

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО
ГОСПОДАРСТВА імені О. М. БЕКЕТОВА
Навчально-науковий Інститут енергетичної, інформаційної та
транспортної інфраструктури
Кафедра автоматизації та комп'ютерно-інтегрованих технологій

РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

на тему: Розробка інтелектуальної системи голосового керування
комп'ютером із застосуванням алгоритмів розпізнавання мовлення

Виконав: здобувач вищої освіти
3 курсу, групи Сінж-2023-1-У
напряму підготовки (спеціальності)
174 «Автоматизація, комп'ютерно-
інтегровані технології та робототехніка»
Марков В. В.

(прізвище та ініціали)

Керівник Білецький І.В., проф. каф. АКІТ
(прізвище та ініціали, наук. ступ., вч. звання)

Рецензент Ківіренко О.Б., начальник
виробництва ТОВ «Альфа-Композіт»
(прізвище та ініціали, наук. ступ., вч. звання)

Харків – 2026

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО
ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА**

**Навчально-науковий Інститут енергетичної, інформаційної та
транспортної інфраструктури**

Кафедра автоматизації та комп'ютерно-інтегрованих технологій
Освітньо-кваліфікаційний рівень – бакалавр
Галузь знань 17 «Електроніка, автоматизація та електронні комунікації»
Спеціальність 174 «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри АКІТ



БАРАНОВ О.О.

« 19 » червня 2026 року

З А В Д А Н Н Я

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Марков Владислав Валерійович

1. Тема роботи Розробка інтелектуальної системи голосового керування
комп'ютером із застосуванням алгоритмів розпізнавання мовлення.

Затверджена наказом університету від « 22 » травня 2026 року № 440-03.

Керівник роботи Білецький Ігор Васильович, доктор економічних наук,
професор, професор кафедри АКІТ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання роботи здобувачем вищої освіти « 15 » червня 2026 р.

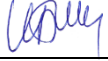







3. Вихідні дані до роботи Система голосового керування комп'ютером із
застосуванням алгоритмів розпізнавання мовлення.

4. Зміст розрахунково пояснювальної записки (перелік питань, які потрібно
розробити): Вступ. Теоретичні основи та аналіз систем голосового керування.
проектування інтелектуальної системи голосового керування. Проектування
інтелектуальної системи голосового керування. Програмна реалізація системи
голосового керування комп'ютером. Охорона праці, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових
креслень)

Презентація.

6. Консультанти розділів проєкту (роботи)

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Аналіз проблеми	Білецький І.В.	11.05.2026 	21.05.2026 
Основна частина	Білецький І.В.	22.05.2026 	31.05.2026 
Спеціальний розділ	Білецький І.В.	01.06.2026 	13.06.2026 
Охорона праці	Малишева В.В.	07.06.2026 	13.06.2026 

7. Дата видачі завдання «_11_» травня 2026 р.





Керівник  Білецький І.В.

(підпис)

Завдання прийняв до виконання  Марков В. В.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання розділів	Примітка
1	Розробка 1го розділу бакалаврської роботи	11.05.2026 - 21.05.2026	
2	Розробка 2го розділу бакалаврської роботи	22.05.2026 - 31.05.2026	
3	Розробка 3го розділу бакалаврської роботи	01.06.2026 - 13.06.2026	
4	Розробка розділу з охорони праці	07.06.2026 - 13.06.2026	
5	Рецензування бакалаврської роботи	15.06.2026	Ківіренко О.Б
6	Захист на ДЕК	25.06.2026	

Здобувача вищої освіти  Марков В. В.

(підпис)

Керівник  Білецький І.В.

(підпис)

РЕФЕРАТ

Розробка інтелектуальної системи голосового керування комп'ютером із застосуванням алгоритмів розпізнавання мовлення – Марков Владислав Валерійович, дипломна робота бакалавра, Харків, Харківський національний університет міського господарства імені О.М. Бекетова, кількість сторінок 98, кількість таблиць 37, кількість рисунків 19, кількість джерел літератури 26.

У сучасних умовах розвитку інформаційних технологій зростає потреба у зручних та ефективних способах взаємодії користувача з комп'ютером. Традиційні засоби керування, такі як клавіатура та миша, не завжди забезпечують достатній рівень швидкості та зручності роботи. Одним із перспективних напрямів є використання технологій розпізнавання мовлення, які дозволяють здійснювати керування комп'ютером за допомогою голосових команд. Розробка інтелектуальної системи голосового управління комп'ютером на основі алгоритмів розпізнавання мовлення сприяє підвищенню ефективності роботи користувача, розширює можливості автоматизації та покращує доступність комп'ютерних систем, що зумовлює актуальність дослідження та практичну значущість даної роботи.

Метою роботи є розробка та дослідження інтелектуальної системи голосового управління комп'ютером на основі алгоритмів розпізнавання мовлення та словника команд, яка забезпечує ефективне виконання дій користувачем і підвищує зручність взаємодії з комп'ютерною системою.

Об'єктом дослідження є процеси взаємодії користувача з комп'ютером через голосові команди та системи розпізнавання мовлення.

Предметом дослідження є методи та алгоритми обробки голосових команд і їх інтеграції у систему інтелектуального управління комп'ютером.

Для досягнення поставленої мети в роботі необхідно розв'язати такі задачі:

- Провести аналіз існуючих методів голосового управління та алгоритмів розпізнавання мовлення.

- Розробити структуру інтелектуальної системи голосового управління комп'ютером.
- Створити словник команд (глосарій) для системи та алгоритми його інтеграції.
- Реалізувати програмну систему голосового управління комп'ютером із застосуванням алгоритмів розпізнавання мовлення.

У процесі виконання бакалаврської роботи використано такі методи дослідження, як аналіз літературних та наукових джерел з теми голосового управління та розпізнавання мовлення; системний підхід до побудови архітектури програмної системи; методи програмної реалізації алгоритмів розпізнавання мовлення.

КЛЮЧОВІ СЛОВА: інтелектуальна система, алгоритми розпізнавання мовлення, обробка даних, модель.

ABSTRACT

Development of an Intelligent VoiceControl System for a Computer Using Speech Recognition Algorithms – Markov Vladislav Valeriyovych, bachelor's thesis, Kharkiv, Kharkiv National University of Urban Economy named after O.M. Beketov, number of pages 98, number of tables 37, number of figures 19, number of literature sources 26.

In modern conditions of information technology development, there is a growing need for convenient and effective ways for the user to interact with the computer. Traditional control tools, such as the keyboard and mouse, do not always provide a sufficient level of speed and convenience of work. One of the promising areas is the use of speech recognition technologies, which allow controlling the computer using voice commands. The development of an intelligent computer voice control system based on speech recognition algorithms helps to increase the efficiency of the user's work, expands the possibilities of automation and improves the availability of computer systems, which determines the relevance of the research and practical significance of this work.

The purpose of the work is to develop and research an intelligent computer voice control system based on speech recognition algorithms and a command dictionary, which ensures the effective execution of actions by the user and increases the convenience of interaction with the computer system.

The object of research is the processes of user interaction with a computer through voice commands and speech recognition systems.

The subject of the research is methods and algorithms for processing voice commands and their integration into the intelligent computer control system.

To achieve the goal set in the work, it is necessary to solve the following tasks:

- To analyze existing voice control methods and speech recognition algorithms.
- To develop the structure of an intelligent computer voice control system.

- To create a command dictionary (glossary) for the system and its integration algorithms.

- To implement a computer voice control software system using speech recognition algorithms.

In the process of completing the bachelor's thesis, the following research methods were used: analysis of literary and scientific sources on the topic of voice control and speech recognition; a systematic approach to building the architecture of a software system; methods of software implementation and testing of speech recognition algorithms.

KEYWORDS: intelligent system, speech recognition algorithms, data processing, model.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ	12
1.1 Принципи та методи розпізнавання мовлення	12
1.2 Огляд існуючих систем голосового керування комп'ютером	17
1.3 Технології та інструменти для створення систем розпізнавання мовлення	25
Висновок до розділу 1	29
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ....	31
2.1 Визначення функціональних та нефункціональних вимог до системи голосового керування .	31
2.2 Розробка архітектури системи	33
2.3 Алгоритми розпізнавання та обробки голосових команд	37
Висновок до розділу 2.....	44
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ КОМП'ЮТЕРОМ.....	45
3.1 Розробка програмного забезпечення системи голосового керування комп'ютером	45
3.2 Інтеграція системи з операційною системою та прикладними програмами	61
3.3 Сценарії використання та приклади роботи системи	65
Висновок до розділу 3.....	70
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	71
4.1 Організаційно-правові основи забезпечення безпеки праці	71
4.2 Характеристика об'єкта та виявлення потенційних небезпек	72
4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження.....	74
Висновок до розділу 4.....	80
ЗАГАЛЬНІ ВИСНОВКИ	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	84
Додаток А.....	87

ВСТУП

Сучасні комп'ютерні системи все більше інтегруються у повсякденне життя та виробничі процеси, що підвищує потребу у зручних та ефективних методах взаємодії користувача з комп'ютером. Традиційні способи введення інформації (клавіатура, миша, сенсорні екрани) мають обмежену зручність, особливо в умовах швидкого робочого темпу або для людей з обмеженими фізичними можливостями.

Голосове управління комп'ютером на основі алгоритмів розпізнавання мовлення дозволяє підвищити швидкість і комфорт користування системами, забезпечити інноваційний рівень автоматизації та створити підґрунтя для впровадження інтелектуальних асистентів і роботизованих систем. Застосування алгоритмів штучного інтелекту у поєднанні зі словниками команд дозволяє забезпечити точне та ефективне виконання завдань користувачем через голосові команди, що робить тему дослідження актуальною для автоматизації, комп'ютерно-інтегрованих технологій та робототехніки [1–3].

Метою роботи є створення та дослідження інтелектуальної системи керування комп'ютером за допомогою голосових команд, що базується на використанні алгоритмів розпізнавання мовлення та словника команд. Розроблена система повинна забезпечувати точне виконання команд користувача та підвищувати зручність взаємодії людини з комп'ютером.

Об'єктом дослідження є процес взаємодії користувача з комп'ютерною системою за допомогою голосових команд із використанням технологій розпізнавання мовлення.

Предметом дослідження є методи та алгоритми оброблення голосових команд, а також способи їх інтеграції в систему інтелектуального керування комп'ютером.

Для досягнення поставленої мети в роботі необхідно розв'язати такі задачі:

- Провести аналіз сучасних методів голосового керування та алгоритмів розпізнавання мовлення.
- Розробити архітектуру інтелектуальної системи голосового керування комп'ютером.
- Сформувати словник команд (глосарій) для системи та визначити алгоритми його інтеграції.
- Реалізувати програмну систему голосового керування комп'ютером із використанням алгоритмів розпізнавання мовлення.

У процесі виконання бакалаврської роботи використано комплекс методів дослідження, які забезпечують теоретичне обґрунтування та практичну реалізацію системи голосового управління комп'ютером.

Насамперед застосовано аналіз літературних та наукових джерел, присвячених технологіям голосового управління та алгоритмам розпізнавання мовлення. Використання цього методу дозволило дослідити сучасний стан розвитку технологій обробки мовлення, визначити основні підходи до створення систем голосового керування, а також проаналізувати переваги та недоліки існуючих рішень. Проведений аналіз дав можливість обґрунтувати вибір технологій та інструментів, які були використані під час розробки програмної системи.

Для побудови структури програмної системи було використано системний підхід. Застосування цього методу дозволяє розглядати систему голосового управління як сукупність взаємопов'язаних компонентів, кожен із яких виконує певні функції. У межах системного підходу було визначено основні модулі системи, зокрема модуль прийому голосового сигналу, модуль обробки мовлення, модуль розпізнавання команд та модуль виконання дій у комп'ютерній системі. Такий підхід забезпечує структурованість архітектури програмного продукту, спрощує процес розробки та подальшої модифікації системи.

Також у роботі використано методи програмної реалізації алгоритмів розпізнавання мовлення. Їх застосування передбачає створення програмних

модулів, що здійснюють обробку аудіосигналу, перетворення його у текстову форму та подальший аналіз для визначення відповідної команди. Реалізація цих алгоритмів дозволяє автоматизувати процес обробки голосових команд і забезпечує можливість керування комп'ютером без використання традиційних пристроїв введення.

Застосування зазначених методів дослідження забезпечило комплексний підхід до розробки інтелектуальної системи голосового управління комп'ютером, поєднуючи теоретичний аналіз існуючих рішень із практичною реалізацією та перевіркою ефективності запропонованих алгоритмів.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ СИСТЕМ ГОЛОСОВОГО КЕРУВАННЯ

1.1 Принципи та методи розпізнавання мовлення

Розпізнавання мовлення є важливим напрямом розвитку інформаційних технологій, що забезпечує природну взаємодію людини з комп'ютерними системами. Основною метою технологій розпізнавання мовлення є автоматичне перетворення усного мовлення у текст або набір команд, які можуть бути використані для керування програмним забезпеченням чи технічними пристроями. Такі системи широко застосовуються у голосових помічниках, мобільних додатках, системах автоматизації та інтелектуальних інтерфейсах керування.

Процес розпізнавання мовлення складається з кількох взаємопов'язаних етапів: отримання аудіосигналу, його попередньої обробки, виділення ознак та застосування алгоритмів розпізнавання. Кожен із цих етапів відіграє важливу роль у забезпеченні точності та ефективності роботи системи (таблиця 1.1).

Таблиця 1.1 – Основні етапи розпізнавання мовлення

Етап	Опис
Захоплення аудіосигналу	Отримання голосового сигналу за допомогою мікрофона
Оцифрування сигналу	Перетворення аналогового звуку в цифровий формат
Попередня обробка	Фільтрація шумів, нормалізація гучності, поділ сигналу на фрагменти
Виділення ознак	Аналіз спектральних характеристик сигналу
Розпізнавання мовлення	Використання алгоритмів для визначення слів або команд
Інтерпретація	Перетворення розпізнаного тексту у відповідні дії системи

Першим етапом є отримання аудіосигналу за допомогою мікрофона. Оскільки мовлення людини є аналоговим сигналом, воно повинно бути перетворене у цифрову форму за допомогою аналого-цифрового перетворення. У результаті цього процесу звуковий сигнал представляється у вигляді послідовності дискретних значень, які можуть бути оброблені комп'ютером. Якість оцифрування значною мірою впливає на точність подальшого розпізнавання.

Після цього виконується попередня обробка аудіосигналу. На цьому етапі здійснюється фільтрація шумів, нормалізація рівня сигналу, а також поділ сигналу на короткі часові фрагменти, які називаються фреймами. Такий підхід дозволяє аналізувати мовлення як послідовність невеликих сегментів, що спрощує подальшу обробку. Однією з важливих задач попередньої обробки є також виявлення ділянок сигналу, які містять мовлення, та відокремлення їх від пауз або сторонніх шумів.

Наступним етапом є виділення ознак аудіосигналу. Безпосередня обробка «сирого» звукового сигналу є складною через його велику розмірність, тому з нього виділяються найбільш інформативні характеристики.

Таблиця 1.2 – Основні методи виділення ознак мовного сигналу

Метод	Опис	Особливості
MFCC (Mel-Frequency Cepstral Coefficients)	аналіз частотних характеристик сигналу	найпоширеніший метод у системах розпізнавання
LPC (Linear Predictive Coding)	прогнозування параметрів сигналу	використовується у системах стиснення мовлення
Spectrogram analysis	аналіз спектра сигналу	дозволяє візуально досліджувати мовний сигнал

До таких характеристик належать спектральні та енергетичні параметри сигналу. Одним із найбільш поширених методів виділення ознак є використання мел-частотних кепстральних коефіцієнтів (MFCC). Цей метод базується на перетворенні сигналу у частотну область та моделюванні сприйняття звуку людським слухом. Отримані коефіцієнти дозволяють ефективно описати акустичні особливості мовлення та використовуються як вхідні дані для алгоритмів розпізнавання.

Після виділення ознак застосовуються спеціальні алгоритми розпізнавання мовлення, що визначають, яке слово або команда були вимовлені. Існує кілька основних підходів до реалізації цього процесу (таблиця 1.3).

Таблиця 1.3 – Основні алгоритми розпізнавання мовлення

Алгоритм	Принцип роботи	Переваги	Недоліки
Приховані марковські моделі (HMM)	статистичне моделювання послідовностей сигналу	ефективні для обробки часових сигналів	потребують великої кількості даних
Нейронні мережі (DNN)	аналіз складних закономірностей у даних	висока точність	потребують значних обчислювальних ресурсів
Рекурентні нейронні мережі (RNN, LSTM)	врахування послідовності сигналів у часі	добре працюють з мовними даними	складність навчання
Трансформерні моделі	аналіз великих обсягів мовних даних	дуже висока точність	потребують потужного обладнання

Одним із класичних методів є використання прихованих марковських моделей. Цей підхід ґрунтується на статистичному описі мовного сигналу як послідовності станів, що відповідають певним фонемам або елементам мовлення. Приховані марковські моделі дозволяють визначати ймовірність переходів між станами та знаходити найбільш імовірну послідовність слів.

Іншим підходом є використання методів машинного навчання. У таких системах модель навчається на великій кількості прикладів мовлення та поступово формує здатність розпізнавати слова або команди. Особливо ефективними є методи глибокого навчання, які використовують багатошарові нейронні мережі для аналізу складних закономірностей у звукових даних.

Сучасні системи розпізнавання мовлення часто застосовують рекурентні нейронні мережі, які здатні враховувати часову структуру сигналу. Також широко використовуються моделі типу LSTM, що дозволяють ефективно обробляти довгі послідовності даних і зберігати інформацію про попередні елементи сигналу. Останніми роками значного поширення набули трансформерні архітектури, які забезпечують високу точність розпізнавання та активно застосовуються у сучасних системах штучного інтелекту. Важливу роль у таких системах відіграють мовні моделі, які дозволяють враховувати граматичні та статистичні закономірності мови (таблиця 1.4).

Таблиця 1.4 – Функції мовних моделей у системах розпізнавання мовлення

Функція	Опис
Аналіз контексту	визначення найбільш імовірної послідовності слів
Підвищення точності	зменшення кількості помилок під час розпізнавання
Обробка природної мови	адаптація системи до особливостей мови

Мовна модель використовується для визначення найбільш імовірної послідовності слів у певному контексті. Мовні моделі дозволяють підвищити точність розпізнавання, враховуючи граматичні та статистичні закономірності мови.

Часто сучасні системи розпізнавання мовлення об'єднують кілька алгоритмів у єдину модель. Наприклад, нейронні мережі використовуються для аналізу акустичних ознак, а мовні моделі допомагають визначати найбільш імовірну послідовність слів.

На основі технологій розпізнавання мовлення створюються голосові інтерфейси, які забезпечують взаємодію користувача з комп'ютерною системою за допомогою голосових команд. Голосові інтерфейси дозволяють значно спростити процес керування програмами, виконання системних команд та доступ до інформації. Вони особливо корисні у ситуаціях, коли використання традиційних пристроїв введення, таких як клавіатура або миша, є незручним або неможливим.

Сучасні підходи до створення голосових інтерфейсів передбачають використання технологій штучного інтелекту, хмарних сервісів обробки мовлення та інтеграцію з різними програмними платформами. Такі системи здатні не лише розпізнавати окремі слова або команди, а й аналізувати природну мову користувача, що робить взаємодію більш інтуїтивною та ефективною.

Розпізнавання мовлення є складним багаторівневим процесом, який поєднує методи цифрової обробки сигналів, статистичного аналізу та машинного навчання. Використання сучасних алгоритмів та моделей дозволяє створювати ефективні системи голосового керування, що відкриває нові можливості для автоматизації взаємодії людини з комп'ютерними системами.

1.2 Огляд існуючих систем голосового керування комп'ютером

Розвиток технологій розпізнавання мовлення сприяв появі різноманітних систем голосового керування комп'ютером. Такі системи дозволяють користувачам виконувати різні дії за допомогою голосових команд, що значно спрощує взаємодію з комп'ютерними системами та підвищує ефективність роботи. Вони можуть використовуватися для запуску програм, керування файлами, введення тексту, навігації по інтерфейсу операційної системи та виконання інших задач.

Сучасні програмні рішення у сфері голосового керування базуються на алгоритмах обробки мовлення, методах машинного навчання та технологіях штучного інтелекту. Розглянемо найбільш поширені системи голосового керування, їх функціональні можливості, а також основні переваги та недоліки.

Однією з найвідоміших систем є *Windows Speech Recognition*, яка інтегрована в операційну систему Windows. Ця система дозволяє користувачам керувати комп'ютером за допомогою голосу, виконувати базові системні команди, відкривати програми та вводити текст у різні додатки. До основних функцій системи належать розпізнавання голосових команд, диктування тексту, навігація по інтерфейсу операційної системи та керування програмами.

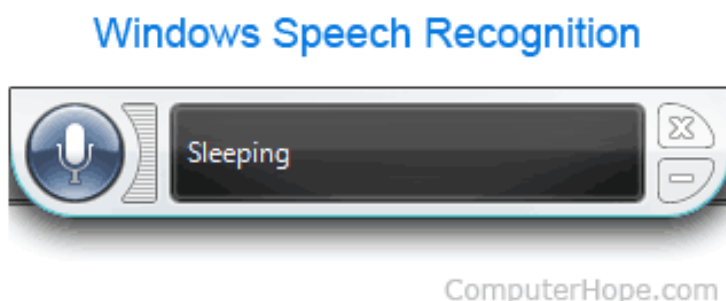


Рисунок 1.1 – Windows Speech Recognition

Перевагою цієї системи є її інтеграція з операційною системою, що забезпечує простоту налаштування та використання. Крім того, вона не

потребує встановлення додаткового програмного забезпечення. Проте її недоліками є обмежена точність розпізнавання в деяких умовах, залежність від якості мікрофона та обмежена підтримка різних мов.

Іншим популярним рішенням є система *Dragon NaturallySpeaking*, яка є однією з найбільш точних програм для розпізнавання мовлення. Вона широко використовується для диктування тексту, автоматичного введення інформації у документи та керування комп'ютером. Система підтримує різні команди керування, дозволяє створювати власні голосові команди та має можливість адаптації до голосу користувача.



Рисунок 1.2 – Dragon NaturallySpeaking

Основною перевагою *Dragon NaturallySpeaking* є висока точність розпізнавання мовлення та широкий набір функціональних можливостей. Система добре підходить для професійного використання, наприклад у сфері медицини, юриспруденції або технічної документації. Водночас її недоліками є висока вартість програмного забезпечення та необхідність попереднього навчання системи для досягнення максимальної точності.

Ритм і динаміка життя людей з кожним днем прогресує. У людей немає часу читати довгі пости, дивитися довгі відеоролики, у них навіть немає часу

писати текст. Тому люди все частіше користуються голосовими помічниками. За прогнозами Gartner найближчим часом 30 % сеансів перегляду вебсторінок будуть відбуватися без використання екранів, а 50 % з усіх запитів будуть голосовими.

Ще одним прикладом сучасних систем голосового керування є голосові помічники, такі як Google Assistant, Apple Siri та Amazon Alexa. Хоча вони переважно використовуються на мобільних пристроях або розумних пристроях, багато їхніх функцій можуть бути інтегровані з комп'ютерними системами. Ці системи використовують хмарні технології та алгоритми штучного інтелекту для розпізнавання мовлення та обробки природної мови.

Siri (Apple) – це голосовий асистент, який став одним із перших популярних прикладів використання технологій розпізнавання мовлення. У 2011 році компанія Apple представила цього помічника разом зі смартфоном iPhone 4s, інтегрувавши його безпосередньо в операційну систему пристрою. На початковому етапі Siri міг взаємодіяти приблизно з дванадцятьма додатками. Користувачі мали можливість за допомогою голосових команд створювати події та нагадування в календарі, отримувати інформацію про погодні умови, користуватися навігацією, а також надсилати електронні листи.

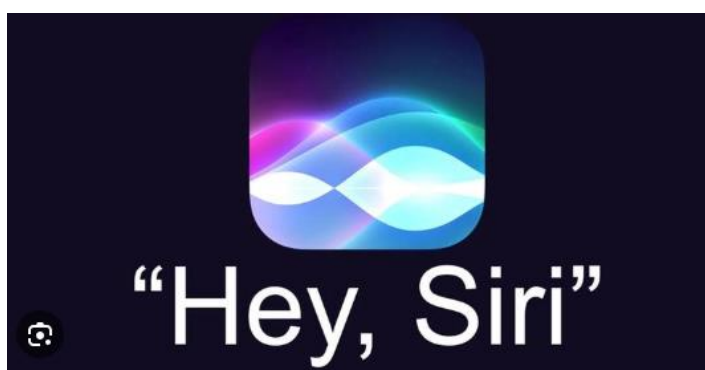


Рисунок 1.3 – Siri (Apple)

З часом функціональні можливості цього голосового помічника значно розширилися. Приблизно через дев'ять років після першого запуску система

стала значно більш інтелектуальною та функціональною. За цей період база знань Siri збільшилася приблизно у двадцять разів. Сучасна версія помічника здатна виконувати значно більше завдань, зокрема надсилати голосові повідомлення, здійснювати переклад текстів різними іноземними мовами, а також виконувати широкий спектр інших дій, пов'язаних із керуванням пристроєм та обробкою інформації.

Google Assistant (Google) з'явився приблизно через рік після появи Siri і став одним із найпоширеніших голосових помічників. Його основна функціональність орієнтована на ефективний пошук інформації в Інтернеті, а також на тісну інтеграцію з хмарними сервісами компанії Google. Система активно взаємодіє з різними сервісами екосистеми Google та забезпечує повну синхронізацію з операційною системою Android, що дозволяє користувачам керувати пристроєм, отримувати інформацію та виконувати різноманітні завдання за допомогою голосових команд.



Рисунок 1.4 – Google Assistant (Google)

Cortana (Microsoft) – це голосовий помічник, розроблений компанією Microsoft. Вперше його було представлено у 2013 році як інтелектуального асистента для операційної системи Windows Phone, а згодом він був інтегрований і в операційні системи для персональних комп'ютерів. Основним призначенням Cortana було спрощення взаємодії користувача з пристроєм за допомогою голосових команд.

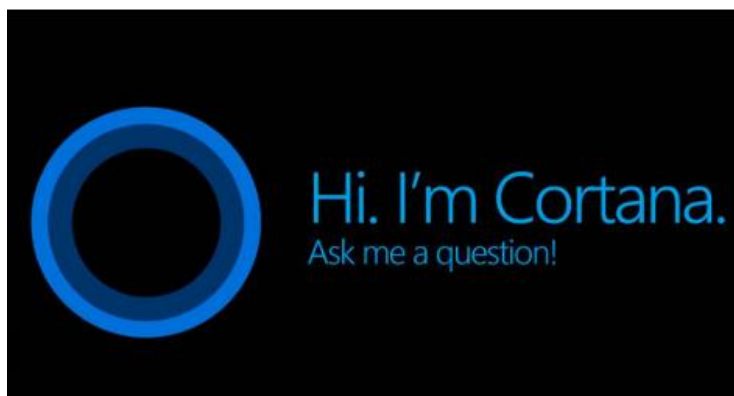


Рисунок 1.5 – Cortana (Microsoft)

За допомогою цього голосового помічника користувачі могли виконувати різні завдання: здійснювати пошук інформації в Інтернеті, керувати програмами, створювати нагадування, планувати події в календарі, а також отримувати відповіді на різні запити. З часом функціональні можливості Cortana були розширені, і голосовий асистент став доступним не лише на комп'ютерах та мобільних пристроях, але й на ігровій консолі Xbox, що дозволило керувати деякими функціями системи за допомогою голосу.

Cortana стала одним із прикладів інтеграції технологій розпізнавання мовлення у програмні продукти компанії Microsoft, спрямованих на підвищення зручності роботи користувачів з цифровими пристроями.

Alexa (Amazon) – це голосовий асистент, створений компанією Amazon, який спочатку був представлений у складі розумних колонок. Основною ідеєю цієї технології було створення зручного інструменту голосової взаємодії між користувачем та цифровими сервісами. Завдяки використанню технологій розпізнавання мовлення *Alexa* здатна виконувати різноманітні команди та відповідати на запити користувачів.

Функціональні можливості цього голосового помічника досить широкі. *Alexa* підтримує голосове спілкування з користувачем, може відтворювати музику, подкасти та аудіокниги, створювати списки справ, встановлювати нагадування та будильники, а також виконувати інші повсякденні завдання. Система може взаємодіяти з різними пристроями «розумного дому», що дозволяє керувати освітленням, побутовою технікою та іншими підключеними пристроями за допомогою голосових команд.

Важливою особливістю Alexa є те, що її розробка була орієнтована не лише на надання інформаційних послуг, а й на спрощення процесу онлайн-покупок. Завдяки інтеграції з сервісами Amazon користувачі можуть швидко знаходити потрібні товари, оформлювати замовлення та керувати покупками за допомогою голосових запитів. Такий підхід значно спрощує процес придбання товарів і робить його більш зручним.



Рисунок 1.6 – Alexa (Amazon)

Згідно з дослідженням компанії YouGov, значну частку ринку голосових помічників займають кілька основних систем. Найбільшу частку – приблизно 36 % – мають голосові асистенти Apple Siri та Google Assistant. Близько 25 % ринку припадає на Amazon Alexa, тоді як Microsoft Cortana займає приблизно 19 %. Така статистика свідчить про значну популярність технологій голосового керування та їх активний розвиток. Очікується, що в майбутньому роль голосових інтерфейсів лише зростатиме, а функціональні можливості подібних систем будуть постійно розширюватися. Голосовий асистент – віртуальний помічник, який працює на основі штучного інтелекту. Він розпізнає мову користувача, може аналізувати його відповіді та виконує команди людини. 91 % користувачів використовують голосових помічників для отримання відповідей на питання, 89,5 % – для прослуховування музики, 85,2 % – для перевірки погоди та 71,4 % – для встановлення таймеру [2].

Голосові помічники Google Assistant, Cortana, Apple Siri та Amazon Alexa є прикладами сучасних інтелектуальних систем, які використовують технології розпізнавання мовлення для забезпечення зручної взаємодії між користувачем і комп'ютерною технікою. Основна ідея їх використання полягає в тому, щоб замінити або доповнити традиційні засоби керування, такі як клавіатура та миша, природним голосовим спілкуванням.

Однією з головних можливостей таких систем є керування комп'ютером за допомогою голосових команд. Користувач може звернутися до голосового помічника та дати команду виконати певну дію, наприклад відкрити програму, знайти інформацію, створити документ або змінити налаштування системи. Голосовий помічник аналізує мовний запит, розпізнає ключові слова та перетворює їх у відповідні системні команди. Наприклад, Google Assistant інтегрується з сервісами Google та операційною системою Android, що дозволяє виконувати різні завдання за допомогою голосу. Користувач може запускати додатки, шукати файли, створювати нотатки або керувати налаштуваннями пристрою. Крім того, асистент може використовуватися для введення тексту в різних додатках через функцію голосового диктування.

Голосовий помічник Cortana, розроблений компанією Microsoft, був інтегрований в операційну систему Windows і дозволяв керувати комп'ютером без використання традиційних пристроїв введення. За допомогою Cortana користувач міг відкривати програми, здійснювати пошук файлів на комп'ютері, створювати нагадування, працювати з календарем та виконувати інші системні дії. Такий підхід значно підвищував зручність використання комп'ютера, особливо під час виконання повсякденних задач.

Apple Siri також надає можливість голосового керування пристроями компанії Apple. Завдяки інтеграції з операційними системами iOS та macOS користувач може відкривати програми, керувати налаштуваннями системи, створювати повідомлення, запускати різні сервіси та виконувати інші дії. Однією з важливих функцій Siri є голосове введення тексту, що дозволяє

швидко створювати повідомлення або документи без використання клавіатури.

Подібні можливості має і голосовий помічник Amazon Alexa, який спочатку був створений для розумних пристроїв, але згодом отримав підтримку взаємодії з комп'ютерами та різними програмними сервісами. Alexa дозволяє запускати додатки, відтворювати мультимедійний контент, створювати списки завдань та керувати різними функціями системи за допомогою голосових команд.

Застосування таких голосових асистентів має низку важливих переваг. По-перше, вони роблять взаємодію з комп'ютером більш природною та інтуїтивною. По-друге, голосове керування може значно прискорити виконання певних дій, оскільки користувачу не потрібно вручну вводити команди або шукати потрібні функції в інтерфейсі програм. Крім того, такі системи є особливо корисними для людей з обмеженими можливостями, оскільки дозволяють користуватися комп'ютером без використання традиційних засобів введення.

Голосові помічники Google Assistant, Cortana, Apple Siri та Amazon Alexa демонструють ефективність використання технологій розпізнавання мовлення у сфері керування комп'ютерними системами. Вони забезпечують можливість виконання базових системних команд, пошуку інформації, створення нагадувань та взаємодію з іншими цифровими сервісами, запуску програм та введення тексту за допомогою голосу, що значно спрощує взаємодію користувача з сучасними інформаційними технологіями. Основною перевагою таких систем є висока точність розпізнавання та можливість обробки природної мови. Крім того, вони можуть постійно вдосконалюватися завдяки використанню хмарних сервісів і великих масивів даних.

Разом з тим, використання таких систем має певні недоліки. Зокрема, для їхньої роботи зазвичай необхідне постійне підключення до Інтернету. Крім того, існують питання конфіденційності та безпеки даних, оскільки голосові запити можуть оброблятися на віддалених серверах.

Також існують програмні рішення з відкритим програмним кодом, які дозволяють створювати власні системи голосового керування. До таких рішень належать різні бібліотеки та програмні платформи для розпізнавання мовлення, які можуть бути інтегровані у власні програмні проєкти. Вони надають розробникам більшу гнучкість та можливість адаптації системи під конкретні задачі. Перевагою таких рішень є можливість повного контролю над функціональністю системи, її налаштуванням та інтеграцією з іншими програмами. Однак їх використання потребує значних технічних знань та навичок програмування.

Аналіз існуючих систем голосового керування показує, що сучасні програмні рішення мають широкий функціонал та можуть застосовуватися для різних задач. Водночас кожна система має свої переваги та обмеження, пов'язані з точністю розпізнавання, вимогами до апаратного забезпечення, необхідністю підключення до Інтернету та можливостями налаштування. Саме тому при розробці власної системи голосового керування важливо враховувати особливості існуючих рішень та використовувати їхні переваги для створення ефективного та зручного програмного продукту.

1.3 Технології та інструменти для створення систем розпізнавання мовлення

Розробка систем розпізнавання мовлення потребує використання спеціальних технологій, програмних бібліотек та інструментів програмування. Такі інструменти дозволяють обробляти аудіосигнали, розпізнавати мовлення, аналізувати текст та перетворювати голосові команди у дії комп'ютерної системи.

Сучасні системи голосового керування зазвичай складаються з кількох основних компонентів: модуля обробки аудіосигналу, модуля розпізнавання мовлення, модуля обробки тексту та модуля виконання команд. Для реалізації

кожного з цих компонентів використовуються різні програмні інструменти та технології.

Одним із важливих елементів створення таких систем є програмні бібліотеки розпізнавання мовлення. Вони містять готові алгоритми для обробки звуку та визначення слів у мовному потоці. Використання таких бібліотек значно спрощує процес розробки, оскільки розробнику не потрібно створювати алгоритми розпізнавання з нуля (таблиця 1.5).

Таблиця 1.5 – Основні бібліотеки для розпізнавання мовлення

Бібліотека	Короткий опис	Переваги	Недоліки
SpeechRecognition	популярна бібліотека для Python, яка підтримує різні сервіси розпізнавання мовлення	проста у використанні, підтримує багато сервісів	залежність від сторонніх API
Vosk	бібліотека для офлайн розпізнавання мовлення	працює без інтернету, підтримує багато мов	потребує більше ресурсів
DeepSpeech	система розпізнавання мовлення на основі нейронних мереж	висока точність, сучасні алгоритми	складніше налаштування
CMU Sphinx	одна з перших систем розпізнавання мовлення з відкритим кодом	працює офлайн, гнучке налаштування	нижча точність порівняно з сучасними моделями

Бібліотеки SpeechRecognition, Vosk, DeepSpeech, CMU Sphinx використовуються для створення програм, які можуть приймати голосові команди користувача та перетворювати їх у текст.

Окрім бібліотек, для створення систем розпізнавання мовлення часто застосовуються фреймворки машинного навчання. Вони дозволяють створювати та навчати моделі штучного інтелекту, які здатні аналізувати мовні сигнали (таблиця 1.6).

Таблиця 1.6 – Фреймворки для роботи з алгоритмами розпізнавання мовлення

Фреймворк	Призначення	Особливості
TensorFlow	створення та навчання нейронних мереж	використовується для складних моделей штучного інтелекту
PyTorch	розробка та навчання моделей глибокого навчання	зручний для досліджень та експериментів
Keras	високорівнева бібліотека для роботи з нейронними мережами	простий синтаксис та швидка розробка

Завдяки фреймворкам TensorFlow, PyTorch, Keras можна створювати моделі, які здатні більш точно розпізнавати мовлення, навіть у складних умовах.

Не менш важливим елементом є мови програмування, які використовуються для реалізації систем голосового керування. Вибір мови програмування залежить від складності системи, вимог до швидкодії та доступних бібліотек (таблиця 1.7).

Таблиця 1.7 – Основні мови програмування для створення систем голосового керування

Мова програмування	Сфера використання	Переваги
Python	розробка систем штучного інтелекту та обробки мовлення	простота синтаксису, велика кількість бібліотек
JavaScript	створення веб-додатків з голосовим керуванням	підтримка у браузерях
C++	розробка високопродуктивних систем	висока швидкість роботи
Java	розробка мобільних та серверних систем	кросплатформеність

Найчастіше для створення систем розпізнавання мовлення використовується Python, оскільки ця мова має велику кількість бібліотек для роботи зі штучним інтелектом та обробкою аудіоданих.

Окрім програмних інструментів, важливу роль відіграють хмарні сервіси розпізнавання мовлення. Такі сервіси дозволяють використовувати потужні алгоритми обробки мовлення без необхідності створювати власні моделі (таблиця 1.8).

Таблиця 1.8 – Хмарні сервіси розпізнавання мовлення

Сервіс	Компанія	Основні можливості
Google Speech-to-Text	Google	висока точність розпізнавання, підтримка багатьох мов
Microsoft Azure Speech	Microsoft	інтеграція з сервісами Microsoft

Сервіс	Компанія	Основні можливості
Amazon Transcribe	Amazon	автоматичне розпізнавання та аналіз мовлення

Хмарні сервіси Google Speech-to-Text, Microsoft Azure Speech, Amazon Transcribe дозволяють швидко інтегрувати функції розпізнавання мовлення у програмні продукти, що значно спрощує процес створення голосових інтерфейсів.

Створення систем голосового керування потребує використання різноманітних технологій та інструментів. До них належать бібліотеки розпізнавання мовлення, фреймворки машинного навчання, мови програмування та хмарні сервіси. Використання цих інструментів дозволяє розробникам створювати ефективні системи голосового керування, які забезпечують зручну та природну взаємодію користувача з комп'ютером.

Висновок до розділу 1.

У розділі 1 було розглянуто теоретичні основи та сучасні підходи до створення систем голосового керування комп'ютером. Було визначено, що розпізнавання мовлення є багаторівневим процесом, який включає оцифрування аудіосигналу, його попередню обробку, виділення ознак та застосування алгоритмів розпізнавання. Для досягнення високої точності сучасні системи використовують як класичні статистичні методи, наприклад, приховані марковські моделі, так і сучасні алгоритми машинного навчання та нейронних мереж, у тому числі RNN, LSTM та трансформери.

Було проведено аналіз існуючих голосових помічників – Google Assistant, Cortana, Apple Siri та Amazon Alexa. Ці системи дозволяють користувачам керувати комп'ютером за допомогою голосу, виконувати базові системні команди, запускати програми, вводити текст та взаємодіяти з різними сервісами. При цьому вони значно підвищують зручність та ефективність

роботи користувача, а також відкривають можливості для автоматизації бізнес-процесів та інтеграції з «розумними» пристроями.

Було проведено огляд основних технологій та інструментів, які застосовуються при розробці систем розпізнавання мовлення: програмні бібліотеки (SpeechRecognition, Vosk, DeepSpeech, CMU Sphinx), фреймворки машинного навчання (TensorFlow, PyTorch, Keras), мови програмування (Python, C++, Java, JavaScript) та хмарні сервіси (Google Speech-to-Text, Microsoft Azure Speech, Amazon Transcribe). Ці інструменти забезпечують ефективну реалізацію алгоритмів голосового керування та спрощують процес розробки інтелектуальних систем.

Аналіз теоретичних основ і наявних технологічних рішень показав, що розробка ефективної системи голосового керування потребує комплексного підходу, який поєднує сучасні алгоритми розпізнавання мовлення, мовні моделі, програмні бібліотеки та фреймворки штучного інтелекту. Такий підхід забезпечує високу точність, гнучкість і можливість інтеграції системи у різні сфери використання.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ

2.1 Визначення функціональних та нефункціональних вимог до системи голосового керування

Перед початком розробки системи голосового керування необхідно визначити основні вимоги до програмного забезпечення. Формування вимог дозволяє чітко визначити, які функції повинна виконувати система, які обмеження існують під час її використання та які характеристики вона повинна мати. У процесі проєктування системи виділяють функціональні та нефункціональні вимоги. Функціональні вимоги описують, які дії повинна виконувати система. Нефункціональні вимоги визначають якість роботи системи, її швидкість, надійність та інші характеристики.

Функціональні вимоги визначають основні можливості системи голосового керування. Вони описують, які задачі повинна виконувати система під час роботи.

До основних функціональних вимог належать (таблиця 2.1):

- 1) прийом голосових команд користувача через мікрофон;
- 2) розпізнавання мовлення та перетворення аудіосигналу у текст;
- 3) аналіз тексту голосової команди;
- 4) визначення типу команди;
- 5) виконання відповідної системної дії;
- 6) взаємодія з операційною системою комп'ютера.

Таблиця 2.1 – Основні функціональні вимоги.

№	Функціональна вимога	Опис
1	Запис голосової команди	система повинна отримувати аудіосигнал з мікрофона

№	Функціональна вимога	Опис
2	Розпізнавання мовлення	перетворення голосу користувача у текст
3	Аналіз команди	визначення змісту голосової команди
4	Виконання команди	запуск програм або виконання системних дій
5	Зворотний зв'язок	повідомлення користувача про виконання команди

Нефункціональні вимоги описують характеристики роботи системи, які не пов'язані безпосередньо з виконанням функцій, але впливають на її ефективність та зручність використання (таблиця 2.2).

До основних нефункціональних вимог належать:

- 1) швидкість обробки голосових команд;
- 2) точність розпізнавання мовлення;
- 3) стабільність роботи системи;
- 4) простота використання;
- 5) можливість розширення функціоналу.

Таблиця 2.2 – Нефункціональні вимоги.

№	Нефункціональна вимога	Опис
1	Швидкодія	система повинна швидко обробляти голосові команди
2	Точність	розпізнавання мовлення повинно бути достатньо точним
3	Надійність	система повинна стабільно працювати без збоїв
4	Зручність використання	користувач повинен легко взаємодіяти з системою

№	Нефункціональна вимога	Опис
5	Масштабованість	можливість додавання нових голосових команд

Сценарії використання описують, як користувач взаємодіє із системою голосового керування під час виконання певних задач (таблиця 2.3).

Таблиця 2.3 – Основні сценарії використання системи

№	Сценарій використання	Опис
1	Запуск програми	користувач вимовляє команду, система відкриває програму
2	Відкриття системних утиліт	користувач запускає калькулятор або блокнот голосом
3	Виконання системної команди	система виконує команду керування комп'ютером
4	Введення тексту	система перетворює мовлення у текст

Наприклад, користувач може сказати команду «відкрити браузер», після чого система розпізнає мовлення, визначає команду та запускає відповідну програму.

2.2 Розробка архітектури системи

Архітектура системи голосового керування визначає структуру програмного забезпечення, основні компоненти системи та принципи їх взаємодії. Правильно спроектована архітектура дозволяє забезпечити

ефективну обробку голосових команд, підвищити стабільність роботи системи та спростити її подальше розширення або модифікацію. Система голосового керування комп'ютером зазвичай будується за модульним принципом. Це означає, що програмне забезпечення складається з окремих функціональних модулів, кожен з яких виконує певне завдання. Такий підхід дозволяє розділити складну систему на логічні частини, що спрощує розробку, тестування та підтримку програмного продукту. Основним завданням системи є прийом голосової команди користувача, її розпізнавання, інтерпретація та виконання відповідної дії на комп'ютері. Для реалізації цього процесу система складається з кількох основних модулів.

Таблиця 2.4 – Основні модулі системи голосового керування

Модуль	Призначення
Модуль захоплення аудіосигналу	отримує голосовий сигнал користувача через мікрофон
Модуль попередньої обробки	очищає сигнал від шумів та готує його до подальшого аналізу
Модуль розпізнавання мовлення	перетворює голосовий сигнал у текст
Модуль аналізу команд	визначає зміст команди та необхідну дію
Модуль виконання команд	запускає програми або виконує системні дії
Модуль взаємодії з операційною системою	забезпечує зворотний зв'язок із користувачем

Першим етапом роботи системи є отримання голосового сигналу. Для цього використовується мікрофон, який передає аудіосигнал до програмної частини системи. Цей сигнал може містити сторонні шуми або інші звукові перешкоди, тому наступним кроком є його попередня обробка. Для реалізації

цієї функції використовується програмна бібліотека *SpeechRecognition*, яка дозволяє отримувати аудіодані та передавати їх до модуля розпізнавання мовлення. Під час попередньої обробки сигналу здійснюється фільтрація шумів, нормалізація гучності та поділ сигналу на короткі часові інтервали, що дозволяє підвищити якість розпізнавання мовлення та зменшити ймовірність помилок. Після цього сигнал передається до модуля розпізнавання мовлення, який використовує спеціальні алгоритми для перетворення звуку у текстову форму. На цьому етапі застосовуються моделі машинного навчання або готові бібліотеки розпізнавання мовлення. Для цього використовується сервіс *Google Speech Recognition API*, який забезпечує достатньо високу точність розпізнавання мовлення. Отриманий текст надходить до модуля аналізу команд, де відбувається визначення наміру користувача. Система аналізує ключові слова або фрази та визначає, яку дію необхідно виконати. Наприклад, команда «відкрити браузер» буде інтерпретована як команда запуску відповідної програми.

Після визначення типу команди вона передається до модуля виконання команд, який взаємодіє з операційною системою комп'ютера. Цей модуль може виконувати різні дії, зокрема запускати програми, відкривати файли, керувати системними функціями або вводити текст у програмні додатки. Для цього використовується порівняння ключових слів команди з набором заздалегідь визначених команд.

Таблиця 2.5 – Приклад реалізації модуля обробки голосових команд

Голосова команда	Дія системи
«відкрити браузер»	запуск веб-браузера
«відкрити блокнот»	запуск текстового редактора
«відкрити калькулятор»	запуск системного калькулятора
«закрити програму»	завершення активного процесу

Якщо система знаходить відповідність між голосовою командою та визначеним шаблоном, вона передає відповідну дію до модуля виконання команд.

Для підвищення зручності використання система може містити модуль взаємодії з користувачем, який забезпечує зворотний зв'язок. Це може бути текстове повідомлення, звуковий сигнал або голосова відповідь системи, що підтверджує виконання команди.

Модуль виконання команд відповідає за реалізацію дій у системі. На цьому етапі текстова команда перетворюється у конкретну системну операцію.

Для реалізації цього функціоналу використовуються стандартні засоби мови програмування *Python*, зокрема бібліотеки *os* – для запуску програм; *subprocess* – для виконання системних команд; *pyautogui* – для керування діями користувача (натискання клавіш, керування мишею).

Модуль взаємодії з операційною системою забезпечує взаємодію системи голосового керування з операційною системою комп'ютера. Він виконує такі функції, як запуск програм; відкриття файлів; виконання системних команд; керування інтерфейсом користувача. Цей модуль фактично реалізує виконання команд, які були визначені попередніми модулями системи.

Модульна архітектура системи дозволяє легко додавати нові функції або вдосконалювати окремі компоненти. Наприклад, можна замінити алгоритм розпізнавання мовлення на більш сучасний або розширити набір голосових команд без зміни всієї системи.

Розроблена архітектура системи голосового керування забезпечує чітку організацію програмних компонентів та їх ефективну взаємодію. Модульний підхід дозволяє створити гнучку та масштабовану систему, здатну обробляти голосові команди користувача та виконувати відповідні дії у середовищі операційної системи.

Загальний алгоритм роботи розробленої системи голосового керування можна представити у вигляді послідовності етапів (Таблиця 2.6).

Таблиця 2.6 – Алгоритм роботи системи голосового керування

№ етапу	Опис
1	Користувач вимовляє голосову команду
2	Система записує аудіосигнал через мікрофон
3	Аудіосигнал передається до модуля розпізнавання мовлення
4	Голосова команда перетворюється у текст
5	Текстова команда аналізується системою
6	Визначається відповідна системна дія
7	Система виконує команду у середовищі ОС

Розроблена архітектура системи забезпечує ефективну обробку голосових команд користувача та їх виконання у середовищі операційної системи. Модульна структура програми дозволяє легко розширювати функціональність системи, додавати нові команди та вдосконалювати алгоритми розпізнавання мовлення.

2.3 Алгоритми розпізнавання та обробки голосових команд

Для реалізації системи голосового керування комп'ютером використовуються алгоритми, які забезпечують перетворення мовного сигналу у текст та подальшу інтерпретацію цього тексту як команд керування. На рисунку 2.1 наведено узагальнену схему алгоритму роботи системи.

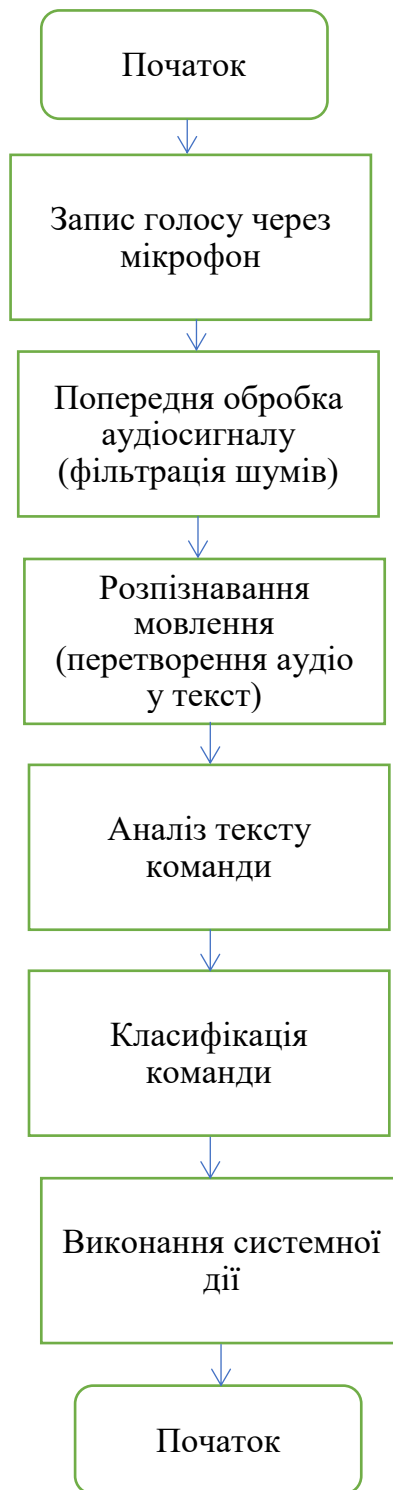


Рисунок 2.1 – Алгоритм обробки голосової команди

Основне завдання таких алгоритмів полягає у визначенні змісту голосової команди та виконанні відповідної дії в операційній системі. Загалом процес обробки голосових команд складається з кількох основних етапів: отримання аудіосигналу, попередньої обробки, розпізнавання мовлення,

класифікації команди та виконання відповідної дії. Кожен із цих етапів реалізується за допомогою окремих алгоритмів.

Алгоритм попередньої обробки аудіосигналу. Перед розпізнаванням мовлення аудіосигнал проходить етап попередньої обробки. Основною метою цього етапу є покращення якості сигналу та зменшення впливу сторонніх шумів (таблиця 2.7).

Таблиця 2.7 – Основні операції попередньої обробки наведено.

Етап	Опис
Фільтрація шуму	усунення сторонніх шумів із сигналу
Нормалізація	вирівнювання рівня гучності сигналу
Фреймування	поділ сигналу на короткі часові інтервали
Виділення ознак	визначення характеристик сигналу

У більшості систем для виділення характеристик мовного сигналу використовуються мел-частотні кепстральні коефіцієнти (MFCC), які добре відображають особливості людського мовлення.

Алгоритм розпізнавання мовлення. Після попередньої обробки аудіосигнал передається до алгоритму розпізнавання мовлення. У розробленій системі для цього використовується бібліотека SpeechRecognition, яка забезпечує взаємодію з сервісом Google Speech Recognition.

Алгоритм роботи модуля розпізнавання можна представити у вигляді такої послідовності, як отримання аудіосигналу; передача аудіофайлу до сервісу розпізнавання; аналіз спектральних характеристик сигналу; порівняння сигналу з мовними моделями; формування текстової інтерпретації команди.

Таблиця 2.8 – Основні моделі розпізнавання мовлення

Модель	Принцип роботи	Особливості
Приховані марковські моделі (HMM)	статистичне моделювання мовлення	використовуються у класичних системах
Нейронні мережі (DNN)	аналіз складних закономірностей сигналу	висока точність розпізнавання
Рекурентні мережі (RNN, LSTM)	врахування часової структури сигналу	добре працюють з послідовними даними

У сучасних системах найчастіше використовуються глибокі нейронні мережі, які забезпечують високу точність розпізнавання навіть у складних умовах.

Алгоритм класифікації голосових команд. Після отримання текстової інтерпретації мовлення система повинна визначити, яку дію необхідно виконати. Для цього використовується алгоритм класифікації команд. У розробленій системі застосовується метод пошуку ключових слів, який дозволяє швидко визначити тип команди.

Таблиця 2.9 – Таблиця відповідності голосових команд

Голосова команда	Тип команди	Виконувана дія
відкрити браузер	запуск програми	відкривається веб-браузер
відкрити блокнот	запуск програми	запускається текстовий редактор
відкрити калькулятор	системна команда	відкривається калькулятор
закрити програму	керування системою	завершується активний процес

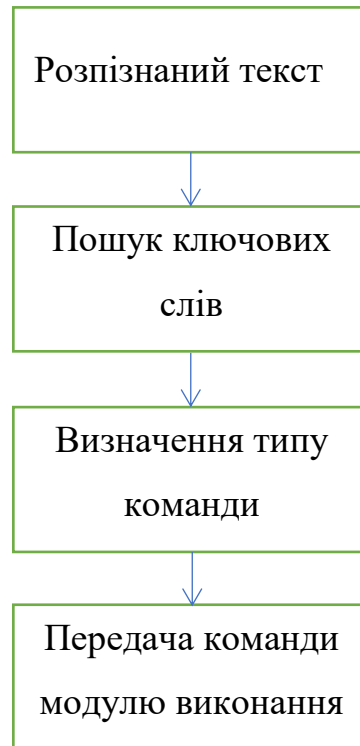


Рисунок 2.2 – Схема класифікації команд

Такий підхід дозволяє швидко обробляти голосові команди та виконувати відповідні дії. Алгоритм виконання команд. Після класифікації команда передається до модуля виконання. Цей модуль використовує системні функції операційної системи для реалізації необхідної дії.

Таблиця 2.10 – Основні дії, які підтримує система:

Тип дії	Опис
запуск програм	відкриття встановлених програм
керування файлами	відкриття документів
системні команди	запуск системних утиліт
введення тексту	передача тексту до активного додатку

Основним завданням цього модуля є перетворення текстової команди, отриманої після розпізнавання мовлення, у конкретну системну операцію. Для

цього використовується набір програмних інструкцій, які взаємодіють із функціями операційної системи та дозволяють запускати програми, відкривати файли або виконувати інші системні дії.

Процес виконання команди складається з кількох послідовних етапів. Спочатку система отримує текст команди після її класифікації. Далі відбувається перевірка, чи відповідає ця команда одному з визначених шаблонів у базі команд. Якщо відповідність знайдена, система визначає тип дії, яку необхідно виконати. Після цього формується відповідний системний виклик, який передається операційній системі для виконання.



Рисунок 2.3 – Загальна схема роботи алгоритмів системи

Алгоритм виконання команд можна представити у вигляді такої послідовності дій: отримання текстової команди після етапу класифікації → пошук відповідної команди у базі підтримуваних команд → визначення типу дії (запуск програми, відкриття файлу, виконання системної операції) → формування системного запиту → передача запиту операційній системі → виконання відповідної дії.

Для реалізації цих операцій у програмі використовуються стандартні можливості мови програмування та системні бібліотеки. Наприклад, у разі реалізації системи мовою *Python* можуть використовуватися бібліотеки *os*, *subprocess* або *pyautogui*, які дозволяють запускати програми, виконувати системні команди або керувати інтерфейсом користувача (таблиця 2.11).

Таблиця 2.11 – Приклад відповідності між голосовою командою та виконуваною дією

Голосова команда	Тип дії	Виконувана операція
відкрити браузер	запуск програми	відкриття веб-браузера
відкрити блокнот	запуск програми	запуск текстового редактора
відкрити калькулятор	системна команда	запуск калькулятора
закрити програму	керування системою	завершення активного процесу

Модуль виконання команд забезпечує безпосередню взаємодію системи голосового керування з операційною системою комп'ютера. Завдяки цьому користувач може керувати програмами та виконувати різні дії за допомогою голосових команд, що значно спрощує роботу з комп'ютером.

Алгоритми розпізнавання та обробки голосових команд забезпечують повний цикл взаємодії користувача з комп'ютером за допомогою голосу. Використання алгоритмів попередньої обробки аудіосигналу, моделей розпізнавання мовлення та методів класифікації команд дозволяє ефективно

перетворювати мовлення користувача у відповідні системні дії. Запропонований алгоритм забезпечує достатню швидкість роботи системи та можливість її подальшого розширення шляхом додавання нових голосових команд.

Висновок до розділу 2.

У розділі 2 було розглянуто основні етапи проєктування системи голосового керування комп'ютером. Спочатку було виконано формування вимог до системи, де визначено її функціональні та нефункціональні характеристики, а також основні сценарії використання, що дозволило встановити перелік задач, які повинна виконувати система, зокрема прийом голосових команд, їх розпізнавання, аналіз та виконання відповідних дій у середовищі операційної системи. Крім того, було визначено вимоги до швидкодії, точності розпізнавання та зручності використання системи.

Наступним етапом стала розробка архітектури системи. Було запропоновано модульну структуру програмного забезпечення, що складається з окремих компонентів: модуля захоплення аудіосигналу, модуля розпізнавання мовлення, модуля обробки команд та модуля виконання системних дій. Така архітектура забезпечує чітку організацію системи, ефективну взаємодію між її компонентами та можливість подальшого розширення функціональних можливостей. Також у розділі було розглянуто алгоритми розпізнавання та обробки голосових команд. Описано процес перетворення аудіосигналу у текст, аналіз текстової команди та її класифікацію. Для цього використовуються сучасні моделі розпізнавання мовлення та алгоритми обробки даних, які дозволяють ефективно інтерпретувати голосові команди користувача та перетворювати їх у відповідні дії системи. Отримані результати є основою для подальшої програмної реалізації системи та її практичного застосування.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ КОМП'ЮТЕРОМ

3.1 Розробка програмного забезпечення системи голосового керування комп'ютером

Розглянемо процес програмної реалізації системи голосового керування комп'ютером. Основною метою розробки є створення програмного забезпечення, яке дозволяє приймати голосові команди користувача, розпізнавати їх та виконувати відповідні дії в операційній системі.

Під час реалізації системи було обрано мову програмування *Python*, оскільки вона має простий синтаксис, велику кількість бібліотек для роботи зі штучним інтелектом та обробкою мовлення, а також дозволяє швидко створювати прототипи програмних систем.

Для створення системи голосового керування комп'ютером було використано сучасне середовище розробки *Visual Studio Code*. Це одне з найпопулярніших середовищ програмування, яке підтримує велику кількість мов програмування, зокрема Python, Java, C++, JavaScript та інші. Воно має зручний інтерфейс, високу швидкість роботи та широкі можливості для розширення функціоналу за допомогою додаткових модулів.

Visual Studio Code (VS Code) розроблено компанією Microsoft. Це безкоштовний редактор коду, який широко використовується як студентами, так і професійними розробниками програмного забезпечення. Однією з головних переваг цього середовища є підтримка великої кількості розширень, які дозволяють налаштувати робоче середовище відповідно до потреб проєкту.

Під час розробки системи голосового керування *Visual Studio Code* використовувався для створення програмного коду, його редагування, перевірки синтаксису та запуску програми. Також це середовище дозволяє зручно працювати зі структурою проєкту та швидко переходити між різними файлами програми.

Проаналізуємо основні можливості середовища розробки. Середовище Visual Studio Code має низку важливих функцій, які значно спрощують процес програмування (таблиця 3.1).

Таблиця 3.1 – Функції Visual Studio Code

Функція	Опис
Підсвічування синтаксису	автоматичне виділення елементів коду різними кольорами
Автодоповнення коду	підказки під час написання програмного коду
Налагодження програм	можливість знаходження та виправлення помилок
Робота з проєктами	зручне керування файлами та папками
Підтримка розширень	можливість встановлення додаткових модулів

Наприклад, під час написання коду на Python середовище автоматично пропонує варіанти завершення команд або функцій, що дозволяє зменшити кількість помилок і прискорює процес розробки.

Для ефективної роботи з мовою програмування Python у середовищі Visual Studio Code були встановлені спеціальні розширення (таблиця 3.2).

Таблиця 3.2 – Розширення у середовищі Visual Studio Code

Розширення	Призначення
Python	підтримка програмування на Python
Pylance	аналіз коду та автодоповнення
Code Runner	швидкий запуск програм
Python Debugger	налагодження програм

Наприклад, розширення Python дозволяє запускати програму безпосередньо з середовища розробки. Для цього достатньо натиснути кнопку запуску або використати команду Run Python File.

Після написання програмного коду файл можна запустити безпосередньо з Visual Studio Code. Для цього використовується вбудований термінал.

Приклад запуску програми:

```
python main.py
```

Після виконання цієї команди запускається основний файл програми, який активує систему голосового керування.

Ще однією важливою функцією середовища Visual Studio Code є наявність вбудованого терміналу, який дозволяє виконувати різні системні команди без необхідності відкривати додаткові програми.

Приклад встановлення бібліотек у терміналі:

```
pip install SpeechRecognition
```

```
pip install PyAudio
```

Ці команди встановлюють необхідні бібліотеки для роботи системи розпізнавання мовлення. У середовищі Visual Studio Code всі файли проєкту відображаються у вигляді дерева каталогів. Це дозволяє легко організувати структуру програмного забезпечення.

Приклад структури проєкту (рисунок 3.1):

```
voice_control_system
|
├── main.py
├── audio_module.py
├── recognition_module.py
├── command_module.py
└── execution_module.py
```

Рисунок 3.1 – Структура проєкту

Така організація дозволяє розділити програму на окремі модулі, що значно спрощує підтримку та подальший розвиток системи.

Використання Visual Studio Code під час розробки системи голосового керування має кілька таких важливих переваг, як простий та зрозумілий інтерфейс; підтримка великої кількості мов програмування; можливість встановлення додаткових розширень; зручна робота зі структурою проєкту; інтеграція з системами контролю версій. Завдяки цим можливостям середовище Visual Studio Code є ефективним інструментом для створення програмного забезпечення, зокрема систем голосового керування.

Використання сучасного середовища розробки дозволило значно спростити процес створення програмного забезпечення, підвищити якість програмного коду та забезпечити зручну організацію структури програмного проєкту.

Модуль запису аудіо є одним із основних компонентів системи голосового керування. Його головним завданням є отримання голосової команди користувача за допомогою мікрофона та передача записаного аудіосигналу до наступного етапу обробки. Під час роботи системи цей модуль постійно очікує появи голосової команди. Коли користувач вимовляє команду, мікрофон комп'ютера фіксує звуковий сигнал і передає його до програми. Після цього аудіосигнал записується у тимчасовий об'єкт або файл, який надалі використовується для розпізнавання мовлення. У розробленій системі для запису звуку використовується бібліотека SpeechRecognition, яка надає зручні інструменти для роботи з аудіосигналами. Вона дозволяє отримувати звук із мікрофона, керувати тривалістю запису та передавати аудіодані до модулів розпізнавання мовлення. Процес роботи модуля запису аудіо можна описати такими етапами:

- 1) ініціалізація об'єкта розпізнавання мовлення;
- 2) підключення до мікрофона комп'ютера;
- 3) очікування голосової команди користувача;
- 4) запис аудіосигналу;
- 5) передача отриманого аудіо до модуля розпізнавання мовлення.

На рисунку 3.1 схематично поредставлено роботу модуля запису аудіо.



Рисунок 3.2 – Робота модуля запису аудіо.

У програмному коді цей модуль реалізований на мові Python за допомогою функції, яка використовує мікрофон для запису голосу користувача.

```
import speech_recognition as sr

def record_audio():
    recognizer = sr.Recognizer()

    with sr.Microphone() as source:
        print("Слухаю команду...")
        audio = recognizer.listen(source)

    return audio
```

У цьому фрагменті коду створюється об'єкт розпізнавання, після чого програма підключається до мікрофона та починає запис аудіосигналу. Коли користувач вимовляє команду, функція `listen()` записує звук і повертає

аудіооб'єкт, який передається до модуля розпізнавання мовлення. Модуль запису аудіо забезпечує отримання голосових команд користувача та є першим етапом роботи системи голосового керування. Від якості запису аудіосигналу залежить точність подальшого розпізнавання мовлення та ефективність роботи всієї системи.

Модуль розпізнавання мовлення є ключовим компонентом системи голосового керування комп'ютером. Основним завданням цього модуля є перетворення аудіосигналу, отриманого з мікрофона, у текстову форму, яку система може аналізувати та використовувати для виконання відповідних команд. Після того як модуль запису аудіо отримує голосову команду користувача, аудіосигнал передається до модуля розпізнавання мовлення. На цьому етапі відбувається аналіз звукових даних, виділення мовних ознак та їх подальше перетворення у текст.

У розробленій системі для реалізації цього процесу використовується бібліотека SpeechRecognition, яка забезпечує доступ до сервісів автоматичного розпізнавання мовлення. Зокрема, застосовується сервіс Google Speech Recognition, який використовує сучасні алгоритми машинного навчання для визначення слів та фраз у мовному сигналі. Процес роботи модуля розпізнавання мовлення можна подати у вигляді послідовних етапів:

- 1) отримання аудіосигналу від модуля запису;
- 2) попередня обробка звукових даних;
- 3) передача аудіосигналу до сервісу розпізнавання;
- 4) аналіз мовного сигналу;
- 5) формування текстового представлення голосової команди;
- 6) передача тексту до модуля обробки команд.

Схема роботи модуля розпізнавання мовлення схематично представлена на рисунку 3.3.

Основною функцією цього модуля є перетворення аудіооб'єкта у текст. Для цього використовується спеціальний метод бібліотеки, який виконує аналіз мовлення та повертає текстовий результат.



Рисунок 3.3 – Робота модуля розпізнавання мовлення

Приклад реалізації модуля розпізнавання мовлення на мові Python:

```
import speech_recognition as sr

def recognize_speech(audio):
    recognizer = sr.Recognizer()

    try:
        command = recognizer.recognize_google(audio, language="uk-UA")
        print("Ви сказали:", command)
        return command.lower()
```

except sr.UnknownValueError:

```
print("Команду не розпізнано")  
return ""
```

except sr.RequestError:

```
print("Помилка сервісу розпізнавання")  
return ""
```

У цьому коді створюється об'єкт Recognizer, який виконує розпізнавання мовлення. Метод recognize_google() передає аудіосигнал до сервісу Google для обробки. Після завершення аналізу система отримує текстову форму голосової команди. Якщо сервіс не може розпізнати мовлення або виникає помилка підключення, програма обробляє відповідні винятки та повідомляє користувача про проблему. Основні функції модуля узагальнено у таблиці 3.3

Таблиця 3.3 – . Основні функції модуля розпізнавання мовлення

Функція модуля	Опис
Отримання аудіосигналу	приймання записаного звуку від модуля запису
Аналіз мовного сигналу	обробка аудіоданих для визначення слів
Перетворення мовлення у текст	формування текстової команди
Передача результату	передача тексту до модуля обробки команд

Модуль розпізнавання мовлення забезпечує перетворення голосових команд користувача у текстовий формат, що дозволяє системі виконувати подальший аналіз та реалізовувати відповідні дії. Ефективність роботи цього модуля безпосередньо впливає на точність та швидкість функціонування всієї системи голосового керування комп'ютером.

Модуль обробки команд призначений для аналізу текстової команди, яка була отримана після етапу розпізнавання мовлення. Основним завданням цього модуля є визначення змісту команди користувача та перетворення її у конкретну дію, яку повинна виконати система. Після того як модуль розпізнавання мовлення перетворює аудіосигнал у текст, формується текстова команда. Саме на цьому етапі відбувається передача текстової команди до модуля обробки. Передача здійснюється у вигляді текстового рядка (string), який містить слова, сказані користувачем.

Наприклад, після розпізнавання мовлення система може отримати такі текстові команди, як «відкрити браузер», «відкрити калькулятор», «відкрити блокнот». Ці текстові дані передаються у функцію аналізу, яка перевіряє наявність ключових слів та визначає тип команди. Схема роботи модуля обробки команд схематично представлена на рисунку 3.4.

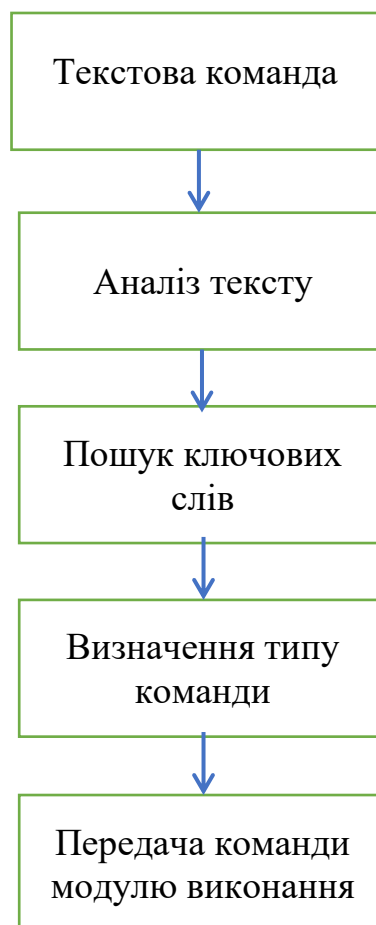


Рисунок 3.4 – Робота модуля обробки команд

Процес роботи модуля обробки команд включає такі етапи:

- 1) отримання текстової команди від модуля розпізнавання мовлення;
- 2) аналіз отриманого тексту;
- 3) пошук ключових слів або фраз;
- 4) визначення типу команди;
- 5) передача визначеної команди до модуля виконання.

У розробленій системі аналіз команди виконується за допомогою перевірки наявності певних слів у тексті. Якщо система знаходить відповідне слово або фразу, вона визначає тип команди та повертає відповідний код дії.

Приклад реалізації модуля обробки команд на мові Python:

```
def analyze_command(command):
```

```
    if "браузер" in command:
```

```
        return "open_browser"
```

```
    elif "калькулятор" in command:
```

```
        return "open_calculator"
```

```
    elif "блокнот" in command:
```

```
        return "open_notepad"
```

```
    elif "вийти" in command:
```

```
        return "exit"
```

```
    else:
```

```
        return "unknown"
```

У цьому фрагменті програми функція `analyze_command()` отримує текстову команду користувача та перевіряє, чи містить вона певні ключові

слова. Якщо слово знайдено, функція повертає спеціальний ідентифікатор команди, який потім використовується модулем виконання.

Таблиця 3.4 – Основні функції модуля обробки команд

Функція	Опис
Отримання текстової команди	приймання результату розпізнавання мовлення
Аналіз тексту	перевірка змісту команди
Визначення ключових слів	пошук слів, що відповідають певним діям
Формування команди	створення коду або типу команди
Передача команди	передача результату модулю виконання

Модуль обробки команд виконує роль проміжної ланки між розпізнаванням мовлення та виконанням системних дій. Він дозволяє системі інтерпретувати текст голосової команди та визначати, яку саме дію необхідно виконати в операційній системі.

Модуль виконання команд є завершальним етапом роботи системи голосового керування комп'ютером. Його основне призначення полягає у виконанні конкретних дій в операційній системі на основі команди, яка була визначена на попередньому етапі аналізу. Після того як текстова команда користувача була проаналізована модулем обробки команд, система отримує визначений тип команди. Ця команда передається до модуля виконання, який відповідає за запуск програм, відкриття веб-сторінок або виконання інших системних дій. Іншими словами, якщо попередній модуль визначає що саме хоче зробити користувач, то модуль виконання реалізує фактичне виконання цієї дії у середовищі операційної системи. Процес роботи модуля виконання команд можна описати такими етапами:

- 1) отримання ідентифікатора команди від модуля обробки;
- 2) визначення відповідної системної дії;
- 3) виклик системної функції операційної системи;
- 4) виконання необхідної операції.

Схема роботи модуля виконання команд схематично представлена на рисунку 3.5.

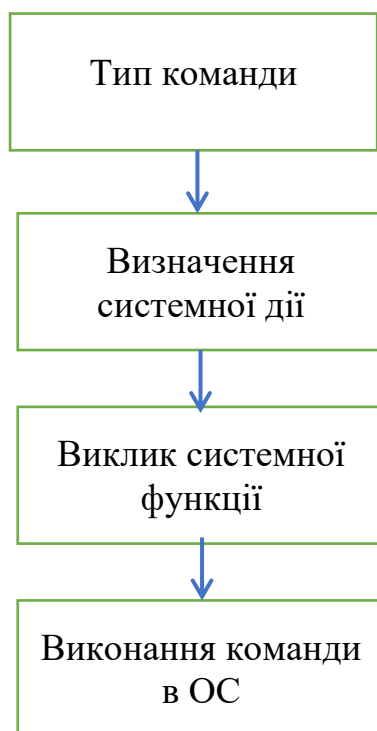


Рисунок 3.5 – Робота модуля виконання команд

У розробленій системі виконання команд реалізовано за допомогою стандартних бібліотек мови програмування Python, таких як `os`, `subprocess` та `webbrowser`. Ці бібліотеки дозволяють взаємодіяти з операційною системою та запускати різні програми. Приклад реалізації модуля виконання команд на мові Python:

```
import os
import webbrowser

def execute_command(command):

    if command == "open_browser":
        webbrowser.open("https://www.google.com")

    elif command == "open_calculator":
        os.system("calc")
```

```

elif command == "open_notepad":
    os.system("notepad")

elif command == "exit":
    print("Завершення роботи програми")
    exit()

else:
    print("Команда не розпізнана")

```

У цьому прикладі функція `execute_command()` отримує тип команди та визначає, яку саме дію потрібно виконати. Наприклад:

команда `open_browser` відкриває веб-браузер;
 команда `open_calculator` запускає програму калькулятора;
 команда `open_notepad` відкриває текстовий редактор;
 команда `exit` завершує роботу програми.

Основні функції модуля виконання команд узагальнено у таблиці 3.5.

Таблиця 3.5 – Основні функції модуля виконання команд

Функція модуля	Опис
Отримання команди	приймання визначеного типу команди
Визначення дії	встановлення відповідної системної операції
Виклик функцій ОС	використання системних команд
Виконання операції	запуск програм або виконання дій

Модуль виконання команд забезпечує реалізацію дій, які відповідають голосовим командам користувача. Завдяки цьому система голосового керування може взаємодіяти з операційною системою та виконувати різні завдання без використання клавіатури або миші.

Основний модуль системи є центральним елементом програмного забезпечення, який координує роботу всіх інших модулів системи голосового керування. Саме цей модуль забезпечує взаємодію між окремими компонентами програми та визначає загальний алгоритм роботи системи. Головним завданням основного модуля є послідовний виклик інших функціональних модулів: запису аудіо, розпізнавання мовлення, обробки команд та виконання системних дій. Іншими словами, він керує усім процесом від отримання голосової команди користувача до виконання відповідної дії на комп'ютері. Під час роботи системи основний модуль запускає програму та переходить у режим постійного очікування голосових команд. Коли користувач вимовляє команду, система починає обробку сигналу за визначеним алгоритмом. Схема роботи основного модуля системи схематично представлена на рисунку 3.6.

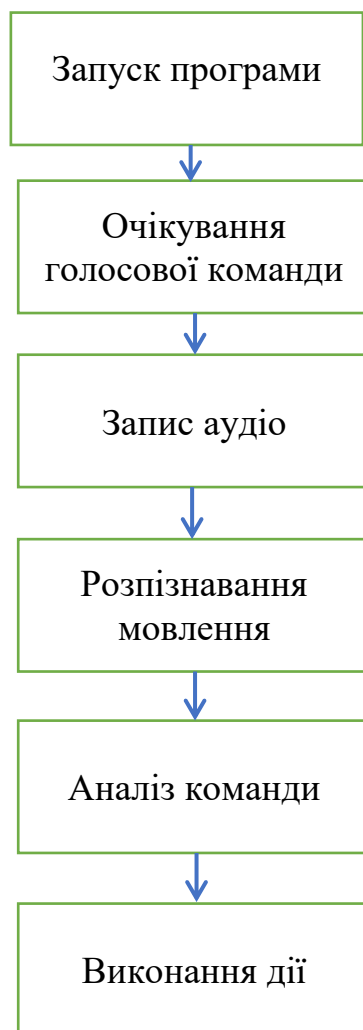


Рисунок 3.6 – Робота основного модуля системи

Основні етапи роботи основного модуля такі:

- 1) запуск програми;
- 2) виклик модуля запису аудіо;
- 2) передача аудіосигналу до модуля розпізнавання мовлення;
- 4) отримання текстової команди;
- 5) аналіз текстової команди;
- 6) передача визначеної команди до модуля виконання;
- 7) виконання відповідної системної дії.

Основний модуль також забезпечує циклічну роботу системи, що дозволяє програмі постійно приймати нові голосові команди. Для цього використовується цикл, який повторює всі етапи роботи системи до моменту завершення програми. Приклад реалізації основного модуля на мові Python:

```
from audio_module import record_audio
from recognition_module import recognize_speech
from command_module import analyze_command
from execution_module import execute_command
```

```
def main():
```

```
    while True:
```

```
        audio = record_audio()
```

```
        text_command = recognize_speech(audio)
```

```
        if text_command == "":
```

```
            continue
```

```
        command_type = analyze_command(text_command)
```

```
execute_command(command_type)
```

```
if __name__ == "__main__":  
    main()
```

У цьому фрагменті коду функція main() є основною точкою входу в програму. Вона послідовно викликає функції інших модулів та забезпечує безперервну роботу системи.

Цикл while True дозволяє програмі постійно очікувати нові голосові команди користувача. Якщо команда була успішно розпізнана, вона передається на аналіз, після чого виконується відповідна системна дія. Основні функції основного модуля наведено у таблиці 3.6.

Таблиця 3.6 – Основні функції основного модуля.

Функція	Опис
Координація роботи системи	керування взаємодією між модулями
Запуск процесу обробки	ініціалізація роботи системи
Виклик функціональних модулів	послідовне виконання операцій
Організація циклу роботи	забезпечення постійного прийому команд

Основний модуль системи виконує роль центрального керуючого елемента, який об'єднує всі модулі програми в єдину систему. Він забезпечує послідовну обробку голосових команд та організовує взаємодію між компонентами програмного забезпечення. Алгоритм роботи програмної системи складається з 6 етапів (таблиця 3.7)

Таблиця 3.7 – Алгоритм роботи програмної системи

№	Етап
1	система очікує голосову команду

№	Етап
2	записується аудіосигнал
3	виконується розпізнавання мовлення
4	текст команди аналізується
5	визначається тип команди
6	виконується відповідна дія

Розроблене програмне забезпечення реалізує систему голосового керування комп'ютером, яка дозволяє користувачу виконувати базові системні дії за допомогою голосових команд. Використання мови програмування Python та відповідних бібліотек забезпечує простоту реалізації, гнучкість системи та можливість її подальшого розширення.

3.2 Інтеграція системи з операційною системою та прикладними програмами

Одним із важливих етапів розробки системи голосового керування є її інтеграція з операційною системою та прикладними програмами. Інтеграція забезпечує можливість виконання різних дій на комп'ютері на основі голосових команд користувача. До таких дій належать запуск програм, керування системними функціями, робота з файлами, а також взаємодія з іншими елементами операційної системи. Розроблена система реалізує взаємодію з операційною системою за допомогою стандартних бібліотек мови програмування *Python*, які дозволяють виконувати системні команди та керувати програмами. Основними бібліотеками, що використовуються для інтеграції, є *os*, *subprocess*, *webbrowser* та *shutil*.

Розглянемо механізм взаємодії голосових команд з операційною системою. Після того як голосова команда користувача була розпізнана та проаналізована, система визначає тип дії, яку потрібно виконати. Далі ця

команда передається до модуля виконання, який викликає відповідні функції операційної системи. Загальний механізм взаємодії системи з операційною системою можна представити у вигляді послідовності етапів (рисунок 3.7).

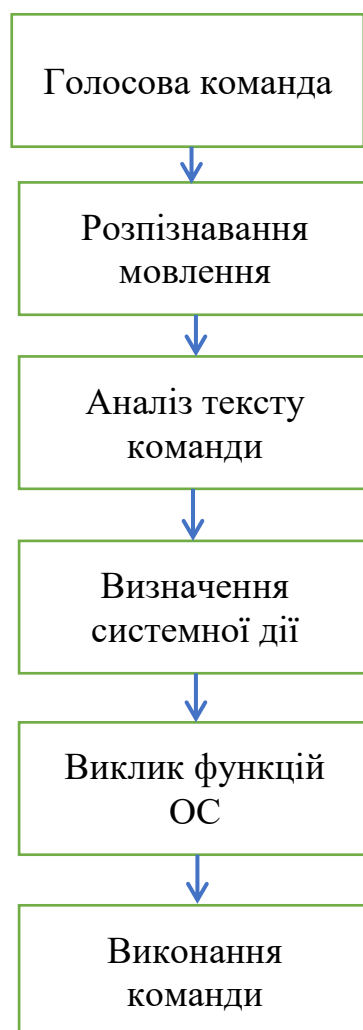


Рисунок 3.7 – Загальний механізм взаємодії системи з операційною системою.

Такий підхід дозволяє перетворити голосову команду користувача на конкретну дію в операційній системі.

Однією з основних функцій системи голосового керування є *запуск прикладних програм*. Користувач може відкривати різні програми, вимовляючи відповідну голосову команду.

Для запуску програм використовується бібліотека `os`, яка дозволяє виконувати системні команди операційної системи. Приклад реалізації запуску програм на мові Python:

```
import os

def open_application(command):

    if command == "open_calculator":
        os.system("calc")

    elif command == "open_notepad":
        os.system("notepad")

    elif command == "open_paint":
        os.system("mspaint")
```

У цьому прикладі система перевіряє тип команди та запускає відповідну програму операційної системи.

Керування системними функціями. Система може виконувати деякі системні функції, наприклад, відкриття налаштувань системи; завершення роботи програми; керування веб-браузером; виконання системних команд. Для цього можуть використовуватися стандартні системні команди операційної системи. Приклад реалізації на мові Python:

```
import os

def system_control(command):

    if command == "shutdown":
        os.system("shutdown /s /t 0")

    elif command == "restart":
        os.system("shutdown /r /t 0")
```

Такі команди дозволяють керувати роботою комп'ютера за допомогою ГОЛОСОВИХ КОМАНД.

Ще однією важливою функцією системи є можливість виконання операцій з файлами. Це може включати відкриття файлів, створення нових документів або переміщення файлів між папками. Для роботи з файлами використовуються стандартні можливості Python. Приклад відкриття файлу на мові Python:

```
import os

def open_file(path):
    os.startfile(path)
```

У цьому випадку система відкриває файл за вказаним шляхом за допомогою стандартної програми операційної системи.

Система голосового керування також може відкривати веб-сторінки в браузері. Для цього використовується бібліотека *webbrowser*. Приклад відкриття веб-сторінки на мові Python:

```
import webbrowser

def open_website():

    webbrowser.open("https://www.google.com")
```

Ця функція дозволяє відкривати потрібний веб-сайт після відповідної голосової команди користувача.

Основні функції інтеграції системи голосового керування з операційною системою наведено у таблиці 3.8.

Таблиця 3.8 – Основні можливості інтеграції системи

Функція системи	Опис
Запуск програм	відкриття прикладних програм операційної системи
Керування системою	виконання системних команд
Робота з файлами	відкриття та обробка файлів

Функція системи	Опис
Робота з веб-ресурсами	відкриття веб-сайтів у браузері
Керування програмами	взаємодія з запущеними додатками

Інтеграція системи голосового керування з операційною системою дозволяє значно розширити функціональні можливості програмного забезпечення. Користувач отримує можливість виконувати різні дії на комп'ютері без використання клавіатури або миші.

До основних переваг такого підходу належать підвищення зручності роботи з комп'ютером; швидкий доступ до програм та файлів; автоматизація виконання типових операцій; можливість використання системи людьми з обмеженими фізичними можливостями.

Інтеграція системи голосового керування з операційною системою забезпечує ефективну взаємодію користувача з комп'ютером та дозволяє реалізувати виконання різноманітних команд за допомогою голосового інтерфейсу.

3.3 Сценарії використання та приклади роботи системи

Розроблена система голосового керування призначена для спрощення взаємодії користувача з комп'ютером за допомогою голосових команд. Практичне використання системи передбачає виконання типових дій операційної системи без використання клавіатури або миші. До таких дій належать запуск програм, відкриття веб-ресурсів, робота з файлами та виконання системних команд.

У процесі роботи користувач вимовляє голосову команду, після чого система виконує її розпізнавання, аналізує текст команди та виконує відповідну дію в операційній системі.

Алгоритм роботи системи під час обробки голосової команди складається з кількох послідовних етапів (рисунок 3.8).

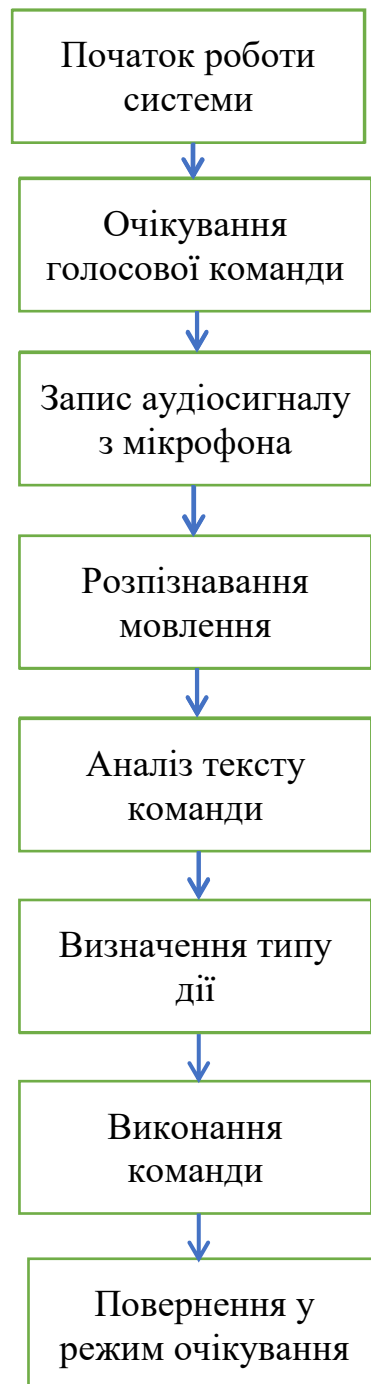


Рисунок 3.8 – Алгоритм роботи системи під час обробки голосової команди

Такий алгоритм дозволяє системі безперервно приймати та виконувати нові голосові команди користувача.

Основні сценарії використання системи. Розроблена система підтримує декілька типових сценаріїв використання. Найбільш поширені з них наведено у таблиці 3.9.

Таблиця 3.9 – Основні сценарії використання системи

№	Сценарій використання	Опис
1	Запуск програм	користувач відкриває стандартні програми операційної системи
2	Робота з веб-ресурсами	відкриття сайтів у браузері
3	Виконання системних команд	керування роботою комп'ютера
4	Завершення роботи програми	завершення роботи системи голосового керування

Система використовує набір ключових голосових команд, які відповідають певним діям. Приклади таких команд наведено у таблиці 3.10.

Таблиця 3.10 – Приклади голосових команд

Голосова команда	Дія системи
«відкрити браузер»	запуск веб-браузера
«відкрити калькулятор»	запуск програми калькулятора
«відкрити блокнот»	запуск текстового редактора
«відкрити Google»	відкриття веб-сайту
«вийти»	завершення роботи програми

Розглянемо приклад роботи системи при виконанні голосової команди «відкрити калькулятор».

Алгоритм обробки команди (рисунок 3.9):

користувач вимовляє команду «відкрити калькулятор» → мікрофон записує аудіосигнал → модуль розпізнавання мовлення перетворює аудіосигнал у текст → система отримує текстову команду → модуль обробки команд визначає ключове слово «калькулятор» → система формує команду `open_calculator` → модуль виконання команд запускає програму калькулятора.

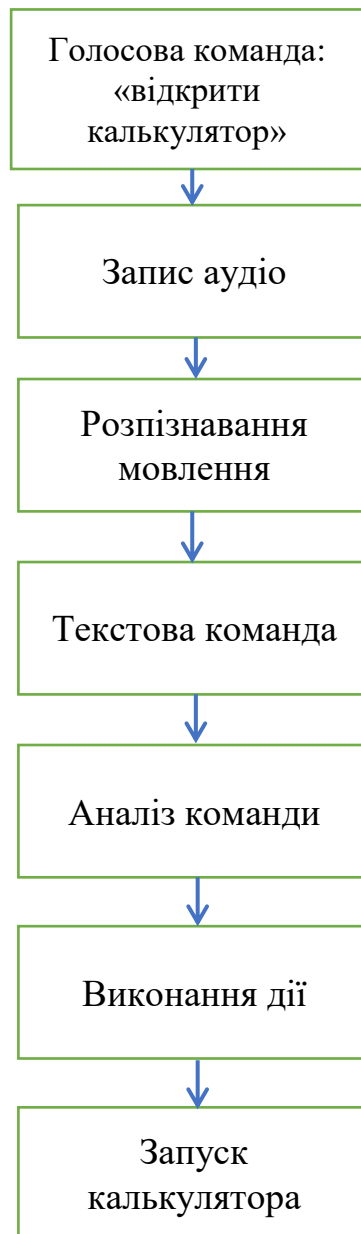


Рисунок 3.9 – Схема обробки голосової команди «відкрити калькулятор».

Таблиця 3.11 – Приклад взаємодії користувача з системою голосового керування.

Крок	Дія користувача	Реакція системи
1	запуск програми	система переходить у режим очікування
2	користувач говорить «відкрити браузер»	система записує аудіо

Крок	Дія користувача	Реакція системи
3	система розпізнає мовлення	формується текстова команда
4	команда аналізується	визначається тип дії
5	система виконує команду	відкривається браузер

Приклад роботи програми у середовищі виконання на мові Python:

Система запущена

Очікування голосової команди...

Користувач: "відкрити браузер"

Розпізнана команда: відкрити браузер

Виконання команди...

Браузер відкрито

Після виконання команди система знову переходить у режим очікування нової голосової команди.

Використання розробленої системи голосового керування має кілька важливих переваг. А саме, спрощення взаємодії користувача з комп'ютером; швидкий доступ до програм та функцій операційної системи; можливість керування комп'ютером без використання клавіатури або миші; підвищення зручності роботи з комп'ютером.

Розроблена система голосового керування дозволяє ефективно виконувати базові операції комп'ютера за допомогою голосових команд. Використання таких систем відкриває нові можливості для взаємодії людини з комп'ютерними технологіями та сприяє подальшому розвитку інтелектуальних інтерфейсів керування.

Висновок до розділу 3.

У розділі 3 було розглянуто основні сценарії використання розробленої системи голосового керування комп'ютером та наведено приклади її практичної роботи. Описано процес взаємодії користувача із системою, починаючи від введення голосової команди та завершуючи виконанням відповідної дії в операційній системі.

Було представлено приклади типових голосових команд, які дозволяють запускати прикладні програми, відкривати веб-ресурси, а також керувати окремими функціями операційної системи. Зокрема, продемонстровано виконання команд для відкриття браузера, текстового редактора та калькулятора. Для кожного прикладу описано послідовність обробки команди, що включає запис аудіосигналу, розпізнавання мовлення, аналіз текстової команди та виконання відповідної системної дії.

Результати розглянутих сценаріїв підтверджують можливість практичного використання розробленої системи для виконання базових операцій комп'ютера за допомогою голосових команд. Запропонований підхід забезпечує зручну взаємодію користувача з комп'ютером та може бути використаний як основа для подальшого розширення функціональних можливостей системи голосового керування.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Організаційно-правові основи забезпечення безпеки праці

Організаційно-правові основи забезпечення безпеки праці є важливою складовою під час розробки та експлуатації інтелектуальних інформаційних систем, зокрема систем голосового керування комп'ютером із застосуванням алгоритмів розпізнавання мовлення. Виконання робіт, пов'язаних із програмуванням, налаштуванням програмного забезпечення, тестуванням алгоритмів та використанням комп'ютерної техніки, потребує створення безпечних та комфортних умов праці.

Правову основу охорони праці в Україні становлять Конституція України, Кодекс законів про працю України, Закон України «Про охорону праці», Закон України «Про пожежну безпеку», Закон України «Про загальнообов'язкове державне соціальне страхування», а також державні санітарні норми та правила, нормативні документи з електробезпеки та пожежної безпеки. Відповідно до Закону України «Про охорону праці», роботодавець зобов'язаний створити безпечні та нешкідливі умови праці, організувати контроль за станом охорони праці, проводити інструктажі та навчання працівників, а також забезпечити відповідність робочих місць встановленим вимогам.

Основними принципами державної політики у сфері охорони праці є пріоритет життя та здоров'я працівників; відповідальність роботодавця за створення безпечних умов праці; профілактика виробничого травматизму; використання сучасних засобів захисту; комплексне вирішення питань безпеки праці.

Організаційно-правові основи забезпечення безпеки праці під час розробки інтелектуальної системи голосового керування комп'ютером базуються на дотриманні вимог чинного законодавства, створенні безпечних умов праці, забезпеченні електро- та пожежної безпеки, а також захисті

інформаційних ресурсів. Реалізація цих заходів дозволяє мінімізувати професійні ризики та забезпечити ефективну і безпечну роботу працівників.

4.2 Характеристика об'єкта та виявлення потенційних небезпек

Під час розробки програмного забезпечення та роботи з комп'ютерною технікою важливим аспектом є забезпечення безпечних умов праці користувача. Розробка програмного забезпечення та тестування системи голосового керування здійснюється у приміщенні з використанням персонального комп'ютера, мікрофона та програмного середовища розробки. Тому основні небезпеки пов'язані з роботою за комп'ютером, електробезпекою, організацією робочого місця та впливом факторів виробничого середовища.

Під час роботи з комп'ютерною технікою можуть виникати різні небезпечні та шкідливі фактори. До основних факторів, які можуть впливати на користувача, належать фізичні, психофізіологічні та організаційні фактори (таблиця 4.1).

Таблиця 4.1 – Основні потенційні небезпеки

№	Небезпечний або шкідливий фактор	Джерело виникнення	Можливі наслідки
1	Тривала робота за комп'ютером	використання монітора	втома очей, погіршення зору
2	Неправильна організація робочого місця	невідповідна висота столу або крісла	порушення постави, біль у спині
3	Електричний струм	комп'ютерне обладнання	ризик ураження електричним струмом
4	Підвищене навантаження на нервову систему	інтенсивна робота з програмним кодом	втома, зниження працездатності
5	Недостатнє освітлення	неправильне	напруження зору

№	Небезпечний або шкідливий фактор	Джерело виникнення	Можливі наслідки
	робочого місця	освітлення приміщення	

До фізичних факторів належать вплив електромагнітного випромінювання від комп'ютерного обладнання, недостатній рівень освітлення, а також тривала робота з монітором. Хоча сучасні монітори мають низький рівень випромінювання, тривале перебування перед екраном може викликати втоми очей та зниження концентрації уваги.

Також важливим фактором є рівень освітлення робочого місця. Недостатнє або неправильно організоване освітлення може призводити до підвищеного навантаження на зоровий апарат.

Ергономічні фактори пов'язані з організацією робочого місця користувача. Неправильне розташування монітора, клавіатури або крісла може призводити до порушення постави, болю в спині та шії (таблиця 4.2).

Таблиця 4.2 – Основні ергономічні ризики

Фактор	Причина	Можливі наслідки
Неправильне положення монітора	невідповідна висота екрана	напруження очей
Невідповідне крісло	відсутність регулювання висоти	біль у спині
Тривала робота без перерв	відсутність режиму відпочинку	перевтома

Психофізіологічні фактори. Під час розробки програмного забезпечення значне навантаження припадає на розумову діяльність. Тривала концентрація уваги, робота з великими обсягами інформації та необхідність пошуку помилок у програмному коді можуть призводити до підвищеної втоми, зниження продуктивності праці та нервового напруження.

Електробезпека. Під час роботи з комп'ютерним обладнанням існує ризик ураження електричним струмом. Це може статися у випадку пошкодження кабелів, несправності електрообладнання або порушення правил експлуатації техніки. До основних причин виникнення електробезпеки належать пошкодження ізоляції електричних проводів; використання несправного обладнання; неправильне підключення техніки до електромережі; відсутність заземлення. Узагальнення основних небезпечних факторів під час роботи з комп'ютером наведено у таблиці 4.3.

Таблиця 4.3 – Основні небезпечні фактори під час роботи з комп'ютером

Група факторів	Основні ризики
Фізичні	втома очей, вплив випромінювання
Ергономічні	порушення постави, біль у спині
Психофізіологічні	перевтома, нервове напруження
Електричні	можливість ураження електричним струмом

Під час розробки та використання системи голосового керування комп'ютером основні небезпеки пов'язані з роботою за персональним комп'ютером та організацією робочого місця користувача. Тому важливим завданням є дотримання вимог охорони праці, забезпечення правильного розташування обладнання та дотримання режиму праці й відпочинку, що дозволить зменшити негативний вплив шкідливих факторів та забезпечити безпечні умови роботи.

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проєктування та розробка заходів щодо їх попередження

Під час виконання робіт, пов'язаних із розробкою програмного забезпечення системи голосового керування комп'ютером, важливим етапом є проведення аналізу та оцінки ризику. Метою такого аналізу є визначення рівня

небезпеки потенційних шкідливих факторів та оцінка ймовірності їх негативного впливу на здоров'я користувача.

Аналіз ризику дозволяє визначити, які небезпечні фактори можуть виникати під час роботи з комп'ютером, оцінити ймовірність їх появи та можливі наслідки. На основі отриманих результатів можна розробити заходи щодо зменшення або усунення цих ризиків.

Процес оцінки ризику включає кілька основних етапів:

- 1) ідентифікація небезпечних факторів;
- 2) визначення ймовірності їх виникнення;
- 3) оцінка можливих наслідків для здоров'я людини;
- 4) визначення рівня ризику;
- 5) розробка заходів щодо зниження ризику.

Оцінка ризику небезпечних факторів. Під час роботи за комп'ютером можуть виникати різні види небезпек, які впливають на стан здоров'я користувача. Для оцінки ризику використовують умовну шкалу, яка враховує ймовірність виникнення небезпечної ситуації та тяжкість можливих наслідків.

Основні небезпечні фактори та оцінка їх ризику наведено в таблиці 4.4.

Таблиця 4.4 – Основні небезпечні фактори

№	Небезпечний фактор	Ймовірність виникнення	Можливі наслідки	Рівень ризику
1	Тривала робота за монітором	висока	втома очей, погіршення зору	середній
2	Неправильна організація робочого місця	середня	біль у спині, порушення постави	середній
3	Недостатнє освітлення	середня	перевтома зорового апарату	середній
4	Психоемоційне навантаження	середня	перевтома, зниження працездатності	низький–середній

№	Небезпечний фактор	Ймовірність виникнення	Можливі наслідки	Рівень ризику
5	Ураження електричним струмом	низька	травмування	низький

Для визначення рівня ризику часто використовується метод матриці ризику, який базується на поєднанні двох показників ймовірність виникнення небезпеки та ступінь тяжкості наслідків.

Таблиця 4.5 – Матриця оцінки ризику

Ймовірність / Наслідки	Низькі	Середні	Високі
Низька	Низький ризик	Низький ризик	Середній ризик
Середня	Низький ризик	Середній ризик	Високий ризик
Висока	Середній ризик	Високий ризик	Дуже високий ризик

Застосування такого підходу дозволяє систематизувати небезпечні фактори та визначити найбільш критичні з них.

Аналіз результатів оцінки ризику. Проведений аналіз показує, що найбільший вплив на здоров'я користувача під час роботи з комп'ютером мають: тривала робота перед екраном монітора; неправильна організація робочого місця; недостатній рівень освітлення; значне психоемоційне навантаження.

Ризик ураження електричним струмом є відносно низьким, оскільки сучасне комп'ютерне обладнання має високий рівень електробезпеки та оснащене відповідними засобами захисту. Узагальнення основних ризиків під час роботи з комп'ютером наведено у таблиці 4.6.

Таблиця 4.6 – Загальна оцінка ризиків

Група факторів	Рівень ризику	Основні причини
Зорове навантаження	середній	тривала робота за монітором

Група факторів	Рівень ризику	Основні причини
Ергономічні фактори	середній	неправильна організація робочого місця
Психофізіологічні фактори	низький–середній	інтелектуальне навантаження
Електричні фактори	низький	використання електрообладнання

Проведений аналіз показує, що основні ризики під час роботи над розробкою системи голосового керування комп'ютером пов'язані з тривалою роботою за персональним комп'ютером та організацією робочого місця. Найбільш значними є ризики, пов'язані із зоровим навантаженням та ергономічними факторами.

Для зменшення негативного впливу цих факторів необхідно дотримуватися правил організації робочого місця, забезпечити достатній рівень освітлення, а також робити регулярні перерви під час роботи з комп'ютером. Це дозволить знизити рівень ризику та забезпечити безпечні умови праці.

Під час роботи з комп'ютерною технікою та електричним обладнанням важливим аспектом є забезпечення безпеки у разі виникнення надзвичайних ситуацій. До надзвичайних ситуацій можуть належати пожежі, аварійні відключення електроенергії, пошкодження електрообладнання, а також інші події, які можуть становити загрозу для здоров'я людей та збереження матеріальних цінностей.

Під час розробки та використання системи голосового керування комп'ютером робота виконується у приміщенні з використанням персонального комп'ютера, периферійних пристроїв та електромережі. Тому основні надзвичайні ситуації, які можуть виникнути, пов'язані з порушенням електробезпеки або виникненням пожежі.

До основних надзвичайних ситуацій, які можуть виникнути під час роботи з комп'ютерною технікою, належать: виникнення пожежі внаслідок

короткого замикання; пошкодження електричного обладнання; аварійне відключення електроенергії; перегрів електронного обладнання; пошкодження електричних кабелів. Основні можливі небезпечні ситуації наведено у таблиці 4.7.

Таблиця 4.7 – Основні надзвичайні ситуації

№	Надзвичайна ситуація	Причини виникнення	Можливі наслідки
1	Пожежа	коротке замикання, несправність обладнання	пошкодження майна, загроза життю
2	Відключення електроенергії	аварія в електромережі	припинення роботи обладнання
3	Перегрів комп'ютера	недостатня вентиляція	вихід обладнання з ладу
4	Пошкодження кабелів	механічне пошкодження	ризик ураження електричним струмом

Пожежна безпека є важливою складовою безпеки під час роботи з комп'ютерною технікою. Основними причинами пожеж у приміщеннях з електронним обладнанням можуть бути коротке замикання, перевантаження електромережі або несправність електроприладів.

Для запобігання пожежам необхідно дотримуватися таких правил:

- 1) використовувати справне електрообладнання;
- 2) не перевантажувати електромережу;
- 3) регулярно перевіряти стан електричних проводів;
- 4) забезпечувати належну вентиляцію комп'ютерного обладнання;
- 5) не залишати увімкнене обладнання без нагляду.

У приміщенні повинні бути передбачені первинні засоби пожежогасіння, наприклад вогнегасники.

У разі виникнення пожежі необхідно діяти відповідно до встановлених правил безпеки:

- 1) негайно повідомити відповідні служби;
- 2) відключити електроживлення обладнання;
- 3) використати наявні засоби пожежогасіння;
- 4) організовано покинути небезпечне приміщення.

Схема дій у разі виникнення пожежі може бути представлена так:

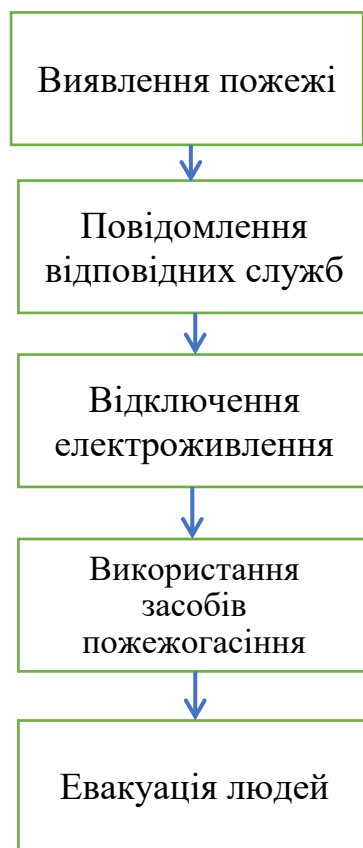


Рисунок 4.1 – Алгоритм дій у разі виникнення пожежі

У разі аварійного відключення електроенергії можливе раптове припинення роботи комп'ютера, що може призвести до втрати даних або пошкодження обладнання.

Для запобігання таким ситуаціям рекомендується:

- 1) використовувати джерела безперебійного живлення;
- 2) регулярно зберігати результати роботи;

3) правильно завершувати роботу комп'ютера після відновлення електроживлення.

Евакуація людей у надзвичайних ситуаціях. У разі виникнення небезпечної ситуації важливо забезпечити своєчасну евакуацію людей з приміщення. Для цього необхідно дотримуватися таких правил:

- 1) зберігати спокій;
- 2) швидко залишити приміщення через евакуаційні виходи;
- 3) не користуватися ліфтами;
- 4) допомагати іншим людям у разі потреби.

Забезпечення безпеки у надзвичайних ситуаціях є важливою складовою організації безпечних умов праці під час роботи з комп'ютерною технікою. Основними небезпечними ситуаціями можуть бути пожежі, аварійні відключення електроенергії та пошкодження електричного обладнання. Дотримання правил пожежної та електричної безпеки, а також правильна організація робочого місця дозволяють зменшити ризик виникнення небезпечних ситуацій та забезпечити безпечні умови роботи.

Висновок до розділу 4.

У розділі 4 було розглянуто основні питання охорони праці та безпеки під час виконання робіт, пов'язаних із розробкою програмного забезпечення системи голосового керування комп'ютером. Проведено аналіз умов праці користувача, який працює за персональним комп'ютером, а також визначено основні потенційно небезпечні та шкідливі фактори, що можуть впливати на здоров'я людини.

Було ідентифіковано можливі небезпеки, які виникають під час роботи з комп'ютерною технікою. До них належать зорове навантаження, неправильна організація робочого місця, психофізіологічне перевантаження та ризик ураження електричним струмом під час роботи з електрообладнанням.

Під час аналізу та оцінки ризиків визначено рівень впливу цих факторів на користувача. Було встановлено, що найбільший вплив мають ергономічні

та зорові фактори, які пов'язані з тривалою роботою за комп'ютером. Для зменшення їх негативного впливу необхідно правильно організувати робоче місце, дотримуватися режиму праці та відпочинку, а також забезпечити достатній рівень освітлення.

Було розглянуто питання безпеки у надзвичайних ситуаціях. Проаналізовано можливі аварійні ситуації, такі як пожежі, пошкодження електричного обладнання та аварійні відключення електроживлення. Описано основні правила поведінки у таких ситуаціях, а також заходи щодо запобігання виникненню небезпечних подій.

Дотримання вимог охорони праці, правил електробезпеки та пожежної безпеки, а також правильна організація робочого місця забезпечують безпечні умови праці під час розробки та використання програмних систем. Реалізація запропонованих заходів дозволяє мінімізувати вплив небезпечних факторів та підвищити рівень безпеки під час роботи з комп'ютерною технікою.

ЗАГАЛЬНІ ВИСНОВКИ

У дипломній роботі було досліджено та реалізовано інтелектуальну систему голосового керування комп'ютером, що дозволяє користувачеві взаємодіяти з програмним забезпеченням за допомогою голосових команд. Актуальність даної теми зумовлена швидким розвитком технологій розпізнавання мовлення, а також зростаючою потребою у зручних та інтуїтивних способах взаємодії людини з комп'ютером.

У першому розділі було розглянуто теоретичні основи систем голосового керування, зокрема принципи та методи розпізнавання мовлення, а також проведено аналіз існуючих систем голосового керування комп'ютером. Окрім цього, було досліджено сучасні технології та інструменти, що використовуються для створення систем розпізнавання мовлення. Проведений аналіз дозволив визначити основні підходи до побудови подібних систем та обрати оптимальні технологічні рішення для подальшої реалізації.

У другому розділі було здійснено проектування інтелектуальної системи голосового керування. Зокрема, визначено функціональні та нефункціональні вимоги до системи, розроблено її загальну архітектуру, а також описано алгоритми розпізнавання та обробки голосових команд. Це дозволило сформуванню логічну структуру системи та забезпечити ефективну обробку голосових даних і перетворення їх у відповідні дії на комп'ютері.

У третьому розділі було реалізовано програмне забезпечення системи голосового керування комп'ютером. Проведено інтеграцію системи з операційною системою та прикладними програмами, що забезпечило можливість виконання різних команд за допомогою голосу. Також було розглянуто основні сценарії використання системи та продемонстровано приклади її роботи, що підтвердило працездатність і практичну доцільність розробленого рішення.

У четвертому розділі розглянуто питання охорони праці та безпеки в надзвичайних ситуаціях. Було проведено ідентифікацію потенційних небезпек,

виконано аналіз та оцінку ризиків, а також розглянуто заходи щодо забезпечення безпеки праці під час роботи з комп'ютерною технікою та програмним забезпеченням.

У процесі виконання дипломної роботи було досягнуто поставленої мети, а саме, розроблено та реалізовано систему голосового керування комп'ютером. Отримані результати підтверджують можливість ефективного використання технологій розпізнавання мовлення для спрощення взаємодії користувача з комп'ютерною системою. Розроблена система може бути використана для підвищення зручності роботи користувачів, а також має потенціал для подальшого вдосконалення та розширення функціональних можливостей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://www.gartner.com/en/insights>
2. Голосові помічники: що це і навіщо вони HR. URL: <https://hurma.work/blog/voice-assistants-shho-cze-i-navishho-voni-hr-ok-google-alexa-siri-cortana/>
3. Jurafsky D., Martin J. H. Speech and Language Processing : електронний ресурс. – Stanford University, 2023. – Режим доступу: <https://web.stanford.edu/~jurafsky/slp3/>
4. Goodfellow I., Bengio Y., Courville A. Deep Learning : електронний ресурс. – MIT Press, 2021. – Режим доступу: <https://www.deeplearningbook.org/>
5. Aggarwal C. C. Neural Networks and Deep Learning : електронний ресурс. – Springer, 2021. – Режим доступу: <https://link.springer.com/book/10.1007/978-3-030-43823-4>
6. Wang Y., Leow C. S., Kobayashi A., Utsuro T., Nishizaki H. ExKaldi-RT: A Real-Time Automatic Speech Recognition Extension Toolkit of Kaldi // arXiv. – 2021. – Режим доступу: <https://arxiv.org/abs/2104.01786>
7. Mehrish A., Majumder N., Bhardwaj R., Mihalcea R., Poria S. A Review of Deep Learning Techniques for Speech Processing // arXiv. – 2023. – Режим доступу: <https://arxiv.org/abs/2302.06539>
8. Radford A. та ін. Robust Speech Recognition via Large-Scale Weak Supervision // arXiv. – 2023. – Режим доступу: <https://arxiv.org/abs/2212.04356>
9. Google Developers. Speech-to-Text Documentation : електронний ресурс. – 2024. – Режим доступу: <https://cloud.google.com/speech-to-text/docs>
10. Mozilla Foundation. DeepSpeech: An Open Source Speech-to-Text Engine: електронний ресурс. – 2021. – Режим доступу: <https://github.com/mozilla/DeepSpeech>
11. Ipha Serpei. Vosk Speech Recognition Toolkit: електронний ресурс. – 2024. – Режим доступу: <https://alphacephei.com/vosk/>

12. Python Software Foundation. SpeechRecognition Library for Python : електронний ресурс. – 2024. – Режим доступу: <https://pypi.org/project/SpeechRecognition/>
13. TensorFlow. Simple Audio Recognition Tutorial : електронний ресурс. – 2023. – Режим доступу: https://www.tensorflow.org/tutorials/audio/simple_audio
14. OpenAI. Whisper: Automatic Speech Recognition System: електронний ресурс. – 2024. – Режим доступу: <https://github.com/openai/whisper>
15. Microsoft. Azure Speech Service Documentation: електронний ресурс. – 2024. – Режим доступу: <https://learn.microsoft.com/en-us/azure/ai-services/speech-service/>
16. IBM. Speech to Text API Documentation: електронний ресурс. – 2023. – Режим доступу: <https://cloud.ibm.com/apidocs/speech-to-text>
17. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. – Київ: ДП «УкрНДНЦ», 2016.
18. Закон України Про охорону праці (Відомості Верховної Ради України (ВВР), 1992, № 49, ст.668). <https://zakon.rada.gov.ua>
19. Методичні вказівки до виконання розділу «Охорона праці та безпека в надзвичайних ситуаціях». Харків : ХНУМГ ім. О. М. Бекетова, 2021.
20. ДСТУ EN 60204-1:2019 Безпечність машин. Електрообладнання машин. Частина 1. Загальні вимоги (EN 60204-1:2018, IDT; IEC 60204-1:2016, MOD)
21. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.
22. ДБН В.2.5-28:2018 Природне і штучне освітлення
23. ДСТУ 4297:2004 Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги
24. ДСТУ EN 54-1:2022 Системи виявлення пожежі та пожежної сигналізації - Частина 1: Вступ (EN 54–1:2021, IDT)

25. ДСТУ EN ISO 7010:2019 Графічні символи. Кольори та знаки безпеки. Зареєстровані знаки безпеки (EN ISO 7010:2012; A1:2014; A2:2014; A3:2014; A4:2014; A5:2015; A6:2016; A7:2017, IDT; ISO 7010:2011; Amd 1:2012; Amd 2:2012; Amd 3:2012; Amd 4:2013; Amd 5:2014; Amd 6:2014; Amd 7:2016, IDT)

26. ДБН В.2.2-5:2023 Захисні споруди цивільного захисту.

ДОДАТОК А

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО ГОСПОДАРСТВА
імені О. М. БЕКЕТОВА**

**Навчально-науковий інститут енергетичної, інформаційної та транспортної інфраструктури
Кафедра автоматизації та комп'ютерно-інтегрованих технологій**

Розробка інтелектуальної системи голосового керування комп'ютером із застосуванням алгоритмів розпізнавання мовлення

здобувач вищої освіти групи СІНЖ 2023-1-у
Спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Марков Владислав Валерійович

Керівник Білецький Ігор Васильович, д. е. н., проф., проф. каф. АКІТ

АКТУАЛЬНІСТЬ

Голосове управління комп'ютером на основі алгоритмів розпізнавання мовлення дозволяє підвищити швидкість і комфорт користування системами, забезпечити інноваційний рівень автоматизації та створити підґрунтя для впровадження інтелектуальних асистентів і роботизованих систем. Застосування алгоритмів штучного інтелекту у поєднанні зі словниками команд дозволяє забезпечити точне та ефективне виконання завдань користувачем через голосові команди, що робить тему дослідження актуальною для автоматизації, комп'ютерно-інтегрованих технологій та робототехніки.

МЕТА

Метою роботи є створення та дослідження інтелектуальної системи керування комп'ютером за допомогою голосових команд, що базується на використанні алгоритмів розпізнавання мовлення та словника команд. Розроблена система повинна забезпечувати точне виконання команд користувача та підвищувати зручність взаємодії людини з комп'ютером.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- Провести аналіз сучасних методів голосового керування та алгоритмів розпізнавання мовлення.
- Розробити архітектуру інтелектуальної системи голосового керування комп'ютером.
- Сформувати словник команд для системи та визначити алгоритми його інтеграції.
- Реалізувати програмну систему голосового керування комп'ютером із використанням алгоритмів розпізнавання мовлення.

3

Об'єкт, предмет і методи дослідження

Об'єктом дослідження є процес взаємодії користувача з комп'ютерною системою за допомогою голосових команд із використанням технологій розпізнавання мовлення.

Предметом дослідження є методи та алгоритми оброблення голосових команд, а також способи їх інтеграції в систему інтелектуального керування комп'ютером.

У процесі виконання роботи використано такі методи дослідження, як аналіз літературних та наукових джерел з теми голосового управління та розпізнавання мовлення; системний підхід до побудови архітектури програмної системи; методи програмної реалізації алгоритмів розпізнавання мовлення.

4

Основні етапи розпізнавання мовлення

Основною метою технологій розпізнавання мовлення є автоматичне перетворення усного мовлення у текст або набір команд, які можуть бути використані для керування програмним забезпеченням чи технічними пристроями. Такі системи широко застосовуються у голосових помічниках, мобільних додатках, системах автоматизації та інтелектуальних інтерфейсах керування.

Етап	Опис
Захоплення аудіосигналу	Отримання голосового сигналу за допомогою мікрофона
Оцифрування сигналу	Перетворення аналогового звуку в цифровий формат
Попередня обробка	Фільтрація шумів, нормалізація гучності, поділ сигналу на фрагменти
Виділення ознак	Аналіз спектральних характеристик сигналу
Розпізнавання мовлення	Використання алгоритмів для визначення слів або команд
Інтерпретація	Перетворення розпізаного тексту у відповідні дії системи

5

Основні методи виділення ознак мовного сигналу

Метод	Опис	Особливості
MFCC (Mel-Frequency Cepstral Coefficients)	аналіз частотних характеристик сигналу	найпоширеніший метод у системах розпізнавання
LPC (Linear Predictive Coding)	прогнозування параметрів сигналу	використовується у системах стиснення мовлення
Spectrogram analysis	аналіз спектра сигналу	дозволяє візуально досліджувати мовний сигнал

6

Основні алгоритми розпізнавання мовлення

Алгоритм	Принцип роботи	Переваги	Недоліки
Приховані марковські моделі (HMM)	статистичне моделювання послідовностей сигналу	ефективні для обробки часових сигналів	потребують великої кількості даних
Нейронні мережі (DNN)	аналіз складних закономірностей у даних	висока точність	потребують значних обчислювальних ресурсів
Рекурентні нейронні мережі (RNN, LSTM)	врахування послідовності сигналів у часі	добре працюють з мовними даними	складність навчання
Трансформерні моделі	аналіз великих обсягів мовних даних	дуже висока точність	потребують потужного обладнання

7

Функції мовних моделей у системах розпізнавання мовлення

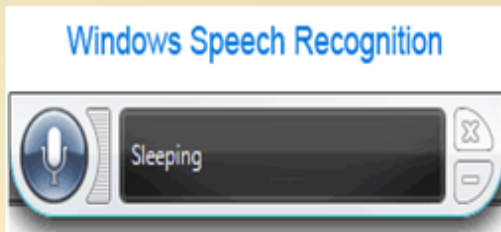
Функція	Опис
Аналіз контексту	визначення найбільш імовірної послідовності слів
Підвищення точності	зменшення кількості помилок під час розпізнавання
Обробка природної мови	адаптація системи до особливостей мови

8

Найбільш поширені системи голосового керування

Windows Speech Recognition

Перевагою системи є її інтеграція з операційною системою, що забезпечує простоту налаштування та використання. Вона не потребує встановлення додаткового програмного забезпечення. Її недоліками є обмежена точність розпізнавання в деяких умовах, залежність від якості мікрофона та обмежена підтримка різних мов.



Dragon NaturallySpeaking

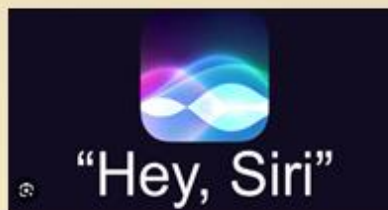
Перевагою системи є висока точність розпізнавання мовлення, широкий набір функціональних можливостей. Система добре підходить для професійного використання, наприклад у сфері медицини, юриспруденції або технічної документації. Її недоліками є висока вартість ПО та необхідність попереднього навчання системи для досягнення максимальної точності.



Найбільш поширені системи голосового керування

Siri (Apple)

Сучасна версія здатна надсилати голосові повідомлення, здійснювати переклад текстів різними іноземними мовами, виконувати широкий спектр інших дій, пов'язаних із керуванням пристроєм та обробкою інформації.



Google Assistant (Google)

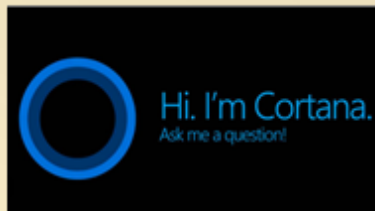
Орієнтована на ефективний пошук інформації в Інтернеті, тісну інтеграцію з хмарними сервісами компанії Google, забезпечує повну синхронізацію з операційною системою Android, що дозволяє користувачам керувати пристроєм, отримувати інформацію та виконувати різноманітні завдання за допомогою голосових команд.



Найбільш поширені системи голосового керування

Cortana (Microsoft)

Один із прикладів інтеграції технологій розпізнавання мовлення у програмні продукти компанії Microsoft, спрямовані на підвищення зручності роботи користувачів з цифровими пристроями.



Alexa (Amazon)

Підтримує голосове спілкування з користувачем, може відтворювати музику, подкасти та аудіокниги, створювати списки справ, встановлювати нагадування та будильники, взаємодіяти з різними пристроями «розумного дому», що дозволяє керувати освітленням, побутовою технікою та іншими підключеними пристроями за допомогою голосових команд.



11

Аналіз існуючих рішень

Голосові помічники Google Assistant, Cortana, Apple Siri та Amazon Alexa демонструють ефективність використання технологій розпізнавання мовлення у сфері керування комп'ютерними системами. Основною перевагою таких систем є висока точність розпізнавання та можливість обробки природної мови. Вони можуть постійно вдосконалюватися завдяки використанню хмарних сервісів і великих масивів даних. Використання таких систем має певні недоліки. Зокрема, для їхньої роботи зазвичай необхідне постійне підключення до Інтернету. Крім того, існують питання конфіденційності та безпеки даних, оскільки голосові запити можуть оброблятися на віддалених серверах.

При розробці власної системи голосового керування важливо враховувати особливості існуючих рішень та використовувати їхні переваги для створення ефективного та зручного програмного продукту.

12

Основні бібліотеки для розпізнавання мовлення

Бібліотека	Короткий опис	Переваги	Недоліки
SpeechRecognition	популярна бібліотека для Python, яка підтримує різні сервіси розпізнавання мовлення	проста у використанні, підтримує багато сервісів	залежність від сторонніх API
Vosk	бібліотека для офлайн розпізнавання мовлення	працює без інтернету, підтримує багато мов	потребує більше ресурсів
DeepSpeech	система розпізнавання мовлення на основі нейронних мереж	висока точність, сучасні алгоритми	складніше налаштування
CMU-Sphinx	одна з перших систем розпізнавання мовлення з відкритим кодом	працює офлайн, гнучке налаштування	нижча точність порівняно з сучасними моделями

13

Фреймворки для роботи з алгоритмами розпізнавання мовлення

Фреймворк	Призначення	Особливості
TensorFlow	створення та навчання нейронних мереж	використовується для складних моделей штучного інтелекту
PyTorch	розробка та навчання моделей глибокого навчання	зручний для досліджень та експериментів
Keras	високорівнева бібліотека для роботи з нейронними мережами	простий синтаксис та швидка розробка

14

Основні мови програмування для створення систем голосового керування

Мова програмування	Сфера використання	Переваги
Python	розробка систем штучного інтелекту та обробки мовлення	простота синтаксису, велика кількість бібліотек
JavaScript	створення веб-додатків з голосовим керуванням	підтримка у браузерях
C++	розробка високопродуктивних систем	висока швидкість роботи
Java	розробка мобільних та серверних систем	кросплатформеність

15

Хмарні сервіси розпізнавання мовлення

Сервіс	Компанія	Основні можливості
Google Speech-to-Text	Google	висока точність розпізнавання, підтримка багатьох мов
Microsoft Azure Speech	Microsoft	інтеграція з сервісами Microsoft
Amazon Transcribe	Amazon	автоматичне розпізнавання та аналіз мовлення

16

Основні вимоги до програмного забезпечення

Функціональні вимоги

№	Функціональна вимога	Опис
1	Запис голосової команди	система повинна отримувати аудіосигнал з мікрофона
2	Розпізнавання мовлення	перетворення голосу користувача у текст
3	Аналіз команди	визначення змісту голосової команди
4	Виконання команди	запуск програм або виконання системних дій
5	Зворотний зв'язок	повідомлення користувача про виконання команди

Нефункціональні вимоги

№	Нефункціональна вимога	Опис
1	Швидкість	система повинна швидко обробляти голосові команди
2	Точність	розпізнавання мовлення повинно бути достатньо точним
3	Надійність	система повинна стабільно працювати без збоїв
4	Зручність використання	користувач повинен легко взаємодіяти з системою
5	Масштабованість	можливість додавання нових голосових команд

17

Основні сценарії використання системи

№	Сценарій використання	Опис
1	Запуск програми	користувач вимовляє команду, система відкриває програму
2	Відкриття системних утиліт	користувач запускає калькулятор або блокнот голосом
3	Виконання системної команди	система виконує команду керування комп'ютером
4	Введення тексту	система перетворює мовлення у текст

18

Основні модулі системи голосового керування

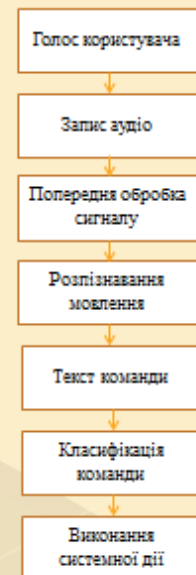
Модуль	Призначення
Модуль захоплення аудіосигналу	отримує голосовий сигнал користувача через мікрофон
Модуль попередньої обробки	очищає сигнал від шумів та готує його до подальшого аналізу
Модуль розпізнавання мовлення	перетворює голосовий сигнал у текст
Модуль аналізу команд	визначає зміст команди та необхідну дію
Модуль виконання команд	запускає програми або виконує системні дії
Модуль взаємодії з операційною системою	забезпечує зворотний зв'язок із користувачем

19

Алгоритм роботи системи голосового керування

№ етапу	Опис
1	Користувач вимовляє голосову команду
2	Система записує аудіосигнал через мікрофон
3	Аудіосигнал передається до модуля розпізнавання мовлення
4	Голосова команда перетворюється у текст
5	Текстова команда аналізується системою
6	Визначається відповідна системна дія
7	Система виконує команду у середовищі ОС

Загальна схема роботи алгоритмів системи



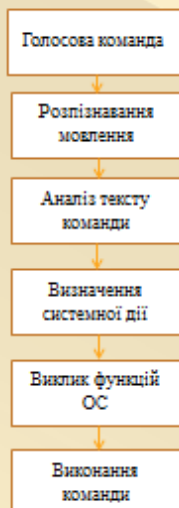
20

Функції Visual Studio Code

Функція	Опис
Підсвічування синтаксису	автоматичне виділення елементів коду різними кольорами
Автодоповнення коду	підказки під час написання програмного коду
Налагодження програм	можливість знаходження та виправлення помилок
Робота з проєктами	зручне керування файлами та папками
Підтримка розширень	можливість встановлення додаткових модулів

21

Загальний механізм взаємодії системи з операційною системою.



Основні можливості інтеграції системи

Функція системи	Опис
Запуск програм	відкриття прикладних програм операційної системи
Керування системою	виконання системних команд
Робота з файлами	відкриття та обробка файлів
Робота з веб-ресурсами	відкриття веб-сайтів у браузері
Керування програмами	взаємодія з запущеними додатками

22

ЗАГАЛЬНІ ВИСНОВКИ

У дипломній роботі досліджено та реалізовано інтелектуальну систему голосового керування комп'ютером, що дозволяє користувачеві взаємодіяти з програмним забезпеченням за допомогою голосових команд.

У першому розділі розглянуто теоретичні основи систем голосового керування, принципи та методи розпізнавання мовлення, проведено аналіз існуючих систем голосового керування комп'ютером, досліджено сучасні технології та інструменти, що використовуються для створення систем розпізнавання мовлення. Проведений аналіз дозволив визначити основні підходи до побудови подібних систем та обрати оптимальні технологічні рішення для подальшої реалізації.

У другому розділі здійснено проектування інтелектуальної системи голосового керування, визначено функціональні та нефункціональні вимоги до системи, розроблено її загальну архітектуру, описано алгоритми розпізнавання та обробки голосових команд.

У третьому розділі реалізовано програмне забезпечення системи голосового керування комп'ютером. Проведено інтеграцію системи з операційною системою та прикладними програмами, що забезпечило можливість виконання різних команд за допомогою голосу, розглянуто основні сценарії використання системи та продемонстровано приклади її роботи.

У четвертому розділі розглянуто питання охорони праці та безпеки в надзвичайних ситуаціях

23

Дякую за увагу!

24