових послідовностей та блоку оцінювання точності. Між цими блоками розміщується код програми (компонента, функція, тощо), що перевіряється. Запропонований засіб метрологічної перевірки ПЗ ЗВ можна інтегрувати у будь-який програмний пакет сучасних ЗВ (типу MATLAB, LabVIEW) як окрему компоненту.

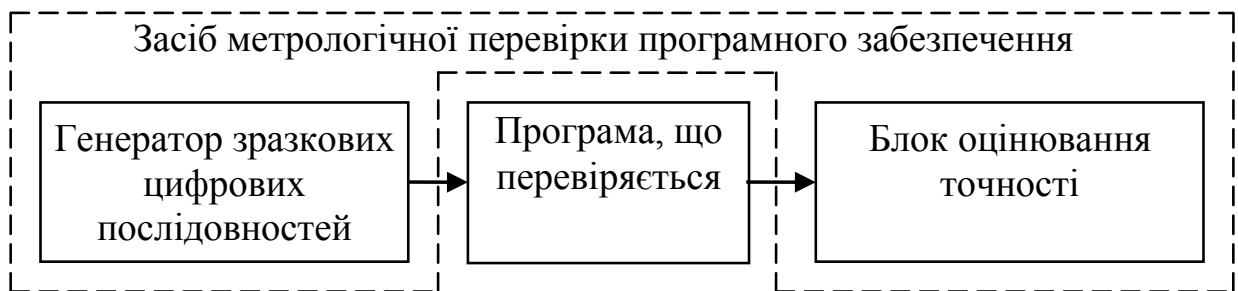


Рисунок 4.1 – Засіб метрологічної перевірки ПЗ ЗВ

Метрологічна перевірка програмного забезпечення на прикладі функції перетворення Фур'є ($\text{fft}()$) відбувається таким чином:

1. формується одна, або декілька зразкових цифрових послідовностей;
2. виконується функція $\text{fft}()$ із використанням сформованих даних;
3. відбувається оцінка похибок функції $\text{fft}()$.

У випадку, коли виробник програмного забезпечення надає інформацію про похибку або непевність функції $\text{fft}()$, то порівнявши результати метрологічної перевірки з регламентованими, можна зробити

висновок щодо придатності даної функції до використання. Якщо такої інформації немає, то можлива лише оцінка похибки або непевності.

4.1 Дослідження похибки функції перетворення Фур'є програмного пакету MATLAB для даних типу double

Оцінка точності функції перетворення Фур'є `fft()` проводилася із використанням теорії похибок. Проводилося дослідження похибок функції `fft()` в залежності від:

- частоти сигналу f_s ;
- фази сигналу φ_s ;
- частоти дискретизації АЦП f_{adc} .

Дослідження залежності похибки функції `fft()` від частоти сигналу (рисунок 4.2). Дослідження проводилося за таких параметрів: $f_{adc} = 100 \text{ кГц}$; $t = 1 \text{ с}$; $A_s = 10 \text{ В}$; $\varphi_s = 0^\circ$; сигнал синусоїдальної форми, частота сигналу f_s змінювалася від 1 до 4999 Гц.

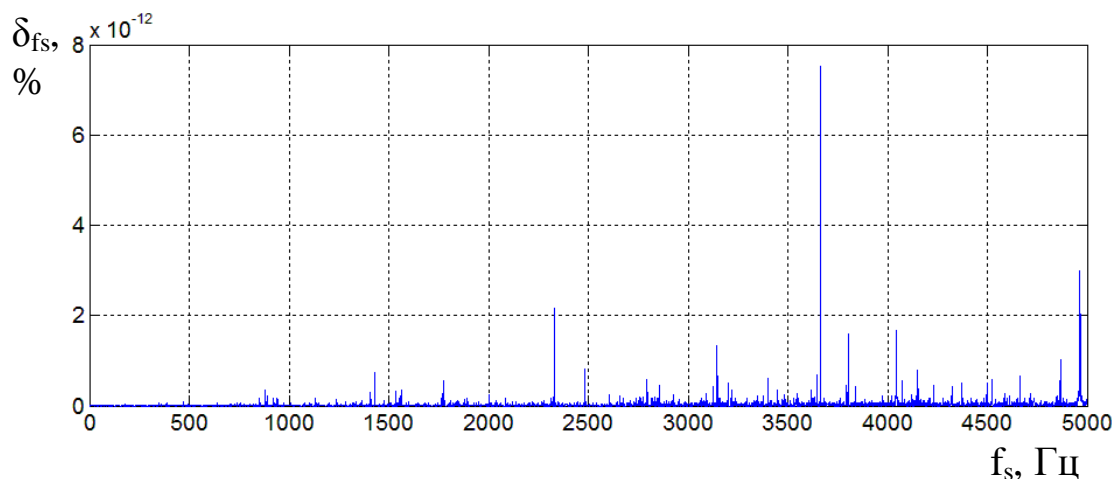


Рисунок 4.2 – Залежність відносної похибки функції `fft()` від частоти сигналу

Для досліджуваного діапазону частот відносна похибка функції перетворення Фур'є не перевищує $8 \cdot 10^{-12}\%$. При низьких значеннях частоти відносна похибка не перевищує $2 \cdot 10^{-14}\%$. Із зростанням частоти сигналу відносна похибка функції $\text{fft}()$ зростає.

Дослідження залежності похибки функції $\text{fft}()$ від фази сигналу.
Дослідження проводилося за таких параметрів: $f_{adc} = 100 \text{ кГц}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $A_s = 10 \text{ В}$; сигнал синусоїдальної форми, фаза сигналу φ_s змінювалася від 0 до 360° . Відносна похибка функції перетворення Фур'є при зміні фази сигналу не перевищує $2 \cdot 10^{-14}\%$ (рисунок 4.3).

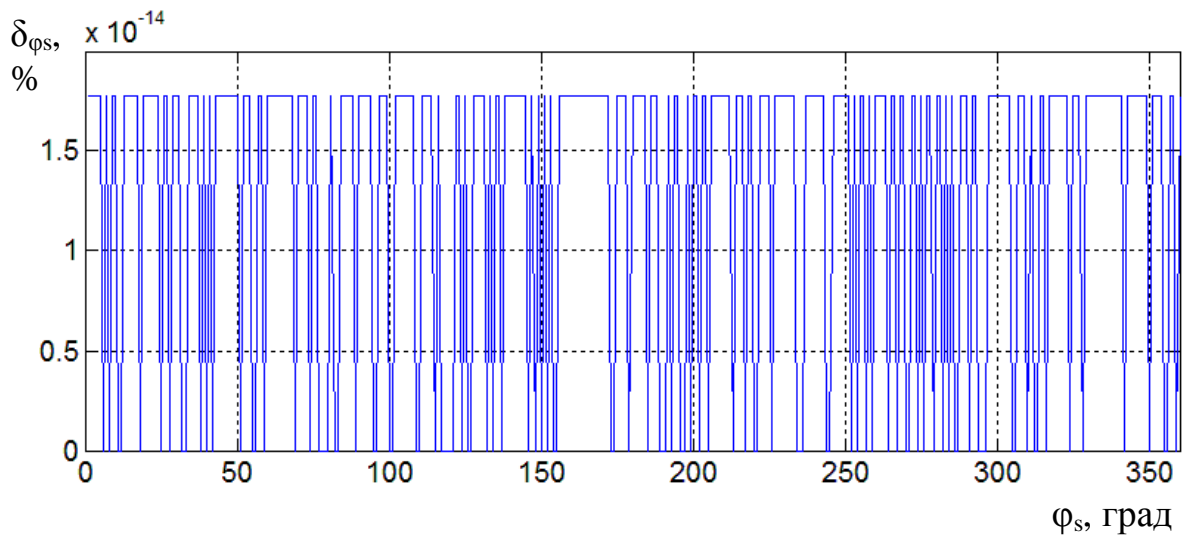


Рисунок 4.3 – Залежність відносної похибки функції $\text{fft}()$ від фази сигналу

На рисунку 4.4 представлено залежність максимальної відносної фазової похибки функції перетворення Фур'є від частоти сигналу. При зростанні частоти сигналу відносна фазова похибка функції перетворення Фур'є зростає практично на 4 порядки, якщо частота сигналу наближується до $\frac{1}{2}$ частоти дискретизації АЦП.

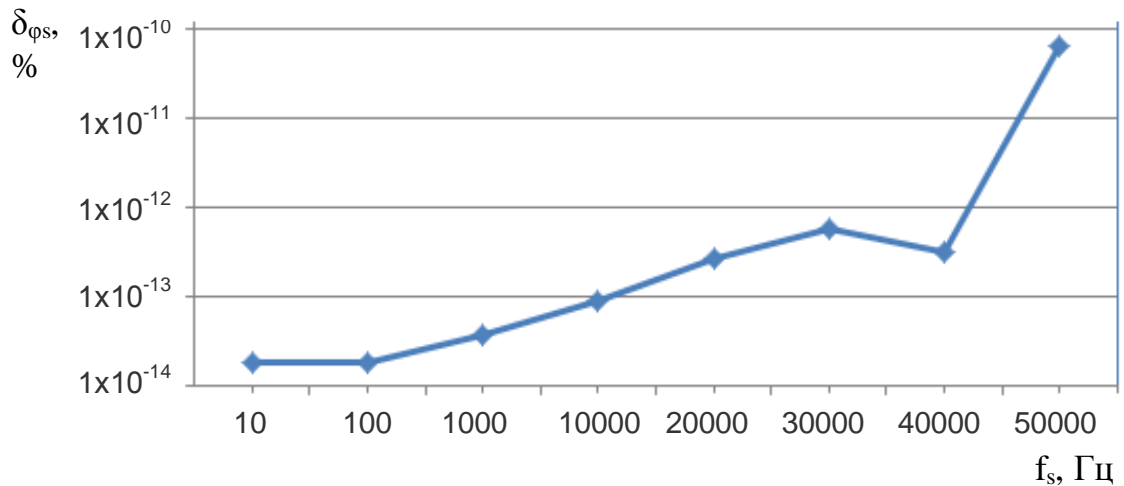


Рисунок 4.4 – Залежність відносної похибки функції перетворення Фур'є від фази сигналу при зміні частоти сигналу

Дослідження залежності похибки функції $\text{fft}()$ від частоти дискретизації АЦП (рисунок 4.5). Дослідження проводилося за таких параметрів: $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $A_s = 10 \text{ В}$; $\varphi_s = 0^\circ$; сигнал синусоїдальної форми; частота дискретизації АЦП f_{adc} змінювалася від 100 Гц до 100 кГц.

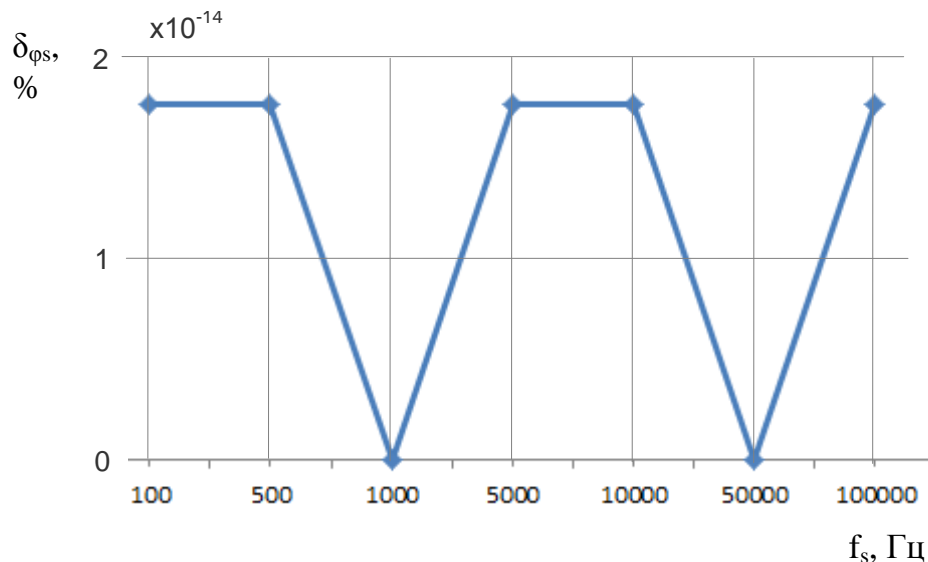


Рисунок 4.5 – Залежність відносної похибки функції перетворення Фур'є від частоти дискретизації АЦП

Для досліджуваного діапазону частот дискретизації АЦП відносна похибка функції перетворення Фур'є не перевищує $2 \cdot 10^{-14} \%$.

За частоти сигналу 10 Гц зміна частоти дискретизації АЦП від 100 Гц до 100 кГц практично не впливає на значення відносної похибки функції перетворення Фур'є.

4.2 Дослідження похибки функції перетворення Фур'є програмного пакету MATLAB для даних типу int

Під час дослідження похибок функції `fft()` для даних типу `int` при формуванні зразкових цифрових послідовностей враховувалась розрядність АЦП. Оцінка точності функції `fft()` проводилася із використанням теорії похибок. Проводилося дослідження похибок функції `fft()` від:

- частоти сигналу f_s ;
- фази сигналу φ_s ;
- рівня сигналу U_{i_adc} ;
- частоти дискретизації АЦП f_{adc} ;
- розрядності АЦП N_{adc} ;
- типу квантування;
- форми сигналу.

Дослідження залежності похибки функції `fft()` від частоти сигналу.
Дослідження проводилося за таких параметрів: $N_{adc} = 16 \text{ bit}$; $f_{adc} = 100 \text{ кГц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 10 \text{ В}$; сигнал синусоїдальної форми, квантування до ближчого рівня, частота сигналу f_s змінювалася від 1 до 4999 Гц. На рисунку 4.6 представлено залежність відносної похибки функції `fft()` від частоти сигналу. Максимальна відносна похибка функції `fft()` не перевищує $6 \cdot 10^{-4} \%$.

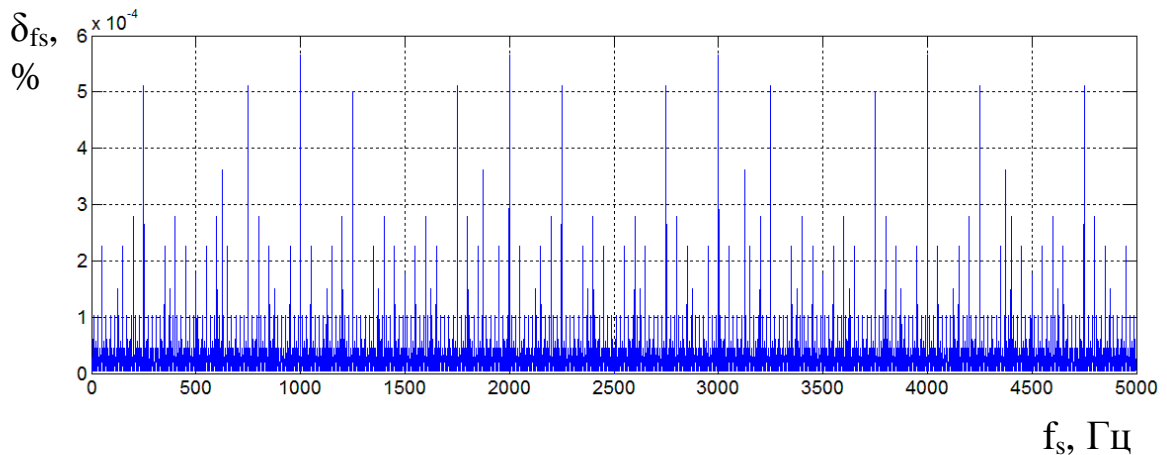


Рисунок 4.6 – Залежність відносної похибки функції $\text{fft}()$ від частоти сигналу

Отже, перехід від типу `double` до типу `int` з врахуванням розрядності АЦП призводить до суттєвого збільшення похибки функції $\text{fft}()$. При використанні тестових послідовностей типу `double` максимальна відносна похибка δf_{max} функції $\text{fft}()$ становить $7,5 \cdot 10^{-12} \%$ (рисунок 4.2), а для квантованого сигналу ($N_{\text{adc}} = 16 \text{ біт}$) – $5,7 \cdot 10^{-4} \%$ (рисунок 4.6). Таким чином квантування сигналу за рівнем збільшило похибку функції $\text{fft}()$ практично на 8 порядків.

Дослідження залежності похибки функції $\text{fft}()$ від фази сигналу (рисунок 4.7). Дослідження проводилося за таких параметрів: $N_{\text{adc}} = 16 \text{ біт}$; $f_{\text{adc}} = 100 \text{ кГц}$; $t = 1 \text{ с}$; $U_{n_{\text{adc}}} = 10 \text{ В}$; $U_{i_{\text{adc}}} = 10 \text{ В}$; $f_s = 10 \text{ Гц}$; сигнал синусоїдальної форми, квантування до ближчого рівня, фаза сигналу φ_s змінюється від 0 до 360° . На представлено залежність відносної похибки функції $\text{fft}()$ від фази сигналу.

При використанні тестових послідовностей типу `double` максимальна відносна похибка $\delta \varphi_{\text{max}}$ функції $\text{fft}()$ становить $1,8 \cdot 10^{-14} \%$ (рисунок 4.3), а для квантованого сигналу ($N_{\text{adc}} = 16 \text{ біт}$) – $1,8 \cdot 10^{-4} \%$ (рисунок 4.7). Таким чином похибка $\delta \varphi$ функції $\text{fft}()$ зростає практично на 10 порядків.

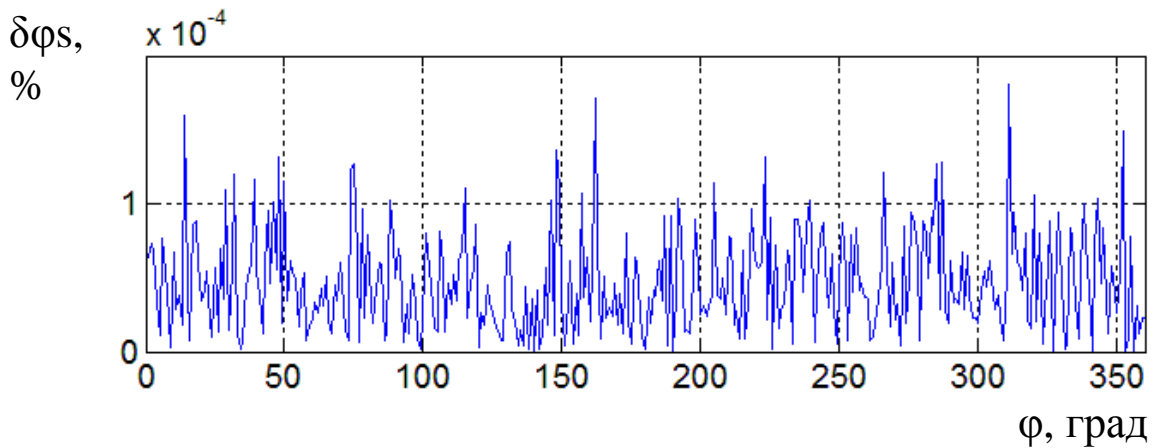


Рисунок 4.7 – Залежність відносної похибки функції $\text{fft}()$ від фази сигналу

Дослідження залежності похибки функції $\text{fft}()$ від рівня сигналу. На рисунку 4.8 представлено залежність відносної похибки функції $\text{fft}()$ від рівня сигналу на вході АЦП. Рівень сигналу U_{i_adc} змінюється від 1 до 10 В. Інші параметри є такими самими, як у попередньому дослідженні.

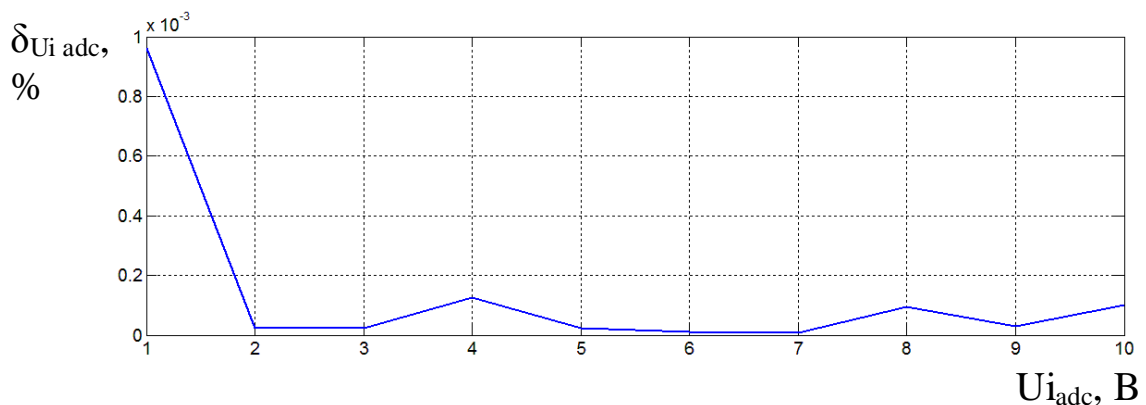


Рисунок 4.8 – Залежність відносної похибки функції $\text{fft}()$ від рівня сигналу

За результатами проведених досліджень можна зробити висновок, що із зменшенням рівня сигналу на вході АЦП похибка функції $\text{fft}()$ зростає.

Дослідження залежності похибки функції $\text{fft}()$ від частоти дискретизації АЦП. Дослідження проводилося за таких параметрів: $N_{adc} = 16 \text{ біт}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 10 \text{ В}$; сигнал

синусоїдальної форми; квантування до ближчого рівня; частота дискретизації АЦП f_{adc} змінювалася від 100 Гц до 100 кГц. Результати досліджень подано на рисунку 4.9.

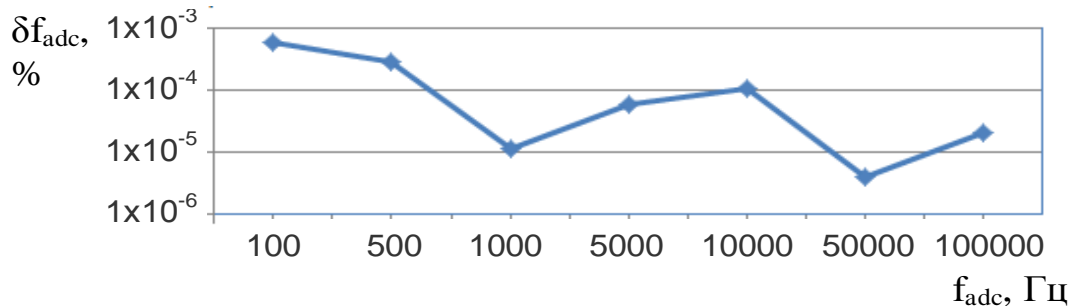


Рисунок 4.9 – Залежність відносної похибки функції перетворення Фур'є від частоти дискретизації АЦП

При використанні тестових послідовностей типу double максимальна відносна похибка $\delta f_{adc \max}$ функції $\text{fft}()$ становить $1,8 \cdot 10^{-14}\%$ (рисунок 4.5), а для квантованого сигналу ($N_{adc} = 16 \text{ біт}$) – $5,7 \cdot 10^{-4}\%$ (рисунок 4.9). Таким чином похибка δf_{adc} функції $\text{fft}()$ зростає практично на 9 порядків.

Дослідження залежності похибки функції $\text{fft}()$ від розрядності АЦП (рисунок 4.10).

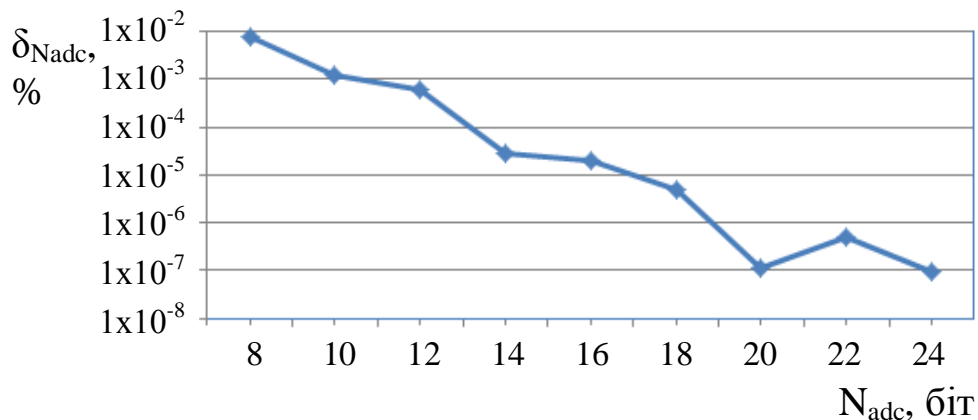


Рисунок 4.10 – Залежність відносної похибки функції перетворення Фур'є від розрядності АЦП

Дослідження проводилося за таких параметрів: $f_{adc} = 100 \text{ кГц}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 10 \text{ В}$; сигнал синусоїдальної форми; квантування до ближчого рівня, розрядність АЦП N_{adc} змінювалася від 8 до 24 біт. Зі збільшенням розрядності АЦП спостерігається зменшення відносної похибки функції перетворення Фур'є.

Дослідження залежності похибки функції $\text{fft}()$ від типу квантування. Дослідження проводилося за таких параметрів: $f_{adc} = 100 \text{ кГц}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $N_{adc} = 16 \text{ біт}$; $U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 10 \text{ В}$; сигнал синусоїдальної форми; квантування до більшого, меншого та ближчого рівня. При квантуванні даних до більшого та меншого рівня похибка функції $\text{fft}()$ є однаковою (таблиця 4.1).

Враховуючи результати проведеного дослідження (таблиця 4.1), можна сказати, що похибка функції $\text{fft}()$ від типу квантування при зміні частоти дискретизації АЦП немає однозначної функціональної залежності. Не завжди оптимальним є квантування до ближчого рівня. У деяких випадках кращим є квантування до більшого або меншого рівня.

Таблиця 4.1 – Залежність відносної похибки функції $\text{fft}()$ від типу квантування

Тип квантування	Частота дискретизації АЦП, Гц					
	5000	10000	25000	50000	100000	200000
До ближчого рівня	$5.69 \cdot 10^{-05}$	$1.02 \cdot 10^{-04}$	$2.57 \cdot 10^{-05}$	$3.81 \cdot 10^{-06}$	$2.01 \cdot 10^{-05}$	$8.74 \cdot 10^{-06}$
До більшого рівня	$7.36 \cdot 10^{-06}$	$1.92 \cdot 10^{-05}$	$3.89 \cdot 10^{-05}$	$1.66 \cdot 10^{-05}$	$7.06 \cdot 10^{-07}$	$1.63 \cdot 10^{-05}$
До меншого рівня	$7.36 \cdot 10^{-06}$	$1.92 \cdot 10^{-05}$	$3.89 \cdot 10^{-05}$	$1.66 \cdot 10^{-05}$	$7.06 \cdot 10^{-07}$	$1.63 \cdot 10^{-05}$

Дослідження залежності похибки функції $\text{fft}()$ від форми сигналу.

Для дослідження впливу форми сигналу на функцію перетворення Фур'є слід врахувати, що усі сигнали, окрім сигналу синусоїдальної форми, містять у собі певну кількість гармонік. Тому, для оцінки впливу форми сигналу, як зразкові використано числові послідовності типу `double`. Для кожного досліджуваного сигналу створювалося дві цифрові послідовності, одна з даними типу `double`, інша – з даними типу `int`. Для кожної з отриманих послідовностей проведено перетворення Фур'є, після чого порівнювались гармоніки двох спектрів.

Дослідження проводилося за таких параметрів: $f_{adc} = 10 \text{ кГц}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $A_s = 9,5 \text{ В}$; $\varphi_s = 0^\circ$, квантування до ближчого рівня. Форма сигналу: типу меандр, трапецієподібна, трикутна та пилкоподібна. Розрядність АЦП змінювалася від 8 до 24 біт. Результати проведених досліджень представлені в таблицях 4.2, 4.3.

Таблиця 4.2 – Залежність середньої відносної похибки функції перетворення Фур'є від форми сигналу при зміні розрядності АЦП

Форма сигналу	Розрядність АЦП, біт				
	8	12	16	20	24
Типу меандр	$4,74 \cdot 10^{-4}$	$2,94 \cdot 10^{-5}$	$1,84 \cdot 10^{-6}$	$1,15 \cdot 10^{-7}$	$7,18 \cdot 10^{-9}$
Трапецієподібна	$1,79 \cdot 10^{-4}$	$1,18 \cdot 10^{-5}$	$8,26 \cdot 10^{-7}$	$4,98 \cdot 10^{-8}$	$3,40 \cdot 10^{-9}$
Трикутна	$3,01 \cdot 10^{-4}$	$1,95 \cdot 10^{-5}$	$1,18 \cdot 10^{-6}$	$3,10 \cdot 10^{-8}$	$5,00 \cdot 10^{-9}$
Пилкоподібна	$3,28 \cdot 10^{-4}$	$2,14 \cdot 10^{-5}$	$1,22 \cdot 10^{-6}$	$7,99 \cdot 10^{-8}$	$4,34 \cdot 10^{-9}$

На рисунку 4.11 представлено графік залежності середньої відносної похибки функції перетворення Фур'є від форми сигналу при зміні розрядності АЦП.

Таблиця 4.3 – Залежність максимальної відносної похибки функції перетворення Фур'є від форми сигналу при зміні розрядності АЦП

Форма сигналу	Розрядність АЦП, біт				
	8	12	16	20	24
Типу меандр	0,37	$2,29 \cdot 10^{-2}$	$1,4 \cdot 10^{-3}$	$8,95 \cdot 10^{-5}$	$5,59 \cdot 10^{-6}$
Трапецієподібна	0,35	$2,17 \cdot 10^{-2}$	$1,4 \cdot 10^{-3}$	$8,56 \cdot 10^{-5}$	$5,28 \cdot 10^{-6}$
Трикутна	0,18	$1,53 \cdot 10^{-2}$	$5,71 \cdot 10^{-4}$	$1,70 \cdot 10^{-6}$	$3,58 \cdot 10^{-6}$
Пилкоподібна	0,11	$7,85 \cdot 10^{-3}$	$4,39 \cdot 10^{-4}$	$2,40 \cdot 10^{-5}$	$1,68 \cdot 10^{-6}$

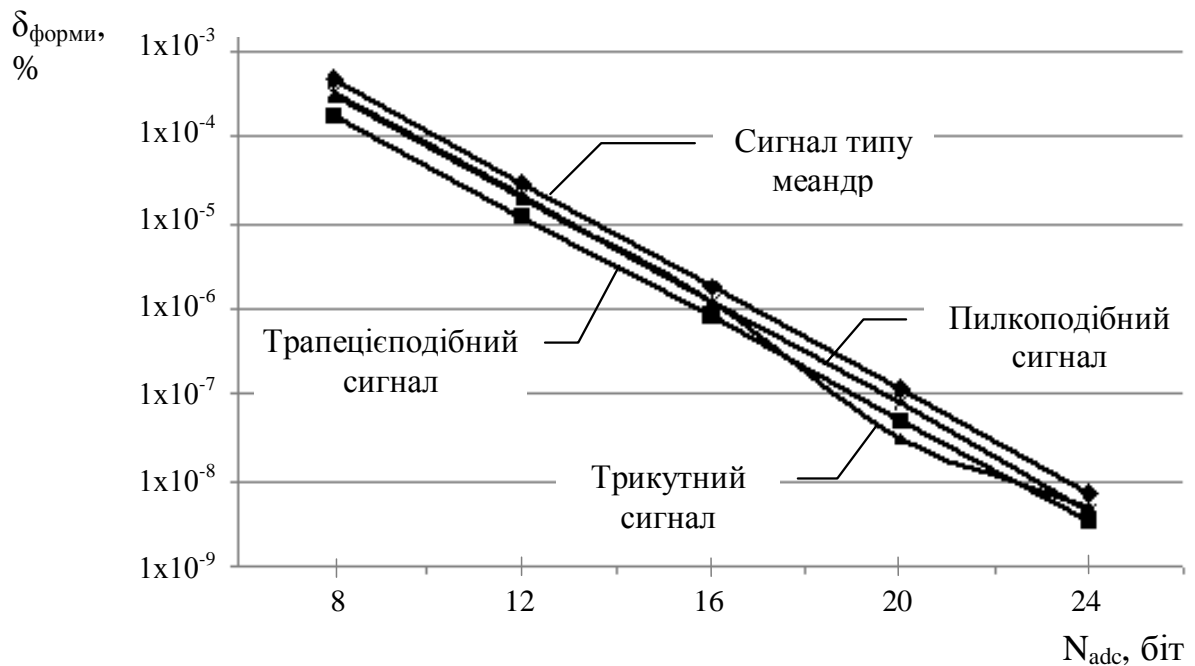


Рисунок 4.11 – Залежність середньої відносної похибки функції перетворення Фур'є від форми сигналу при зміні розрядності АЦП

На основі отриманих результатів (рисунок 4.11) можна сказати, що похибка функції $\text{fft}()$ залежить від форми сигналу. Найбільшу похибку отримано для сигналу типу меандр, найменшу – для трапецієподібної форми.

Крім того із збільшенням розрядності АЦП похибка функції $\text{fft}()$ для усіх сигналів зменшується. Так, наприклад для сигналу пілкоподібної форми для 8-розрядного АЦП середня відносна похибка функції $\text{fft}()$ становить $3,28 \cdot 10^{-4}$ %, а для 24-розрядного – $4,34 \cdot 10^{-9}$ %. Причому, максимальна похибка для 8-розрядного АЦП дорівнює 0,11 %, а для 24-розрядного – $1,68 \cdot 10^{-6}$ % (таблиця 4.3). Таким чином, отримані залежності дозволяють оптимізувати аналогову частину засобу вимірювання за розрядністю АЦП.

4.3 Дослідження похибки та непевності функції перетворення Фур'є програмних пакетів MATLAB та MathCAD з урахуванням похибок аналого-цифрового перетворювача

Як згадувалось вище, генератор зразкових цифрових послідовностей може використовуватися як для метрологічної перевірки, так і для відлагоджування програмного забезпечення. В режимі відлагоджування можуть використовуватися як попередньо описані можливості генератора зразкових цифрових послідовностей, так і додаткові, а саме, можливість формування зразкових цифрових послідовностей із врахуванням похибок аналогової частини ЗВ. Для аналізу результатів, отриманих під час відлагоджування програмного забезпечення, можна використовувати як підхід похибок, так і підхід непевностей.

Нижче представлено результати досліджень похибки функції $\text{fft}()$ із урахуванням адитивної, мультиплікативної, нелінійної, випадкової та сумарної похибок АЦП. Зразкові цифрові послідовності моделювалися із використанням розробленого генератора зразкових цифрових послідовностей.

Залежність відносної похибки функції $\text{fft}()$ від частоти сигналу та похибок АЦП для сигналу синусоїдальної форми. На рисунку 4.12

наведено результати дослідження залежності відносної похибки функції $\text{fft}()$ від частоти сигналу та похибок АЦП.

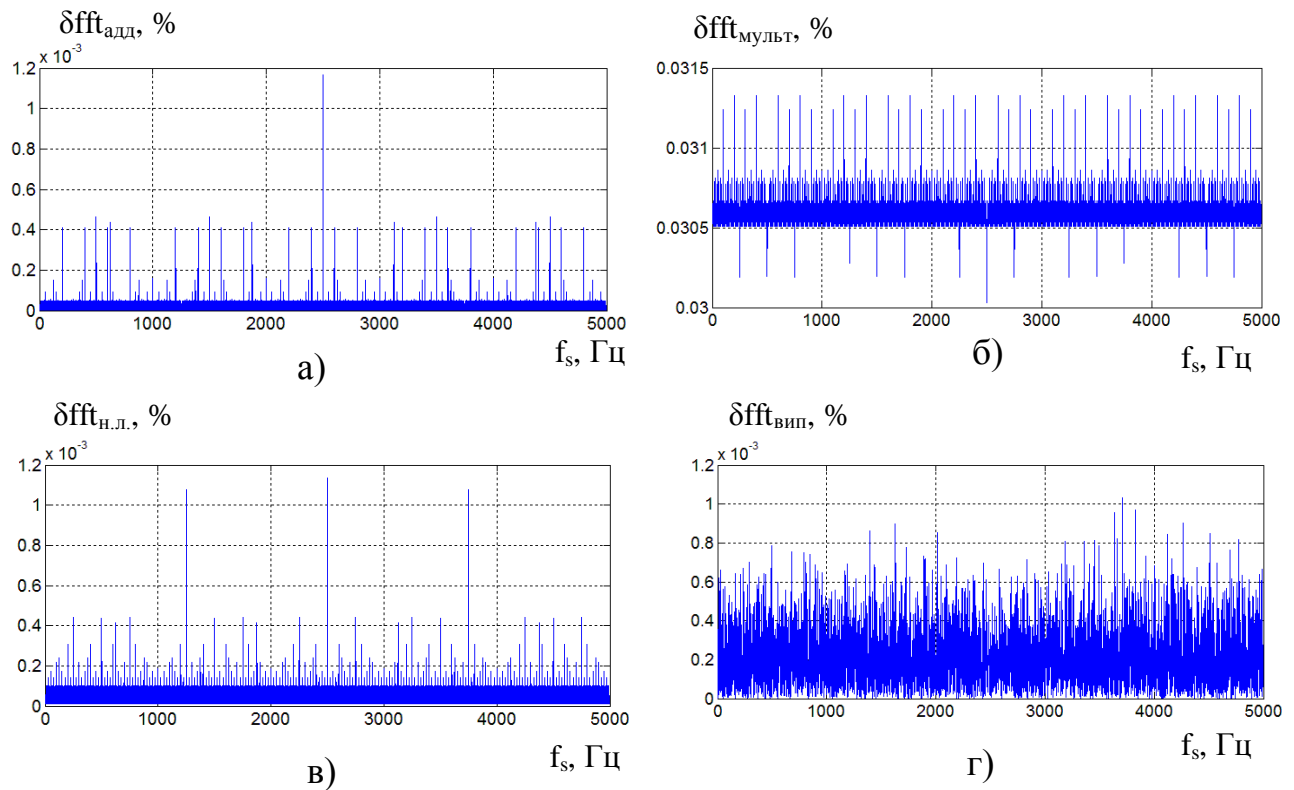


Рисунок 4.12 – Залежність відносної похибки функції $\text{fft}()$ від частоти сигналу та похибок АЦП: а) адитивної; б) мультиплікативної; в) нелінійної; г) випадкової

Дослідження проводилися за таких параметрів: $f_{\text{adc}} = 10000 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $N_{\text{adc}} = 16 \text{ біт}$; $U_{n_{\text{adc}}} = 10 \text{ В}$; $U_{i_{\text{adc}}} = 9,5 \text{ В}$; квантування до ближчого рівня; значення адитивної, мультиплікативної, нелінійної, випадкової похибок АЦП 0,03 % (абсолютна похибка 10 квантів). Формувався сигнал синусоїдальної форми із врахуванням похибок АЦП. Частота сигналу змінювалася від 1 до 4999 Гц. Проводилось перетворення Фур'є та обчислювалася похибка функції $\text{fft}()$ за формулою:

$$\delta \text{fft}_i = \frac{A_i - A_s}{A_i} \cdot 100\% , \quad (4.1)$$

де A_i – амплітуда i -тої гармоніки спектру; A_s – амплітуда сигналу.

На рисунку 4.13 представлено залежність відносної похибки результату обчислення гармонік функцією $\text{fft}()$ від сумарної похибки АЦП.

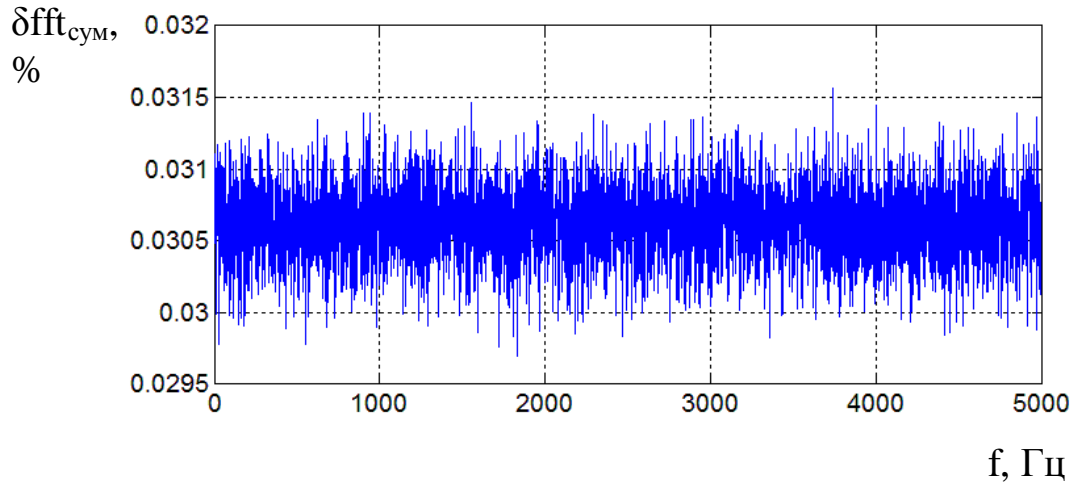


Рисунок 4.13 – Залежність відносної похибки функції $\text{fft}()$ від частоти сигналу та сумарної похибки АЦП

У таблиці 4.4 наведено результати дослідження впливу похибок АЦП на відносну похибку результату обчислення гармонік функцією $\text{fft}()$.

Таблиця 4.4 – Вплив похибок АЦП на похибку функції $\text{fft}()$

$\delta_{\text{АЦП}} / \delta \text{fft}$	Без похибок АЦП	Адитивна похибка	Мультиплікативна похибка	Нелінійна похибка	Випадкова похибка	Сумарна похибка
$\delta \text{fft}_{\text{макс}}$	$5,66 \cdot 10^{-4}$	$1,1 \cdot 10^{-3}$	$3,13 \cdot 10^{-2}$	$1,1 \cdot 10^{-3}$	$1,0 \cdot 10^{-3}$	$3,15 \cdot 10^{-2}$
$\delta \text{fft}_{\text{сер}}$	$3,0 \cdot 10^{-5}$	$3,3 \cdot 10^{-5}$	$3,06 \cdot 10^{-2}$	$3,23 \cdot 10^{-5}$	$2,12 \cdot 10^{-4}$	$3,06 \cdot 10^{-2}$

У другій колонці таблиці наведено максимальну та середню похибки функції $\text{fft}()$ без врахування похибок АЦП (рисунок 4.6).

Проведені дослідження показали, що найбільшу похибку функції $\text{fft}()$ отримано для мультиплікативної похибки АЦП. У порівнянні з похибкою функції $\text{fft}()$, що спричинена лише квантуванням сигналу (рисунок 4.6), максимальна похибка $\delta \text{fft}_{\text{макс}}$ з врахуванням мультиплікативної складової є на два порядки більшою; а для всіх інших складових – на один порядок.

На рисунку 4.14 представлено залежність відносної максимальної похибки функції перетворення Фур'є від значення адитивної, мультиплікативної, нелінійної та випадкової похибок АЦП.

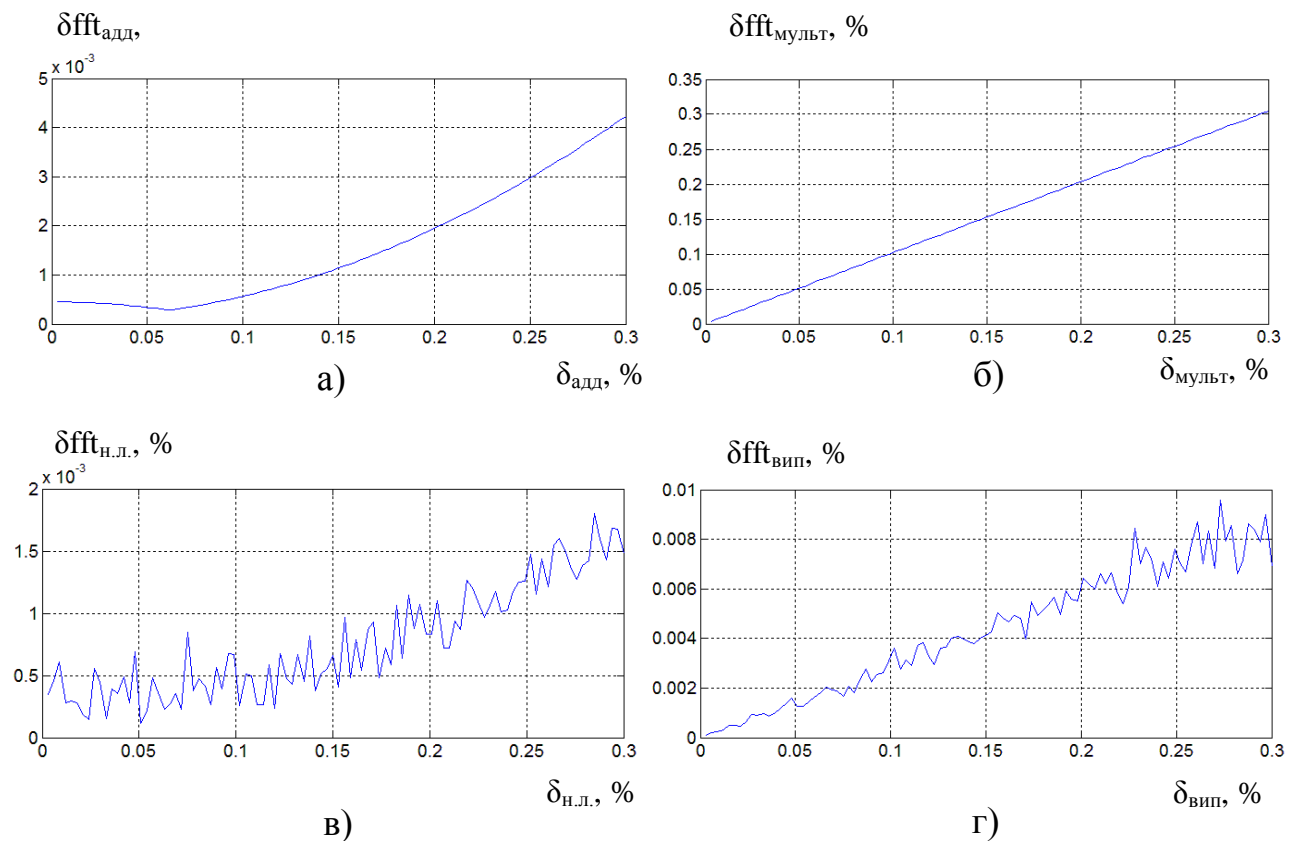


Рисунок 4.14 – Залежність максимальної відносної похибки функції $\text{fft}()$ від значення похибок АЦП: а) адитивної; б) мультиплікативної; в) нелінійної;

г) випадкової

Значення кожної складової похибки змінювалося від 0,003% до 0.3%, що відповідає значенню абсолютної похибки від 1 до 100 квантів. Для кожного значення похибки АЦП проводилося перетворення Фур'є для сигналів із частотами від 1 до 4999 Гц та обчислювалася похибка функції $\text{fft}()$. Із ряду отриманих похибок визначалося максимальне значення.

На рисунку 4.15 представлено залежність максимальної відносної похибки результату обчислення гармонік функцією $\text{fft}()$ від значення сумарної похибки АЦП.

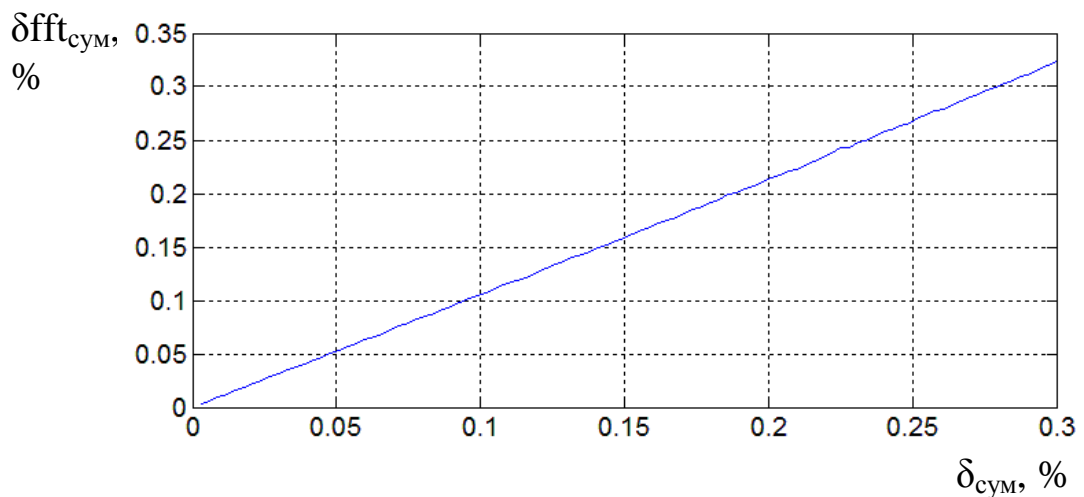


Рисунок 4.15 – Залежність максимальної відносної похибки функції $\text{fft}()$ від значення сумарної похибки АЦП

Проведені дослідження показали, що відносна похибка функції $\text{fft}()$ для сигналу синусоїдальної форми практично дорівнює відносній мультиплікативній похибці АЦП.

Залежність відносної похибки функції $\text{fft}()$ від похибок АЦП для сигналу трикутної форми. Дослідження проводилися за таких параметрів: $f_{\text{adc}} = 10000 \text{ Гц}$; $f_i = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $N_{\text{adc}} = 16 \text{ біт}$; $U_{n_{\text{adc}}} = 10 \text{ В}$; $U_{i_{\text{adc}}} = 9,5 \text{ В}$; квантування до ближчого рівня. Значення похибки

змінювалося від 0,003% до 0.3%, що відповідає значенню абсолютної похибки від 1 до 100 квантів.

На рисунку 4.16 представлено залежність максимальної відносної похибки функції перетворення Фур'є від значення адитивної, мультиплікативної, нелінійної та випадкової похибок АЦП.

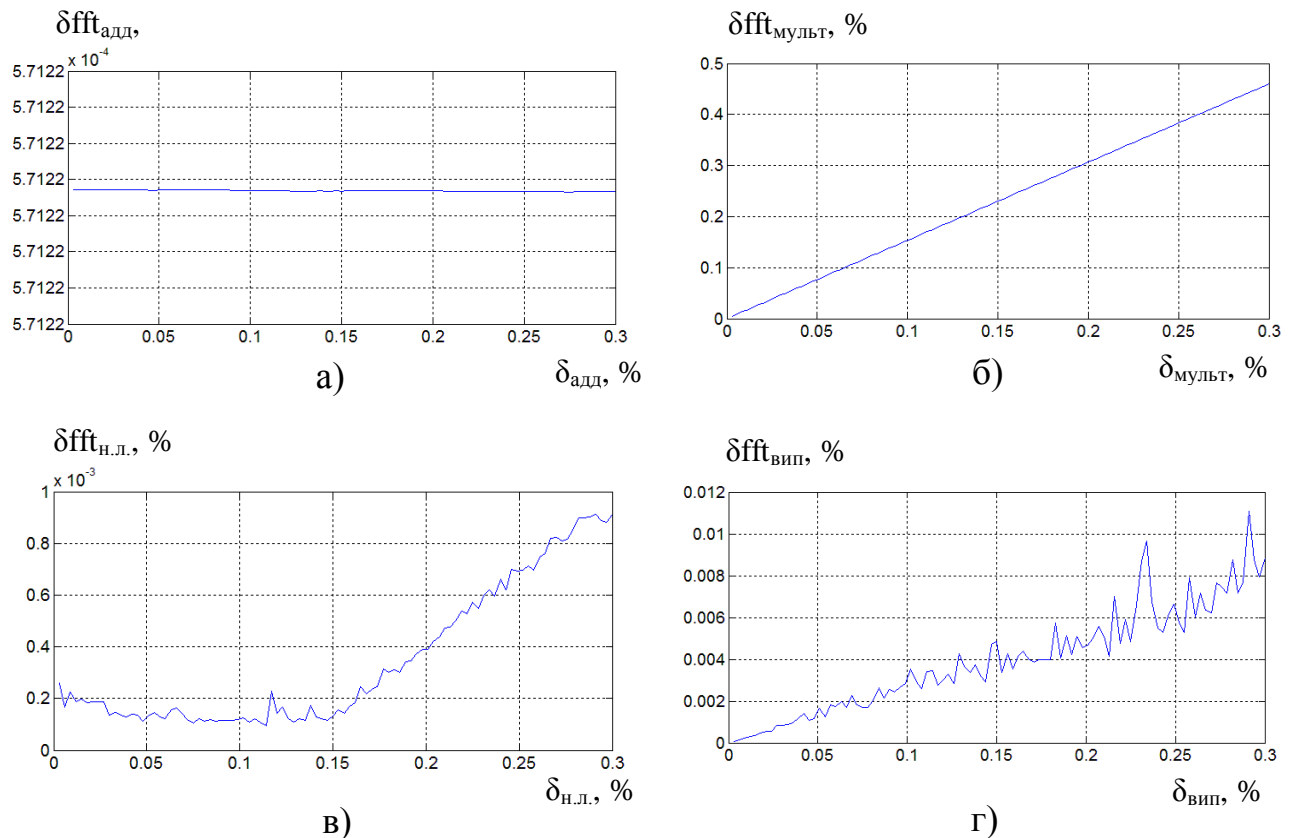


Рисунок 4.16. – Залежність максимальної відносної похибки функції $fft()$ від значення похибок АЦП: а) адитивної; б) мультиплікативної; в) нелінійної; г) випадкової для сигналу трикутної форми

Формувались зразкова цифрова послідовність для даних типу `int` із врахуванням похибок АЦП та зразкова цифрова послідовність для даних типу `double` без врахування похибок АЦП, проводилося перетворення Фур'є та обчислювалася похибка функції $fft()$ за формулою:

$$\delta \text{fft}_i = \frac{A1_i - A2_i}{A1_{10}} \cdot 100\% , \quad (4.2)$$

де $A1_i$ – амплітуда i -тої гармоніки спектру для даних типу `int`; $A2_i$ – амплітуда i -тої гармоніки спектру для даних типу `double`, $A1_{10}$ – амплітуда десятої гармоніки спектру сигналу типу `int` (основної гармоніки).

На рисунку 4.17 представлено залежність максимальної відносної похибки результату обчислення гармонік функцією `fft()` від значення сумарної похибки АЦП для сигналу трикутної форми.

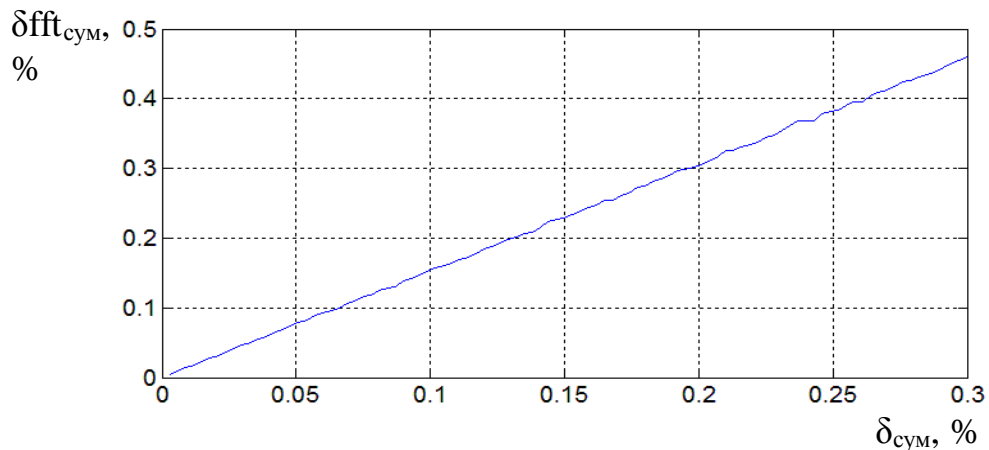


Рисунок 4.17 – Залежність максимальної відносної похибки функції `fft()` від значення сумарної похибки АЦП для сигналу трикутної форми

Проведені дослідження показали, що відносна похибка функції `fft()` для сумарної похибки АЦП практично дорівнює відносній похибці функції `fft()` для мультиплікативної похибки АЦП. Тому подальші дослідження проводилися лише для мультиплікативної складової похибки АЦП.

Залежність похибки функції `fft()` від розрядності АЦП із врахуванням мультиплікативної похибки АЦП. Дослідження проводилося за таких параметрів: $f_{\text{adc}} = 100 \text{ кГц}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$;

$U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 9,9 \text{ В}$; сигнал синусоїдальної форми; квантування до ближчого рівня, розрядність АЦП N_{adc} змінювалася від 8 до 24 біт, значення мультиплікативної похибки становили 0,003, 0,03, та 0,3% (рисунок 4.18).

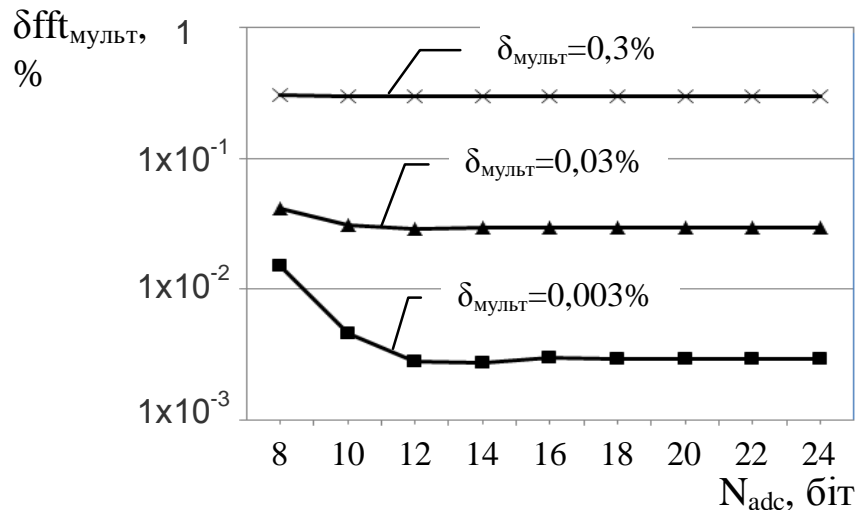


Рисунок 4.18 – Залежність відносної похибки функції $fft()$ від зміни мультиплікативної складової похибки аналогової частини ЗВ при зміні розрядності АЦП

Із зменшенням розрядності АЦП похибка функції $fft()$ зростає. Так, наприклад для $\delta_{мульти} = 0,003\%$ (рисунок 4.18) для $N_{adc}=24$ біт відносна похибка функції $fft()$ $\delta_{fft} = 2,97 \cdot 10^{-3}\%$; для $N_{adc}=16$ біт – $\delta_{fft} = 2,99 \cdot 10^{-3}\%$; для $N_{adc}=8$ біт – $\delta_{fft} = 1,52 \cdot 10^{-2}\%$. Для мультиплікативної похибки АЦП $\delta_{мульти}=0,3\%$ і більше похибка функції $fft()$ практично не змінює свого значення. Похибка квантування у цьому випадку практично не впливає на похибку функції $fft()$, а домінуючим є вплив мультиплікативної похибки АЦП.

Для значень мультиплікативної похибки 0,003, 0,03, та 0,3%, у порівнянні із результатами досліджень, де враховувалося лише квантування сигналу (рисунок 4.10), для АЦП із розрядністю 20 - 24 біт похибка функції

fft() зростає на 5 - 7 порядків. Із зменшенням розрядності та значення мультиплікативної похибки АЦП домінуючим стає вплив дискретизації на похибку функції fft().

Залежність похибки функції fft() від мультиплікативної похибки АЦП при зміні частоти дискретизації АЦП. Дослідження проводилося за таких параметрів: $N_{adc} = 16 \text{ bit}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 9,9 \text{ В}$; сигнал синусоїдальної форми; квантування до ближчого рівня; частота дискретизації АЦП f_{adc} змінювалася від 100 Гц до 100 кГц, значення мультиплікативної похибки становили 0,003, 0,03, та 0,3%. Результати досліджень подано на рисунку 4.19.

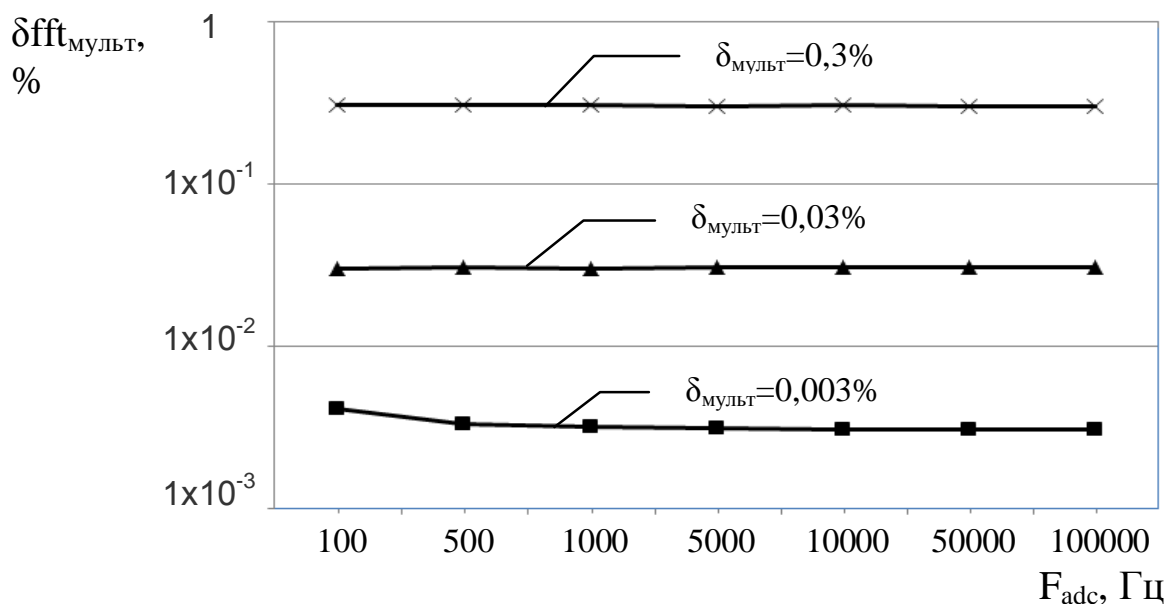


Рисунок 4.19 – Залежність похибки функції fft() від мультиплікативної похибки АЦП при зміні частоти дискретизації АЦП

Із зменшенням частоти дискретизації АЦП похибка функції fft() зростає. Для мультиплікативної похибки АЦП $\delta_{\text{мульти}} = 0,03\%$ і більше похибка функції fft() практично не змінює свого значення.

Для значень мультиплікативної похибки 0,003, 0,03, та 0,3%, у порівнянні із результатами досліджень, де враховувалося лише квантування сигналу (рисунок 4.9), похибка функції $\text{fft}()$ зростає на 1 - 3 порядки.

Залежність похибки функції $\text{fft}()$ від мультиплікативної похибки АЦП при зміні фази сигналу. Дослідження проводилося за таких параметрів: $N_{adc} = 16 \text{ bit}$; $f_{adc} = 100 \text{ кГц}$; $t = 1 \text{ с}$; $U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 9,9 \text{ В}$; $f_s = 10 \text{ Гц}$; сигнал синусоїдальної форми, квантування до ближчого рівня, фаза сигналу φ_s змінюється від 0 до 360°, значення мультиплікативної похибки 0,03%. На рисунку 4.20 представлено залежність відносної похибки функції перетворення Фур'є від мультиплікативної похибки при зміні фази сигналу.

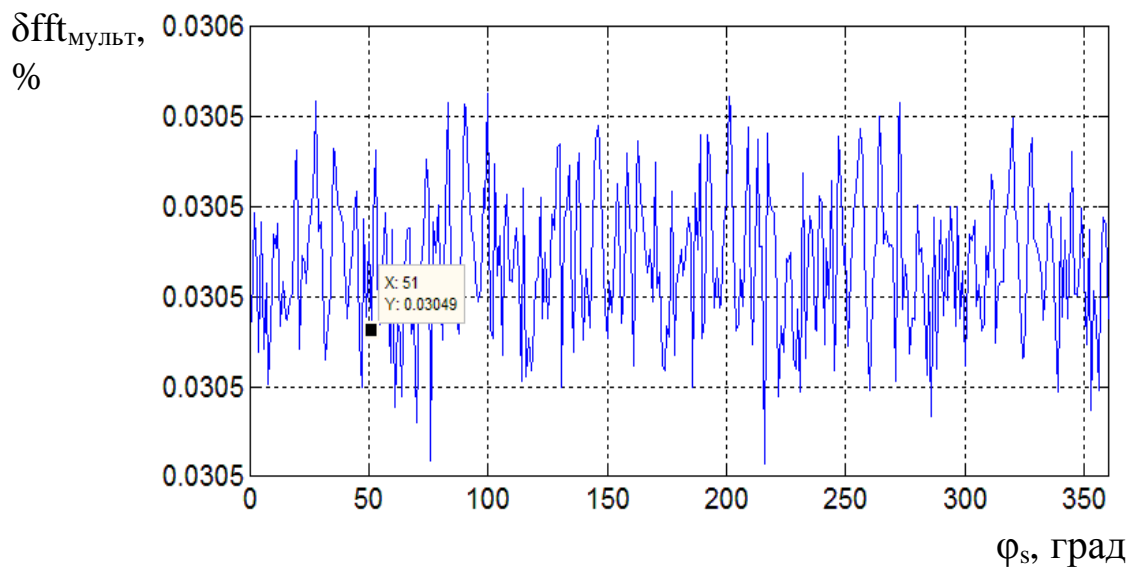


Рисунок 4.20 – Залежність відносної похибки функції перетворення Фур'є від мультиплікативної складової похибки 0,03% при зміні фази сигналу

Для значення мультиплікативної похибки 0,03 %, у порівнянні із результатами досліджень, де враховувалося лише квантування сигналу (рисунок 4.7), похибка функції $\text{fft}()$ зростає на 2 порядки.

На основі проведених досліджень можна сказати, що похибки АЦП суттєво впливають на похибку функції $\text{fft}()$.

Генератор зразкових цифрових послідовностей забезпечує також можливість оцінки непевності результатів обчислень програмним забезпеченням. Нижче наведено приклад оцінки непевності результату обчислень функції $\text{fft}()$ для сигналу синусоїдальної форми. Розраховувалася розширена непевність функції $\text{fft}()$ для довірчої ймовірності $P = 0,99$ (коефіцієнт Стюдента $t_{st} = 2,785$). Під час дослідження враховувались адитивна, мультиплікативна, нелінійна та випадкова складові похибки АЦП. Зразкові цифрові послідовності моделювалися із використанням розробленого генератора зразкових цифрових послідовностей. Генерувалося 1000 зразкових цифрових послідовностей із урахуванням похибок АЦП. Для кожної послідовності значення похибки (адитивної, мультиплікативної, нелінійної або випадкової, а також сумарної) змінювалося випадковим чином у заданому діапазоні. Дослідження проводилося за таких параметрів: $f_{adc} = 1000 \text{ Гц}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $N_{adc} = 16 \text{ біт}$; $U_{n_adc} = 10 \text{ В}$; $U_{i_adc} = 10 \text{ В}$; квантування до ближчого рівня; максимальне значення похибки змінювалося від 0,003 % (абсолютна похибка 1 квант) до 0,03 % (абсолютна похибка 10 квантів). Наприклад, для максимального значення похибки 0,003 % генерувалися випадкові значення у діапазоні від $-0,003\%$ до $+0,003\%$ і т.д.

У таблиці 4.5 представлено результати дослідження впливу складових похибки АЦП на відносну розширену непевність результату обчислення гармонік функцією $\text{fft}()$.

На рисунку 4.21 представлено залежність відносної розширеної непевності функції перетворення Фур'є від адитивної (рисунок 4.21, а), мультиплікативної (рисунок 4.21, б), нелінійної (рисунок 4.21, в) та випадкової (рисунок 4.21, г) складових похибки АЦП. На рисунку 4.22 представлено залежність відносної непевності функції перетворення Фур'є від сумарної похибки АЦП.

Таблиця 4.5 – Порівняльна характеристика впливу похибок АЦП відносно розширену непевність функції перетворення Фур'є у MATLAB

Тип похибки	Максимальна кількість квантів випадкової похибки		
	1	5	10
Адитивна	$9,4 \cdot 10^{-6}$	$1,2 \cdot 10^{-5}$	$1,8 \cdot 10^{-5}$
Мультиплікативна	$1,5 \cdot 10^{-4}$	$7,5 \cdot 10^{-4}$	$1,5 \cdot 10^{-3}$
Нелінійна	$7,4 \cdot 10^{-6}$	$9,8 \cdot 10^{-6}$	$1,3 \cdot 10^{-5}$
Випадкова	$7,4 \cdot 10^{-6}$	$3,3 \cdot 10^{-5}$	$6,8 \cdot 10^{-5}$
Сумарна	$1,6 \cdot 10^{-4}$	$7,6 \cdot 10^{-4}$	$1,5 \cdot 10^{-3}$

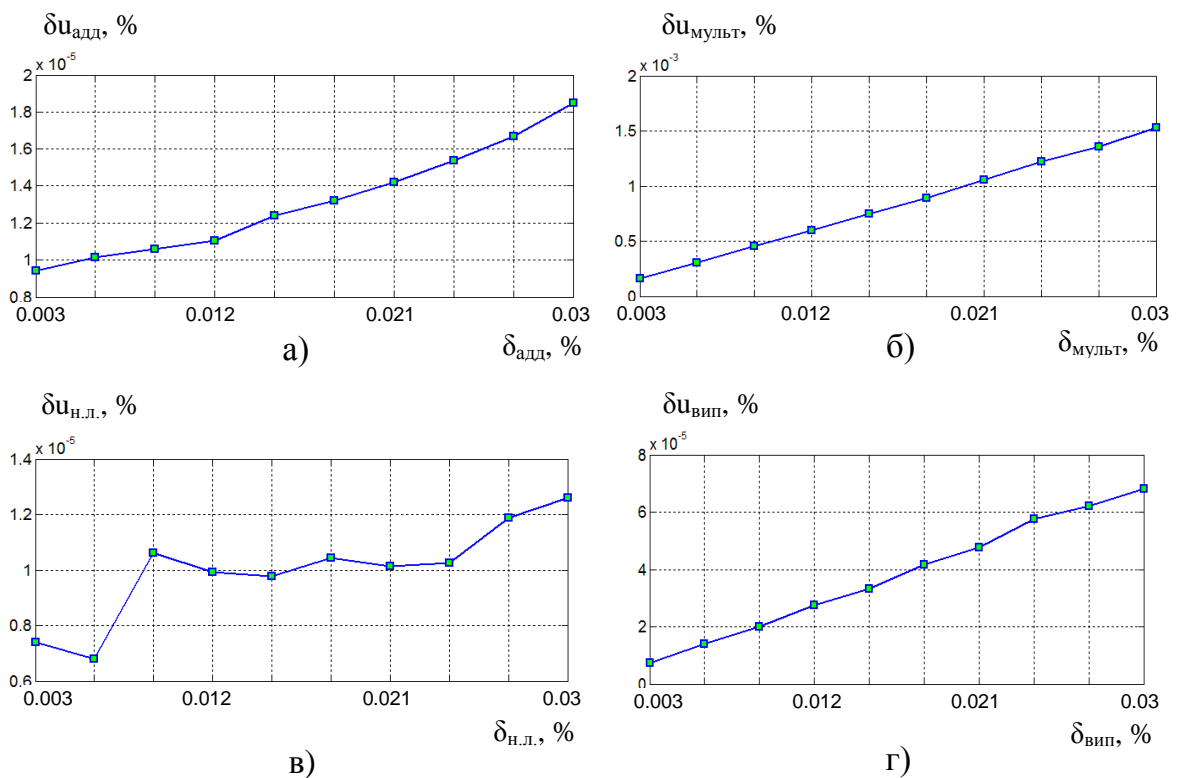


Рисунок 4.21 – Залежність відносної розширеної непевності функції $\text{fft}()$ від похибок АЦП: а) адитивної; б) мультиплікативної; в) нелінійної; г) випадкової

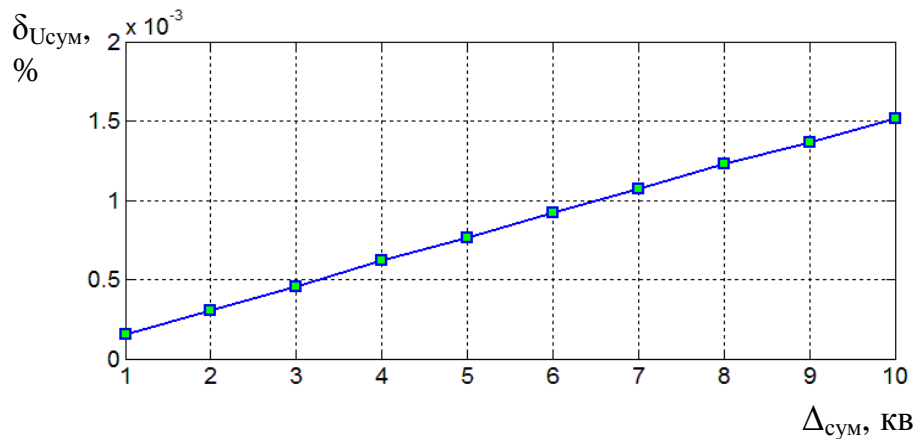


Рисунок 4.22 – Залежність відносної розширеної непевності функції FFT від сумарної похибки аналогової частини ЗВ

Проведені дослідження показали, що відносна розширена непевність результату обчислення гармонік функцією $\text{fft}()$ для сумарної похибки АЦП практично дорівнює відносній розширеній непевності функції $\text{fft}()$ для мультиплікативної похибки АЦП.

Дослідження непевностей функції перетворення Фур'є програмного пакету Mathcad. У програмному пакеті Mathcad проведено дослідження залежності непевності функції $\text{fft}()$ від похибок АЦП. Генерувалося 1000 зразкових цифрових послідовностей. Для кожної послідовності значення похибки змінювалося випадковим чином у заданому діапазоні. Дослідження проводилося за таких параметрів: $f_{\text{adc}} = 1024 \text{ Гц}$; $f_s = 10 \text{ Гц}$; $t = 1 \text{ с}$; $\varphi_s = 0^\circ$; $N_{\text{adc}} = 16 \text{ біт}$; $U_{n_{\text{adc}}} = 10 \text{ В}$; $U_{i_{\text{adc}}} = 10 \text{ В}$; сигнал синусоїдальної форми; квантування до ближчого рівня; сумарна похибка АЦП становить 0,003, 0,009 та 0,015 % (абсолютна похибка 1, 3 та 5 квантів). Результати досліджень подано у таблиці 4.6.

За результатами досліджень (таблиця 4.6), непевності функції перетворення Фур'є для програмних пакетів Mathcad та MATLAB є практично однаковими.

Таблиця 4.6 – Залежність відносної непевності функції перетворення Фур'є від сумарної похибки АЦП для програмних пакетів Mathcad та MATLAB

	Максимальна кількість квантів випадкової сумарної похибки		
	1	3	5
Mathcad	$1,66 \cdot 10^{-4}$	$5,14 \cdot 10^{-4}$	$8,45 \cdot 10^{-4}$
MATLAB	$1,84 \cdot 10^{-4}$	$4,99 \cdot 10^{-4}$	$9,11 \cdot 10^{-4}$

Висновки до розділу 4

1. Із використанням розробленого генератора зразкових цифрових послідовностей проведено метрологічну перевірку функції перетворення Фур'є програмних пакетів MATLAB і Mathcad.
2. За результатами метрологічної перевірки функції fft перетворення Фур'є з використанням даних типу double похибка обчислення гармонік спектру не перевищує $10^{-10}\%$.
3. При використанні зразкових цифрових послідовностей із урахуванням квантування сигналу похибка функції перетворення Фур'є суттєво зростає та для 16-розрядного АЦП не перевищує $6 \cdot 10^{-4}\%$. Квантування сигналу призвело до зростання похибки на 6 порядків.
4. При використанні зразкових цифрових послідовностей із урахуванням квантування сигналу та сумарної похибки АЦП від 0,003% до 0,3%, похибка функції перетворення Фур'є для 16-розрядного АЦП не перевищує відповідно 0,0033% та 0,33% для синусоїдального сигналу, а для трикутного сигналу – 0,0047% та 0,47%.
5. Проведені дослідження показали, що на похибку програмного забезпечення засобів вимірювання суттєвий вплив має дискретизація сигналу.

ВИСНОВКИ

У дисертаційній роботі наведено нове розв'язання актуального науково-технічного завдання, яке полягає у розробленні генератора зразкових цифрових послідовностей для метрологічної перевірки програмного забезпечення засобів вимірювання, а саме:

1. Розроблено алгоритми метрологічної перевірки програмного забезпечення інтелектуальних первинних перетворювачів, засобів вимірювання та кібер-фізичних систем.
2. Запропоновано багаторівневу метрологічну перевірку програмного забезпечення компонентів кібер-фізичних систем, що дозволяє покращити безпеку функціонування кібер-фізичних систем та полегшити процес метрологічної перевірки їх програмного забезпечення.
3. Розроблено генератор зразкових цифрових послідовностей, зокрема його математичну модель та алгоритм роботи, що дозволяє формувати зразкові цифрові послідовності з врахуванням параметрів вимірюваного сигналу та параметрів АЦП.
4. Генератор зразкових цифрових послідовностей можна інтегрувати у будь-який програмний пакет сучасних засобів вимірювання (типу MATLAB, LabVIEW тощо) як окремий блок для метрологічної перевірки програмного забезпечення, а також як засіб відлагоджування програмного забезпечення при розробці засобу вимірювання.
5. Із використанням розробленого генератора зразкових цифрових послідовностей проведено метрологічну перевірку функції перетворення Фур'є програмних пакетів MATLAB і Mathcad.

6. За результатами метрологічної перевірки функції перетворення Фур'є з використанням даних типу double похибка обчислення гармонік спектру не перевищує $10^{-10}\%$.
7. При використанні зразкових цифрових послідовностей із урахуванням квантування сигналу похибка функції `fft()` суттєво зростає та для 16-розрядного АЦП не перевищує $6 \cdot 10^{-4}\%$. Квантування сигналу призвело до зростання похибки на 6 порядків.
8. При використанні зразкових цифрових послідовностей із урахуванням квантування сигналу та сумарної похибки АЦП від 0,003% до 0.3%, похибка функції `fft()` для 16-розрядного АЦП не перевищує відповідно 0,0033% та 0,33% для синусоїдального сигналу, а для трикутного сигналу – 0,0047% та 0,47%.
9. Проведені дослідження показали, що на похибку програмного забезпечення засобів вимірювання суттєвий вплив має дискретизація сигналу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Олесків О.М. Аналітичний огляд процедур та методів метрологічної перевірки програмного забезпечення засобів вимірювання / О.М. Олесків, І.П. Микитин, І.Я. Кунець // Вимірювальна техніка та метрологія. – 2014. – №75. – С. 19-23.
2. Олесків О. Метрологічна перевірка програмного забезпечення засобів вимірювання з різними структурами / О. Олесків, І. Микитин // Комп'ютерні науки та інформаційні технології. – 2015. – № 826. – С. 325-330.
3. Олесків О. М. Особливості метрологічної перевірки компонентів кібер-фізичних систем / О.М. Олесків, І.П. Микитин // Науковий вісник НЛТУ України. – 2016. – Вип. 26.5. – С. 340-344.
4. Oleskiv O. Metrological verification of cyber-physical systems / O. Oleskiv, I. Mykutyin // Zeszyty naukowe Politechniki Rzeszowskiej. Elektrotechnika. – 2015. – №34. – P. 87-92.
5. Порядок атестації програмного забезпечення засобів вимірювальної техніки [Електронний ресурс]: Інструкція. Метрологія. // Держспоживстандарт. – 2006. – №32. – 11 с. – Режим доступу до інформації: <http://metrology.com.ua/normativnaya-baza/drugoe/attestatsiya-po-sit>.
6. Development of Software Requirements [Електронний ресурс]: Informative Document. – WELMEC 7.1. – Issue 2. – 2005. – 48 p. – Режим доступу до інформації: http://www.vniims.ru/009lab/docs/welmec_7_1.pdf.
7. Software Guide (Measuring Instruments Directive 2004/22/EC). 2011. [Електронний ресурс]: WELMEC 7.2. – Issue 5. – 2012. – 120 p. – Режим доступу до інформації: http://www.welmec.org/fileadmin/user_files/publications/WELMEC_07.02_Issue5_SW_2012-03-19.pdf.

8. Guide for Examining Software (Non-automatic Weighing Instruments) [Електронний ресурс]: WELMEC 2.3. – Issue 3. – 2005. – 23 p. – Режим доступу до інформації:

http://www.welmec.org/fileadmin/user_files/publications/2-3.pdf.

9. Проверка защиты программного обеспечения определение ее уровня при испытаниях средств измерений в целях утверждения типа: МИ 3286 – 2010. – ФГУП «ВНИИМС», 2010. – 33 с. – (Рекомендация).

10. Лукашов Ю.Е. Классификация программного обеспечения средств измерений [Електронний ресурс] / Ю.Е. Лукашов, А.А. Сатановский // Режим доступу до інформації: http://www.vniims.ru/download/classification_soft.pdf.

11. Виды программ и программных документов: ГОСТ 19.101-77. – [Дата введения 1980-01-01]. – М.: «Стандартинформ», 2010. – 4 с. (Государственный стандарт союза ССР).

12. Обозначение программ и программных документов: ГОСТ 19.103-77. – [Дата введения 1980-01-01]. – М.: «Стандартинформ», 2010. – 4 с. (Государственный стандарт союза ССР).

13. Общие требования к программным документам: ГОСТ 19.105-78. – [Дата введения 1980-01-01]. – М.: «Стандартинформ», 2010. – 4 с. (Государственный стандарт союза ССР).

14. Руководство по программному обеспечению (основано на Директиве по измерительным приборам 2004/22/ЕС) [Електронний ресурс]: ВЭЛМЭК 7.2. – Издание 1. – 2009. – 130 с. – Режим доступу до інформації: http://009lab.vniims.ru/docs/welmec_7_2_ru.pdf.

15. Тарасенко М. Г. Віртуальний вимірювальний комплекс для дослідження перехідних процесів в електричних колах довільної конфігурації. [Електронний ресурс] / Тарасенко М. Г., Коваль В. П. // Реєстрація, зберігання і обробка даних. – 2006. – Т.8. – №1. – С. 84-92. – Режим доступу до інформації: <http://dSPACE.nbu.gov.ua/bitstream/handle/123456789/50831/09-Tarasenko.pdf>.

16. Измерительные системы. [Электронный ресурс] / Режим доступа до інформації: <http://toe-kgeu.ru/mcc/91-metrology>.
17. Стухляк П. Д. Теорія інформації (інформаційно-вимірювальні системи, похибки, ідентифікація): навчальний посібник / П. Д. Стухляк, О. В. Іванченко, А. В. Букетов, М. А. Долгов. – Херсон: Айлант, 2011. – 371 с.
18. Ненов О. Л. Проектування вбудованих комп'ютерних систем: навчальний посібник / О. Л. Ненов. – Одеса, 2008 – 89 с.
19. Technologie-Konzept. [Електронний ресурс]. – Режим доступа до інформації: <http://www.its-owl.de/technologie-netzwerk/strategie/technologie-konzept/>.
20. Jiafu Wan. Advances in Cyber-Physical Systems Research / Jiafu Wan, Nehua Yan, Hui Suo, Fang Li // KSII Transactions On Internet And Information Systems. – 2011. – Vol. 5. – № 11. – P. 1891-1908.
21. Cyber-physische Systeme. [Електронний ресурс]. – Режим доступа до інформації: <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/informationssysteme/Sektorspezifische-Anwendungssysteme/cyber-physische-systeme>.
22. Cyber-Physical Systems Security for the Smart Grid. [Електронний ресурс]. – Режим доступа до інформації: http://csrc.nist.gov/news_events/cps-workshop/cps-workshop_abstract-8_cardenas.pdf.
23. Catsoulis J. Designing embedded hardware. [Електронний ресурс] / J. Catsoulis. – O'Reilly, 2005. – 400 p. – Режим доступа до інформації: <http://www.maerivoet.org/website/software/arduino/manuals/electronics/designing-embedded-hardware.pdf>.
24. Метрологія та вимірювальна техніка. / [Є. С. Поліщук, М.М. Дорожовець, В. О. Яцук та ін.]. – Л. : Бескид Біт, 2003. – 544 с.
25. Олесків О.М. Багаторівнева метрологічна перевірка кібер-фізичних систем. / О.М. Олесків // Вимірювальна техніка та метрологія. – 2015. – №76. – С. 117-124.

26. Олесків О.М. Особливості метрологічної перевірки програмного забезпечення засобів вимірювання з різними структурами / О. М. Олесків, І. П. Микитин, Т. Фрьогліх // Всеукраїнська науково-технічна конференція молодих вчених у царині метрології «Technical Using of Measurement-2015»: – Тези доповідей. – Славське, 2-6 лютого 2015 р.– С. 83-85.
27. Кветний Р. Н. Інтервальні моделі перетворень сигналів в інформаційно-вимірювальних системах / Кветний Р. Н., Бойко О. Р. – Вінниця : УНІВЕРСУМ-Вінниця, 2009. – 88с.
28. Кричевець О. Метрологічні моделі обчислювальних компонентів вимірювальних інформаційних систем / Олександр Кричевець // Вимірювальна техніка та метрологія. – 2008. – № 68. – С. 220-223.
29. Кондратов В. Т. Определения и базовая классификация измерительных систем / В. Т. Кондратов // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2014. – №3. – С. 85-100.
30. Уилмсхерст Т. Разработка встроенных систем с помощью микроконтроллеров PIC / Т. Уилмсхерст. – М.: МК-Пресс. 2008. – 544 с.
31. Arnold S. Berger. Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. [Електронний ресурс] / Arnold S. Berger. CMP Books 2002. – 237 р. – Режим доступу до інформації: <http://dsp-book.narod.ru/ESDIPTT.pdf>.
32. Уилмсхерст Т. Разработка встроенных систем с помощью микроконтроллеров PIC / Уилмсхерст Т. – М.: МК-Пресс, 2008 – 544 с.
33. Мельник А. О. Кіберфізичні системи: проблеми створення та напрями розвитку / А. О. Мельник // Комп'ютерні системи та мережі. – 2014. – №806. – С. 154-161.
34. Edward A. Lee. Cyber Physical Systems: Design Challenges [Електронний ресурс] / Edward A. Lee // Режим доступу до інформації: http://www.cs.virginia.edu/~son/cs851/papers/CPS.challenges_ISORC08.pdf.

35. A. Rajhans. An Architectural Approach to the Design and Analysis of Cyber-Physical Systems [Электронный ресурс] / A. Rajhans, S.W. Cheng, B. Schmerl, B.H. Krogh, C. Aghi, and A. Bhave // Electronic Communications of the EASST. – 2009. – Vol.21. – Режим доступа до інформації: https://arajhans.github.io/files/papers/RajhansCS+_MPM09.pdf.
36. Mueller W. Virtual prototyping of cyberphysical systems [Электронный ресурс] / Mueller W., Becker M., Elfeky A., DiPasquale A. // Режим доступа до інформації: <https://pdfs.semanticscholar.org/9482/edd81ba96fb1c4a303f7e2fa460e6d10c4ce.pdf>.
37. R. Rajkumar. Cyber-physical systems: the next computing revolution [Электронный ресурс] / R. Rajkumar, I. Lee, L. Sha, J. Stankovic // Design Automation Conference. ACM, 2010. – P. 731-736. – Режим доступа до інформації: <https://www.cs.virginia.edu/~stankovic/psfiles/Rajkumar-DAC2010-Final.pdf>.
38. Software Flexibility [Электронный ресурс] / Режим доступа до інформації: <https://informativearchitecture.wordpress.com/2011/10/04/software-flexibility/>.
39. What Makes Software Flexible [Электронный ресурс] / Режим доступа до інформації: <http://wiki.c2.com/?WhatMakesSoftwareFlexible>.
40. The rising need for software flexibility (with example) [Электронный ресурс] / Режим доступа до інформації: <http://www.mrc-productivity.com/blog/2013/09/the-rising-need-for-software-flexibility-with-example/>.
41. Злыгостева Г.В. Обобщенная модель процедуры испытаний измерительного программного обеспечения / Г.В. Злыгостева, С.В. Муравьев // Известия Томского политехнического университета. – 2011. – Т. 318. – № 4. – С 62-67.

42. Кудеяров Ю.А. Аттестация программного обеспечения средств измерений / Кудеяров Ю.А. – М.: АНО «РСК-Консалтинг», 2006. – 97 с.
43. Кудеяров Ю.А. Аттестация программного обеспечения средств измерений / Ю.А. Кудеяров, О.В. Стефанов, А.Н. Паньков // Компетентность. – 2009. – №3. – С. 22-28.
44. Слаев В.А. Аттестация программного обеспечения, используемого в метрологии: [справочная книга]. / В.А. Слаев, А.Г. Чуновкина // СПб.: НПО «Профессионал», 2009. – 320 с.
45. Аттестация алгоритмов и программ обработки данных при измерениях. Основные положения: МИ 2174_91. [Дата введения – 01.01.1992].– М.:Изд_во стандартов, 1993. – III, 27 с. – (Рекомендация).
46. Ковалевская Е.В. Метрология, качество и сертификация программного обеспечения: [учебная программа, руководство по изучению дисциплины, учебное пособие, практикум по курсу, тестовые задания по дисциплине] / Е.В. Ковалевская //Москва, 2004. – 96 с.
47. Программное обеспечение средств измерений. Общие технические требования: КОOMET R/LM/10:2004 – Албена, Болгария, 2004 г. – 10 с. – (Рекомендация).
48. Directive 2004/22/EC of the European Parliament and of the Council of 31 March 2004 on measuring instruments [Электронный ресурс]: / Official Journal of the European Union. – 31.4.2004. – 80 p. – Режим доступа до інформації: http://www.lne.eu/publications_en/directives/04-22e.pdf.
49. Триль Ф. Актуальные проблемы законодательной метрологии в области встроенного программного обеспечения средств измерений [Электронный ресурс] / Ф.Триль, У. Гротткер, Д.Рихтер // Законодательная и прикладная метрология. 2011. - № 3. -С. 13-22. – Режим доступа до інформації: http://www.vniims.ru/download/pub/2011/2011_3_1.pdf.
50. Державні випробування засобів вимірювальної техніки. Основні положення, організація, порядок проведення і розгляду результатів: ДСТУ

- 3400:2006. – [Чинний від 2006-12-27]. – К.: Держспоживстандарт України, 2007. – III, 39 с. – (Національний стандарт України).
51. Требования к программному обеспечению средств измерений. Основные положения: ГОСТ Р 8.654-2009. – [Дата введения 2010-01-01]. – М.: Стандартиформ, 2009. – III, 19 с. – (Национальный стандарт Российской Федерации).
52. Standard Glossary of Software Engineering Terminology: IEEE 610.12-1990. – [Approved September 28,1990]. – NY.: IEEE Standards Board, 2009. – III, 83 p.
53. Кудеяров Ю.А. Требования к программному обеспечению средств измерений на основе зарубежных и отечественных стандартов и рекомендаций / Ю.А. Кудеяров, Ю.Е. Лукашов, А.А. Сатановский // Законодательная и прикладная метрология. – 2003. – № 5. - С. 32-41.
54. Кудеяров Ю.А. Метрологическая аттестация программного обеспечения средств измерений (Состояние и перспективы) / Ю.А. Кудеяров, Ю.Е. Лукашов, А.А. Сатановский // Законодательная и прикладная метрология. – 2003. – №4. – С. 39-14.
55. Кудеяров Ю.А. Метрологическая экспертиза программного обеспечения средств измерений / Ю.А. Кудеяров. – М.: АНО «РСК-Консалтинг», 2008. - 97 с.
56. Величко О.Н. Нормативная база аттестации программного обеспечения средств измерений / О.Н. Величко // Измерительная техника. – 2007 – № 4. – С. 12-17.
57. Злыгостева Г.В. Современные подходы к аттестации программного обеспечения средств измерений / Г.В. Злыгостева, М.М. Чухланцева // Измерительная техника. – 2007. – №5. – С. 65-68.
58. Методика метрологической аттестации алгоритмов и программ обработки данных результатов измерений при определении объема и массы

нефти и нефтепродуктов. Общие положения: МИ 2676-2001. – Казань, 2001. – 12 с. – (Рекомендация. ГСИ).

59. General requirements for software controlled measuring instruments. [Электронный ресурс] / OIML D 31. – Edition 2008 (E). – 2008. – 53 p. – Режим доступа до інформації: http://www.fundmetrology.ru/depository/04_IntDoc_all/D031-e08.pdf.

60. Программное обеспечение встроенных систем. Общие требования к разработке и документированию: ГОСТ Р 51904-2002. – [Дата введения 2003-07-01]. – М.: «Стандартинформ», 2005. – 67 с. – (Национальный стандарт Российской Федерации).

61. Информационная технология. Руководство по управлению документированием программного обеспечения: ГОСТ Р 9294-93. – [Дата введения 1993-12-20]. – М.: «Стандартинформ», 2008. – 11 с. (Государственный стандарт Российской Федерации).

62. Оценка качества программных средств. Общие положения: ГОСТ 28195-89. – [Дата введения 1990-07-01]. – М.: Издательство стандартов, 2001. – 31 с. (Межгосударственный стандарт).

63. Информационная технология. Пакеты программ. Требования к качеству и тестирование: ГОСТ Р ИСО/МЭК 12119-2000. – [Дата введения 2002-01-01]. М.: «Стандартинформ», 2006. – 19 с. (Государственный стандарт Российской Федерации).

64. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения: ГОСТ 19.701-90. – [Дата введения 1992-01-01]. – М.: «Стандартинформ», 2010. – 24 с. (Государственный стандарт союза ССР).

65. Guidance for the Management of Computers and Software in Laboratories with Reference to ISO/IEC 17025/2005: [Электронный ресурс] / Eurolab Technical Report. – No 2/2006. – 31 p. – Режим доступа до інформації: <http://www.eurolab.org/documents/2-2006.pdf>.

66. Кудеяров Ю.А. Структура и особенности Руководства WELMEC 7.2 / Ю.А. Кудеяров, А.Н. Паньков // Измерительная техника. – 2008.– № 5.– С. 69-77.
67. Злыгостева Г.В. Требования к программному обеспечению средств измерений и измерительных систем / Г.В. Злыгостева // Журнал интеллектуальных технологий. – 2009. – № 14. – С. 50-55.
68. Величко О.Н. Основные критерии, этапы и особенности контроля программного обеспечения средств измерений / О.Н. Величко // Измерительная техника. – 2009. – №6. – С. 11-15.
69. Злыгостева Г.В. Обобщенная модель процедуры испытаний измерительного программного обеспечения / Г.В.Злыгостева, С.В. Муравьев // Известия Томского политехнического университета. – 2011. – Т. 318. – №4. – С. 62-67.
70. Васильев Д.Р. Проверка программного обеспечения средств измерений при испытаниях в целях утверждения типа (в порядке дискуссии) / Д.Р. Васильев // Законодательная и прикладная метрология. – 2011. – № 2. – С. 24-36.
71. Порядок проведения испытаний стандартных образцов или средств измерений в целях утверждения типа: ПР 50.2.104-09. – М.: «Стандартинформ», 2010. – 12 с.
72. Левин С.Ф. Статистические методы и метрологическая аттестация программного обеспечения измерительных систем / С.Ф. Левин // Измерительная техника. – 2008. – № 11. – С. 14-18.
73. Олесків О. М. Процедури та методи метрологічної перевірки програмного забезпечення засобів вимірювання / О. М. Олесків, І. П. Микитин // III науково-технічна конференція «Обчислювальні методи і системи перетворення інформації»: Тези доповідей. – Львів, 25-26 вересня 2014 р. – С. 149-152.

74. Jan Jacobson. Methods for Validation and Testing of Software [Електронний ресурс] / Jan Jacobson, Bengt Johansson (Ed.) // MID Software Work Package 2. – 2004. – 35 p. – Режим доступу до інформації: http://www.welmecwg7.ptb.de/Guides/MID-SW-Report-Validation_Testing.pdf.
75. Software Requirements and Validation Guide [Електронний ресурс]: MID Software Guide. – 29 October 2004. – 113 p. – Режим доступу до інформації: http://www.lne.fr/fr/metrologie/metrologie_legale/documents/MIDSW_1.00.pdf.
76. Білас О. Якість програмного забезпечення та тестування: навч. посіб. / О. Білас. – Львів: вид-во НУ «Львівська політехніка», 2011. – 216 с.
77. Канер С. Тестирование программного обеспечения / Канер С., Фолк Дж., Нгуен Енг. – К: Диасофт, 2000. – 544 с.
78. Макгрегор Дж. Тестирование объектно-ориентированного программного обеспечения / Макгрегор Дж, Сайкс Д. – К: Диасофт, 2002. – 432 с.
79. Основы инженерии качества программных систем / [Андон Ф. И., Коваль Г. И., Коротун Т. М. и др.] – Киев: Академперіодика, 2007. – 680 с.
80. Лаврищева Е. М. Методы и средства инженерии программного обеспечения / Е. М. Лаврищева, В. А. Петрухин. – М.: МФТИ, – 2007. – 415 с.
81. Гленфорд Майерс. Искусство тестирования программ, 3-е издание / Гл. Майерс, Т. Баджетт, К. Сандлер. – М.: «Диалектика», 2012. – 272 с.
82. Соммервилл И. Инженерия программного обеспечения [Електронний ресурс] / И. Соммервилл. – М.: Издательский дом "Вильямс", 2002. - 624 с. – Режим доступу до інформації: www.tajfan.com/wp-content/uploads/2015/03/Инженерия-программного-обеспечения-Иан-Сом.
83. Кормен Т. Алгоритмы. Построение и анализ [Електронний ресурс] / Т.Кормен, Ч.Лейзерсон, Р.Ривест. – Москва: ИД «Вильямс», 2011. – 1296 с. – Режим доступу до інформації: http://www.r-5.org/files/books/computers/algo-list/common/Cormen_Leiserson_Rivest_Stein-Introduction_to_Algorithms-RU.pdf.

84. Smart sensor systems [Електронний ресурс]. – Режим доступу до інформації:
http://samples.sainsburysebooks.co.uk/9780470866924_sample_383509.pdf.
85. Ajay Mahajan. Physical and Virtual Intelligent Sensors for Integrated Health Management Systems / Ajay Mahajan, Christopher Oesch, Haricharan Padmanaban, Lucas Utterback, Sanjeevi Chitikeshi, Fernando Figueroa // International journal on smart sensing and intelligent systems. – 2012. – Vol. 5. – №. 3. – P. 559-575.
86. Олеськів О. М. Проблематика та перспективи метрологічної перевірки кібер-фізичних систем / О. М. Олеськів, І. П. Микитин // II Міжнародна науково-практична конференція «Управління якістю в освіті та промисловості: досвід, проблеми та перспективи»: Тези доповідей. – Львів, 28-30 травня 2015 р. – С. 211.
87. Олеськів О. М. Структури кібер-фізичних систем та алгоритми метрологічної перевірки їх програмного забезпечення / І.П. Микитин, О. М. Олеськів //Всеукраїнська науково-технічна конференція молодих вчених у царині метрології «Technical Using of Measurement – 2016»: – Тези доповідей. – Славське, 1 - 5 лютого 2016 р.– С. 56-59.
88. Кочан Р. В. Прецизійні аналого-цифрові перетворювачі з бездемонтажною метрологічною самоперевіркою: монографія / Р. В. Кочан. – Л. : Вид-во Львів. політехніки, 2012. – 251 с.
89. Интегральные микросхемы: Микросхемы для аналого-цифрового преобразования и средств мультимедиа. Выпуск 1. – М: Додека, 1996 – 384 с.
90. Paul R. Gray. Analysis and design of analog integrated circuits [Fifth Edition] / Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, Robert G. Meyer. – Wiley, 2012 – 881 p.
91. Лінійна алгебра та аналітична геометрія: навчальний посібник / [В. В. Булдігін, І. В. Алексєєва, В. О. Гайдей та ін.]. – Київ: ТВіМС, 2011 – 224 с.

92. Холингворт Джаррод. Borland C++ Builder 6. Руководство разработчика / Джаррод Холингворт, Боб Сворт, Марк Кэшмен, Поль Густавсон. – М.: «Вильямс», 2004. – 976 с.
93. Гаврилів О. С. До методу розділення змінних Фур'є : Навчально-методичний посібник. / О. С. Гаврилів. – Львів. : Вид. Тараса Сороки, 2009. – 32 с.
94. Ричард Лайонс. Цифровая обработка сигналов: Второе издание / Ричард Лайонс. – Москва : Бином-Пресс, 2006. – 656 с.
95. Сергиенко А. Б. Цифровая обработка сигналов: Второе издание. / А. Б. Сергиенко. – СПб : Питер, 2006. – 751 с.
96. Комп'ютерне моделювання систем та процесів. Методи обчислень Частина 1: навчальний посібник / [Р. Н. Кветний, І. В. Богач, О. Р. Бойко та ін.]. – Вінниця: ВНТУ, 2012. – 193 с.
97. Gonzales R. C. Digital Image Processing Using MATLAB / R. C. Gonzales , R. E. Woods, S. Eddins. – Prentice Hall, Upper Saddle River, NJ, 2004 – 492 p.
98. Амосов А. А. Вычислительные методы для инженеров: учеб. пособ. / Амосов А. А. , Дубинский Ю. А., Копченова Н. В. – М. : Высшая школа, 1994. – 554 с.
99. Глинченко А. С. Цифровая обработка сигналов / Глинченко А. С. – Красноярск : Изд-во КГТУ, 2001. – 199с.
100. Чабан В. Чисельні методи / В. Чабан. – Львів : Вид. Нац. ун-ту "Львівська політехніка", 2001. – 186 с.
101. Типи даних [Електронний ресурс] / Режим доступу до інформації: <http://cpp.dp.ua/lectures/dan-ta-operacyi/119-tipi-danij.html>.
102. Типи даних С [Електронний ресурс] / Режим доступу до інформації: <http://cppstudio.com/uk/post/271/>.
103. MATLAB - Data Types [Електронний ресурс] / Режим доступу до інформації: https://www.tutorialspoint.com/matlab/matlab_data_types.htm.

104. Дорожовець М. М. Опрацювання результатів вимірювань: навч. посіб. / М. М. Дорожовець. – Л. : Вид-во Львів. політехніки, 2007. – 621 с.
105. Метрологія. Застосування «Руководства по выражению неопределенности измерений» (РМГ 43:2001, IDT) : ДСТУ-Н РМГ 43:2006. – [Чинний від 2007-07-01]. – К.: Держспоживстандарт України, 2003. – 28 с. – (Національний стандарт України).
106. Чалый В. П. Неопределенность и погрешность, их сходство, различие и употребление в разных метрологических процедурах / В. П. Чалый // Системи обробки інформації. – 2006. – Вип. 7. – С. 82-85.
107. Dorozhovets M. Wybrane problemy praktycznej oceny błędów oraz niepewności wyników pomiaru / M. Dorozhovets // Zeszyty Naukowe Politechniki Rzeszowskiej. Elektrotechnika. – 2006. – № 29. – P. 9-44.
108. Метрологічне забезпечення вимірювань і контролю: навчальний посібник / [Є. Т. Володарський, Кухарчук В. В., Поджаренко В. О. та ін.]. – Вінниця: ВДТУ, 2001. – 219 с.
109. Бичківський Р.В. Основи метрологічного забезпечення / Р.В. Бичківський, В.І. Зорій, П.Г. Столярчук. – Львів, ДУ «Львівська політехніка», 1999. – 180 с.
110. Основи метрології та вимірювальної техніки: у 2 т. / [М. Дрожовець, В. Мотало, Б. Стадник та ін.]. – Львів: Вид-во Національного університету «Львівська політехніка», 2004. – Т.1: Основи метрології. – 2004. – 529 с.

Додаток А

Програмна реалізація генератора зразкових цифрових послідовностей

```

// Projekt1.cpp
//-----
#include <vcl.h>
#pragma hdrstop
//-----

USEFORM("Unit1.cpp", Form1);
USEFORM("Unit2.cpp", Form2);
USEFORM("Unit3.cpp", Form3);
USEFORM("Unit4.cpp", Form4);
//-----

WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->CreateForm(__classid(TForm2), &Form2);
        Application->CreateForm(__classid(TForm3), &Form3);
        Application->CreateForm(__classid(TForm4), &Form4);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)

```

```
        {
            try
            {
                throw Exception("");
            }
            catch (Exception &exception)
            {
                Application->ShowException(&exception);
            }
        }
    }
    return 0;
}
// Unit1.cpp
//-----
#include <iostream>
#include <sstream>
#include <math.h>
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
```

```

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Memo1->Clear();
    randomize();    // Для генерування випадкових чисел
    EditPovidomlennja->Text = "";
    PomylkaVvedennjaDanych = false;
    rozrad_AZP = VvedennjaDanyx(Form2->Edit_rozrad_AZP);
    if(rozrad_AZP == -1)PomylkaVvedennjaDanych = true;
    Memo1->Lines->Add("rozrad_AZP = " + FloatToStr(rozrad_AZP));
    f_Dyskret = VvedennjaDanyx(Form2->Edit_f_Dyskret);
    if(f_Dyskret == -1)PomylkaVvedennjaDanych = true;
    Memo1->Lines->Add("f_Dyskret = " + FloatToStr(f_Dyskret));
    T_Peretv = VvedennjaDanyx(Form2->Edit_T_Peretv);
    if(T_Peretv == -1)PomylkaVvedennjaDanych = true;
    Memo1->Lines->Add("T_Peretv = " + FloatToStr(T_Peretv));
    Uvx_n = VvedennjaDanyx(Form2->Edit_Uvx_n);
    if(Uvx_n == -1)PomylkaVvedennjaDanych = true;
    Memo1->Lines->Add("Uvx_n = " + FloatToStr(Uvx_n));
    Uvx_real = VvedennjaDanyx(Form2->Edit_Uvx_real);
    if(Uvx_real == -1)PomylkaVvedennjaDanych = true;
    Memo1->Lines->Add("Uvx_real = " + FloatToStr(Uvx_real));
    Phase = VvedennjaDanyx(Form3->Edit_Phase);
    if(Phase == -1)PomylkaVvedennjaDanych = true;
}

```

```

Memo1->Lines->Add("Phase = " + FloatToStr(Phase));
Fs = VvedennjaDanyx(Form3->Edit_Signal_frequency);
if(Fs == -1)PomylkaVvedennjaDanych = true;
Memo1->Lines->Add("Fs = " + FloatToStr(Fs));
Ktr = VvedennjaDanyx(Form3->Edit_Ktr);
if(Ktr == -1)PomylkaVvedennjaDanych = true;
Memo1->Lines->Add("Ktr = " + FloatToStr(Ktr));
Klsb_add = VvedennjaDanyx(Form2->Edit_k_LSB_add);
if(Klsb_add == -1)PomylkaVvedennjaDanych = true;
Klsb_mult = VvedennjaDanyx(Form2->Edit_k_LSB_mult);
if(Klsb_mult == -1)PomylkaVvedennjaDanych = true;
Klsb_nl = VvedennjaDanyx(Form2->Edit_k_LSB_nl);
if(Klsb_nl == -1)PomylkaVvedennjaDanych = true;
Knoise = VvedennjaDanyx(Form2->Edit_Knoise);
if(Knoise == -1)PomylkaVvedennjaDanych = true;

k_Uvx = Uvx_real/Uvx_n; // для обмеження сигналу > 0 за Uvx_real
if (k_Uvx > 1)
    {
    Form4->Visible = true;
    Form1->Visible = false;
    }

h_Dyskret = 1/f_Dyskret; // = 1/(f_Dyskret*T_Peretv);
Memo1->Lines->Add("h_Dyskret = " + FloatToStr(h_Dyskret));
n_kvant = pow(2,rozrad_AZP);
Memo1->Lines->Add("n_kvant = " + FloatToStr(n_kvant));
n = f_Dyskret*T_Peretv; // кількість даних
A = (n_kvant/2-1)*k_Uvx; // амплітуда
n_save = 5;
N = n_save*n;

```

```

Data_Save = new __int32 [N];
X = new double [n];
Y = new double [n];
Data = new double [n];    // Сигнал + похибки
Data_Err = new double [n]; // Похибки
iFileLength = n*n_save*sizeof(Data_Save[0]);    // для запису в файл
Memo1->Lines->Add("iFileLength = " + FloatToStr(iFileLength));    //

```

для запису в файл

```

// Формування трикутника/пилки
n4 = f_Dyskret/Fs;
Memo1->Lines->Add("n4="+FloatToStr(n4));
n1 = n4/4;
Memo1->Lines->Add("n1="+FloatToStr(n1));
n2 = n1*2;
Memo1->Lines->Add("n2="+FloatToStr(n2));
n3 = n1*3;
Memo1->Lines->Add("n3="+FloatToStr(n3));
k = 0;
Ymax = n_kvant/2-1;    // для обмеження сигналу > 0 за Uvx_real
Ymin = -(n_kvant/2-1);    // для обмеження сигналу < 0 за Uvx_real
Xgr = new double [n+2];
Ygr = new double [n+2];

Y_kvant = new __int32 [n];    // Квантований сигнал
// Похибки
Yadd = new double [n];
Ymult = new double [n];
Ynelin = new double [n];
Ynoise = new double [n];

```

```

Series1->Clear();
Series2->Clear();
Series4->Clear();
p = 0;
for (l=0; l<n_save; l++)
{
    RandAdd = (rand()/16339.-1); // Для генерування випадкових значень
    Memo1->Lines->Add("RA="+FloatToStr(RandAdd));
    RandMult = (rand()/16339.-1); // похибки в діапазоні від -1 до 1 OMP
    RandNelin = (rand()/16339.-1);
    for (i=0; i<n; i++)
    {
        Data[i] = 0;      // Сигнал + похибки
        Data_Err[i] = 0; // Похибки
    }
    if(Form3->RadioButtonSinus->Checked == true)
    {
        for (i=0; i<n; i++)
        {
            X[i] = h_Dyskret*i;
            Y[i] = A*sin(2.*M_PI*Fs*X[i]+Phase);
        }
    }
}
//////////////////////////////////*****//////////////////////////////////
if(Form3->RadioButtonMeander->Checked == true)
{
    for (i=0; i<n; i++)
    {
        X[i] = h_Dyskret*i;

```

```

Y[i] = A*sin(2.*M_PI*Fs*X[i]+Phase);
if(Y[i] < A & Y[i]>0) Y[i] = A; // формування меандру
if(Y[i] > -A & Y[i]<=0) Y[i] = -A; // формування меандру
}
}
////////////////////////////////////
if(Form3->RadioButtonTrapeze->Checked == true)
{
for (i=0; i<n; i++)
{
X[i] = h_Dyskret*i;
Y[i] =
A*sin(2.*M_PI*Fs*X[i]+Phase);//n_kvant/2*k_Uvx*sin(2.*M_PI*Fs*X[i]+Phase
);

Ymax = A; // формування трапеції
Ymin = -A; // формування трапеції
if (Y[i] > Ymax/Ktr) Y[i] = Ymax/Ktr;
else if (Y[i] < Ymin/Ktr) Y[i] = Ymin/Ktr;
else Y[i] = Y[i];
Y[i] = Y[i]*Ktr; // формування трапеції
}
}
////////////////////////////////////
if(Form3->RadioButtonTrykutnyk->Checked == true)
{
for (m = 0, s=0; m < n/n1+2; m++,s++)
{
Xgr[m] = n1*m*h_Dyskret;
if(s>3) s=0;

```



```

    if (s==0) Ygr[m]= 0;
    else if (s==1) Ygr[m]= A;
    else if (s==2) Ygr[m]= 0;
    else if (s==3) Ygr[m]= -A;
    }
for(i=0, j=0; i<n; i++,j++)
    {
    if (j == n1)
        {j = 0;
        k++;
        }
    if (j == 0)
        {
        x1 = Xgr[k];
        y1 = Ygr[k];
        x2 = Xgr[k+1];
        y2 = Ygr[k+1];
        }
    X[i] = h_Dyskret*i;
    Y[i] = (X[i]-x1)*(y2-y1)/(x2-x1)+y1;
    }
}

//////////////////////////////////*****//////////////////////////////////

if(Form3->RadioButtonPylka->Checked == true)
    {
    for (m = 0, s=0; m < n/n4+2; m++,s++)
        {
        Xgr[m] = n4*m*h_Dyskret;
        if(s>1) s=0;

```

```

if (s==0) Ygr[m]= 0;
else Ygr[m]= A;
}
for(i=0, j=0; i<n; i++,j++)
{
    if (j == n4)
        {j = 0;
        k++;
        }
    if (j == 0)
        {
        x1 = Xgr[k];
        x2 = Xgr[k+1];
        }
    y1 = 0;
    y2 = A;
    X[i] = h_Dyskret*i;
    Y[i] = (X[i]-x1)*(y2-y1)/(x2-x1)+y1;

}
}
////////// ГЕНЕРУВАННЯ ПОХИБОК //////////
for (i=0; i<n; i++)
{
    Ynoise[i] = (rand()/32768.-0.5)*2*Knoise;
    Yadd[i] = Klsb_add*RandAdd;
    Ymult[i]   =   Klsb_mult*Y[i]/(n_kvant-2)*2*RandMult;   //
+(n_kvant/2-1)

```

```

Ynelin[i] = Klsb_n1*(-1*((Y[i])*(Y[i]))/((n_kvant/2-
1)*(n_kvant/2-1))+1)*RandNelin;

```

```

if (Form2->CheckBox_Signal->Checked == true)
    { Data[i] = Data[i] + Y[i];
    }
if (Form2->CheckBox_Noise->Checked == true)
    { Data[i] = Data[i] + Ynoise[i];
    }
if (Form2->CheckBox_AddError->Checked == true)
    { Data[i] = Data[i] + Yadd[i];
      Data_Err[i] = Data_Err[i] + Yadd[i];
    }
if (Form2->CheckBox_MultError->Checked == true)
    { Data[i] = Data[i] + Ymult[i];
      Data_Err[i] = Data_Err[i] + Ymult[i];
    }
if (Form2->CheckBox_NotLinError->Checked == true)
    { Data[i] = Data[i] + Ynelin[i];
      Data_Err[i] = Data_Err[i] + Ynelin[i];
    }
if (Form4->RadioButtonObrizaty->Checked == true)
    {
        if (Data[i] > Ymax)          Data[i] = Ymax; // обмеження
сигналу > 0 за Uvx_real
        else if (Data[i] <= Ymin)    Data[i] = Ymin; // обмеження
сигналу < 0 за Uvx_real
        else                          Data[i] = Data[i];
    }

```

```

if (Form4->RadioButtonMasshtabuvaty->Checked == true)
{
    if (k_Uvx>1)          Data[i] = Data[i]/k_Uvx; // обмеження
сигналу > 0 за Uvx_real
    if (k_Uvx<=1)        Data[i] = Data[i];
}
if (Form1->CheckBox_Graf->Checked == true)
{
    Series1->AddXY(X[i],Data[i],"",clRed);
    Series4->AddXY(Y[i],Data_Err[i],"",clRed);
}
}

```

////////////////////// К В А Н Т У В А Н Н Я ////////////////////////

```

if (Form2->CheckBox_Open_Quantisierung->Checked == true)
{
    if(Form2->RadioButton_DigitOben->Checked == true)
    {
        for (i=0; i<n; i++)
        {
            Y_kvant[i] = floor(Data[i]+0.9999);
            if (Form1->CheckBox_Graf->Checked == true)
            {
                Series2->AddXY(X[i],Y_kvant[i],"",clGreen);
            }
        }
    }
}

```

```

if(Form2->RadioButton_DigitUnten->Checked == true)
{

```

```

for (i=0; i<n; i++)
{
    Y_kvant[i] = floor(Data[i]);
    if (Form1->CheckBox_Graf->Checked == true)
    {
        Series2->AddXY(X[i],Y_kvant[i],"",clGreen);
    }
}
}
if(Form2->RadioButton_DigitNaehere->Checked == true)
{
    for (i=0; i<n; i++)
    {
        Y_kvant[i] = floor(Data[i]+0.5);
        if (Form1->CheckBox_Graf->Checked == true)
        {
            Series2->AddXY(X[i],Y_kvant[i],"",clGreen);
        }
    }
}
}
}

```

delete[] Y_kvant;

delete[] X;

delete[] Y;

delete[] Data;

delete[] Data_Err;

delete[] Xgr;

```

delete[] Ygr;
delete[] Yadd;
delete[] Ymult;
delete[] Ynelin;
delete[] Ynoise;
Memo1->Lines->Add("");
Memo1->Lines->Add("End");
Memo1->Lines->Add("");
}
//-----

```

```

float __fastcall TForm1::VvedennjaDanyx (TEdit *TE)
{
    //Захист від поганого введення числа типу float
    KomuZnajdeno = 0;
    TE->Color = clWindow;
    StrichkaFloat = TE->Text;
    if(StrichkaFloat == "")
    {
        TE->Color = clRed;
        EditPovidomlennja->Text = "Перевірте правильність введення
даних";
        EditPovidomlennja->Font->Color = clRed;
        return -1;
    }
    for (int i = 0; i < StrichkaFloat.Length(); i++)
    {
        if(StrichkaFloat.c_str()[i] == ',')KomuZnajdeno++;
        if(StrichkaFloat.c_str()[i] == '.')

```

```

        {
            StrichkaFloat.c_str()[i] = ',';
            KomuZnajdeno++;
        }
        if(((StrichkaFloat.c_str()[i] < '0')||(StrichkaFloat.c_str()[i] >
'9'))&&(StrichkaFloat.c_str()[i] != '-')&&
            (StrichkaFloat.c_str()[i] != '.')&&(StrichkaFloat.c_str()[i] !=
';')||(KomuZnajdeno > 1))
        {
            //ShowMessage("Перевірте правильність введення даних");
            TE->Color = clRed;
            EditPovidomlennja->Text = "Перевірте правильність введення
даних";

            EditPovidomlennja->Font->Color = clRed;
            return -1;
        }
    }
    TE->Text = StrichkaFloat;
    return TE->Text.ToDouble();
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Form2->Visible = true;
    Form1->Visible = false;
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Form3->Visible = true;
    Form1->Visible = false;
}
//-----
void __fastcall TForm1::Button5Click(TObject *Sender)
{
    if (SaveDialog1->Execute())
    {
        iFileHandle = FileCreate(SaveDialog1->FileName);
        FileWrite(iFileHandle, Data_Save, iFileLength);
        FileClose(iFileHandle);
    }
    if (Form1->CheckBox_Data->Checked == true)
    {
        for (i=0; i<f_Dyskret*T_Peretv; i++)

```

```

        {
            Memo1->Lines->Add(" Y_kvant = " + IntToHex( Y_kvant[i], 8));
        }
    }
}
//-----
void __fastcall TForm1::Button6Click(TObject *Sender)
{
if (OpenDialog1->Execute())
{
    try
    {
        iFileHandle = FileOpen(OpenDialog1->FileName, fmOpenRead);
        iFileLength = FileSeek(iFileHandle,0,2);
        FileSeek(iFileHandle,0,0);
        iBytesRead = FileRead(iFileHandle, Data_Save, iFileLength);
        FileClose(iFileHandle);
        Memo1->Lines->Add("iFileLength = " + IntToStr(iFileLength));
    }
    catch(...)
    {
        Application->MessageBox("Can't perform one of the following file
operations: Open, Seek, Read, Close.", "File Error", IDOK);
    }
}
}
//-----

```


Додаток Б

Акти впровадженнь результатів дисертаційної роботи

ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної роботи
Національного університету

«Львівська політехніка»

к.т.н., доц. Давидчук О.Р.

“ 12 ” 2016 р.



АКТ

про впровадження результатів дисертаційної роботи в навчальний процес

*Олеськів Ольги Михайлівни***«Метрологічна перевірка програмного забезпечення засобів вимірювання»**

представленої на здобуття наукового ступеня кандидата технічних наук за спеціальністю

05.01.02 – *стандартизація, сертифікація та метрологічне забезпечення*

Комісія НУ «Львівська політехніка» у складі:

Голова комісії – голова науково-методичної ради інституту комп'ютерних технологій та метрології, д.т.н., проф. Байцар Р.І.

Члени комісії: професор кафедри "Інформаційно-вимірювальні технології", д.т.н., проф. Скоропад П.І., професор кафедри "Інформаційно-вимірювальні технології", д.т.н., проф. Яцишин С.П. і професор кафедри "Інформаційно-вимірювальні технології", д.т.н., проф. Дорожовець М.М.

даним актом підтверджує, що проведені дисертанткою наукові дослідження виконувалися нею на кафедрі інформаційно-вимірювальних технологій Національного університету «Львівська політехніка» у співпраці з кафедрою вимірювання процесів і сенсорної техніки Технічного університету міста Ільменау (Німеччина). Основні положення та результати дисертаційної роботи впроваджені у навчальний процес кафедри «Інформаційно-вимірювальні технології» Національного університету «Львівська політехніка» при вивченні дисципліни «Тестування інформаційно-вимірювальних систем» для підготовки фахівців за спеціалізацією 152.01 «Метрологія та вимірювальна техніка», а також при підготовці бакалаврських та магістерських кваліфікаційних робіт.

Голова комісії,
голова науково-методичної ради ІКТА
д.т.н., проф.

Байцар Р.І.

Члени комісії:
проф. каф. ІВТ, д.т.н. проф.

Скоропад П.І.

проф. каф. ІВТ, д.т.н. проф.

Яцишин С.П.

проф. каф. ІВТ, д.т.н. проф.

Дорожовець М.М.



ЗАТВЕРДЖУЮ

Проректор з наукової роботи
 Національного університету
 Львівська політехніка»

проф. Чухрай Н. І.

29" 12. 2016 р.

АКТ

про використання результатів дисертаційної роботи

Олеськів Ольги Михайлівни

«Метрологічна перевірка програмного забезпечення засобів вимірювання»

представленої на здобуття наукового ступеня кандидата технічних наук за спеціальністю

05.01.02 – *стандартизація, сертифікація та метрологічне забезпечення*

Комісія у складі: голови – начальника науково-дослідної частини, к.т.н., доцента Жук Л.В. та членів: завідувача кафедри інформаційно-вимірювальних технологій, д.т.н., професора Стадника Б.І., завідувача відділу науково-організаційного супроводу наукових досліджень, к.т.н. Лазько Г.В. і заступника начальника планово-фінансового відділу Чулой Т.М., цим актом підтверджують, що результати дисертації Олеськів Ольги Михайлівни використовувалися при виконанні науково-дослідної роботи «Вимірювання температури мікро- та наноструктурованих об'єктів методом комбінаційного розсіювання світла» (номер державної реєстрації 0115U000431, наказ Міністерства освіти і науки України від 31.10.2014 р. № 1243).

Зокрема, Олеськів Ольгою Михайлівною розроблено програмно-математичні моделі похибок елементів оптичної схеми та моделі спектру комбінаційного розсіювання світла, проведено дослідження похибки вимірювання температури за зсувом еквівалентної частоти спектру комбінаційного розсіювання світла.

Голова комісії,
 начальник науково-дослідної
 частини, к.т.н., доц.

Жук Л.В.

Члени комісії:
 зав. каф. інформаційно-вимірювальних
 технологій, д.т.н. проф.

Стадник Б.І.

зав відділу науково-організаційного
 супроводу наукових досліджень, к.т.н.

Лазько Г.В.

заст. нач. планово-фінансового відділу

Чулой Т.М.