

## КОМП'ЮТЕРНІ НАУКИ

УДК 004.4

DOI <https://doi.org/10.32782/2521-6643-2023.2-66.2>

**Безверхий О. І.**, доктор фізико-математичних наук, професор,  
професор кафедри інформаційних систем і технологій  
Національного транспортного університету  
ORCID: 0000-0002-0834-6335

**Александренко Д. О.**, магістрант кафедри  
інформаційних систем і технологій  
Національного транспортного університету  
ORCID: 0009-0009-1722-2869

**Луц В. Є.**, аспірант кафедри інформаційних систем і технологій  
Національного транспортного університету  
ORCID: 0009-0001-2948-6935

### ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ З МОЖЛИВІСТЮ ГОЛОСОВОГО УПРАВЛІННЯ

*Робота присвячена створенню інформаційної системи розпізнавання голосових команд на основі штучних нейронних мереж. З розвитком комп'ютерних систем стає все більш очевидним, що використання систем розпізнавання мови набагато розшириться, якщо стане можливим використання людської мови при роботі безпосередньо з комп'ютером, і зокрема стане можливим управління машиною звичайним голосом в реальному часі, а також введення і виведення інформації у вигляді звичайної людської мови. Одним з перспективних способів організації людино-машинного взаємодія є передача комп'ютерній системі інструкцій користувача в форматі мовних команд. Голосовий інтерфейс є необхідним компонентом, коли мова йде про створення комфортних умов життя для людей з обмеженими можливостями. В роботі визначено підходи до виділення інформативних ознак, що описують мовний сигнал: метод лінійного передбачення та спектральний аналіз, розглянуто структуру нейронної мережі із одним зворотнім зв'язком і встановлено, що навчання нейронної мережі здійснюється шляхом послідовного пред'явлення навчальної вибірки, з одночасним налаштуванням ваг відповідно до певної процедури, поки помилка налаштування по всій множині не досягне прийнятного низького рівня. Цінність одержаних результатів полягає в вдосконаленні нового методу розпізнавання мовлення, який краще адаптується до мовлення користувача, що потребує мінімум ресурсів і створенні інформаційної системи з можливістю голосового управління за допомогою пристроїв на базі різних операційних систем. На основі такого підходу спроектовано інформаційний кросплатформенний додаток з голосовим інтерфейсом.*

*Ключові слова: голосове управління, фонема, голосовий апарат, штучні нейронні мережі, генетичний алгоритм, спектральні відліки, кепстральні коефіцієнти.*

**Bezverkhyi O. I., Aleksandrenko D. O., Luts V. Ye. Designing an information system with the possibility of voice control**

*The work is devoted to the creation of an information system for recognizing voice commands based on artificial neural networks. With the development of computer systems, it is becoming more and more obvious that the use of speech recognition systems will be greatly expanded if it becomes possible to use human language when working directly with the computer, and in particular, it becomes possible to control the machine with a normal voice in real time, as well as input and output information in the form of ordinary human language. One of the promising ways of organizing human-machine interaction is the transmission of user instructions to the computer system in the form of language commands. Voice interface is a necessary component when it comes to creating comfortable living conditions for people with disabilities. In the paper, the approaches to the selection of informative features describing the speech signal are defined: the method of linear prediction and spectral analysis, the structure of a neural network with one feedback is considered, and it is established that the learning of the neural network is carried out by successive presentation of the training sample, with simultaneous adjustment weights according to a specific procedure until the tuning error across the set reaches an acceptably low level.*

© О. І. Безверхий, Д. О. Александренко, В. Є. Луц, 2023

---

*The value of the obtained results lies in the improvement of a new method of speech recognition, which is better adapted to the user's speech, which requires a minimum of resources and the creation of an information system with the possibility of voice control using devices based on various operating systems. An informative cross-platform application with a voice interface was designed based on this approach.*

*Key words: voice control, phoneme, vocal apparatus, artificial neural networks, genetic algorithm, spectral readings, cepstral coefficients.*

**Постановка проблеми.** Системи голосового управління здатні полегшити та спростити взаємодію користувача не тільки з комп'ютерною системою, але і спілкування між людьми, тому голосове розпізнавання мови є на сьогодні актуальною задачею. Особливо дана ідея розвивається у напрямку концепції так званих розумних пристроїв. Більше того, іноді голосовий інтерфейс є необхідною компонентою, наприклад, коли йде мова про людей з обмеженими можливостями. Задачею таких систем є виділення та розпізнавання із потоку звукового сигналу набору мовних команд. Причому при введенні якоїсь команди система не повинна реагувати на інші частини мовного сигналу.

При створенні системи голосового розпізнавання мови розробник стикається з певними проблемами. По перше, відсутність математичної моделі семантики мовного сигналу, що виражається в тому, що для визначення семантики мовного сигналу можуть застосовуватися ймовірнісні та евристичні методи, що не дають точного результату та точність яких обернено пропорційна кількості смислових одиниць, на які вони розраховані. По друге, індивідуальні характеристики мовця: специфіка вимови, акценти, наголоси тощо. По третє, робота із спонтанною мовою та необхідність виділення наявності ключового слова. По четверте, відмінності в акустичній обстановці, шуми [1].

В даний час існує три важливі напрями розробок в розпізнаванні мови. Перше – набори різномірних параметрів, які об'єднують абсолютну спектральну інформацію з динамічною або спектральну інформацію, залежну від часу, стають загальними. Друге – схожі методи перетворення, часто використовувані при нормалізації і декореляції параметрів для зменшення обчислювальних витрат, стають популярними. Третє – завдання оцінки параметрів сигналу об'єднується з процесом розпізнавання мови.

**Аналіз останніх досліджень і публікацій.** Параметризація аналогового сигналу мови є першим кроком в процесі розпізнавання мови. В даний час значні частини розробок зосереджені на завданнях розпізнавання незалежно від диктора, вся увага зосереджена на розробці описів, які не залежать від зміни диктора. Переважні параметри, які є описами спектральних енергій звуку, а не деталями голосу певного диктора [2]. Мова являє собою звукові коливання повітря в діапазоні частот від 70-100 Гц до декількох кілогерц. Для того щоб у вихідному повітряному потоці виникли коливання з такими частотами, необхідна наявність джерела звуку на шляху повітряного потоку. Справа в тому, що крім тону основної частоти, в звуці інструментів і голосі завжди присутні так звані гармоніки. Гармоніки є звуки інших частот, відмінних від основної частоти. У загальному випадку будь-який звук можна представити у вигляді деякого нескінченного набору абсолютно чистих звуків різних частот. Сукупність частот таких чистих звуків називається спектром звуку. У результаті досліджень було встановлено, що в утворенні мови активно беруть участь чотири частоти, які виникають в резонансних порожнинах мовного тракту. Ці частоти називаються формантами. З точки зору людини, найменшою смисловою одиницею є слово. Слово ділиться на склади. Що ж стосується науки, що має справу з розпізнаванням і синтезом мови, то тут вводиться інше поняття одиниці звукового ладу – фонема [3].

Варто відмітити два моменти:

1. Фонема не є фізичною реалізацією звуку, а є поданням звуку в свідомості (абстракцією).
2. Фонема втілює ідею атомарності, засновану на суб'єктивному уявленні про мови.

Фонем не ототожнюються безпосередньо зі словами або складами. Вони грають роль неподільних частинок, атомів мови і являють собою послідовності звуків. З фонем складаються всі інші конструкції мови, такі як склади і слова.

Досить важливою частиною дослідження для виконання задачі автоматичного розпізнавання мови є вивчення слухового аналізатора, оскільки неадекватна обробка сигналу може призвести до втрати частини корисних ознак та до занадто детального представлення іншої частини.

Комп'ютерні системи розпізнавання мови поступово знаходять застосування не тільки в науковій сфері, але й у побутовій. Прикладом тому можуть служити офісні пакети й інше ПЗ з убудованим розпізнаванням мови для голосового введення текстової інформації. Однією з проблем упровадження голосових технологій у портативній техніці був низький обчислювальний ресурс мікропроцесорів, і недостатній обсяг оперативної пам'яті. Крім цього алгоритми з достатньою надійністю розпізнавання мови в умовах складної шумової обстановки навколишнього середовища були занадто ресурсоємні для портативного застосування. Щоб навчити машину впізнавати мову, її потрібно заставити прослуховувати слова, сказані як однією людиною, так і різними людьми. Задача машини – прослухавши всі дані, взяти

середні значення особливостей вимови, повністю виключити індивідуальність, щоб потім, почувши слово, не зробити помилку.

Найбільші проблеми виникають в наступних умовах: довільний користувач; спонтанна мова, яка супроводжується мовним «сміттям», наявність акустичних завад і скривлень; наявність мовних завад.

**Мета статті.** Для спрощення процесу розпізнавання мови доцільно було б використовувати шаблони окремих звуків єдині для всіх дикторів. Тому для реалізації ефективних незалежних систем автоматизованого розпізнавання мови необхідно виділити інформативні ознаки звуків мови, розробити математичні методи їх опрацювання з метою створення єдиних для всіх дикторів шаблонів.

На сьогоднішній день розпізнавання мови зводиться до вирішення трьох типів задач:

1. Розпізнавання окремо вимовлених слів.
2. Розпізнавання зливої мови.
3. Ідентифікація по зразку мови.

Розробка технології розпізнавання мови вчені почали з освоєння методики виділення інформативних ознак, що описують мовної сигнал. Потім приступили до вирішення завдання класифікації мовних сигналів наборами інформативних ознак. Існують такі підходи до виділення інформативних ознак, що описують мовний сигнал:

- метод лінійного передбачення;
- спектральний аналіз.

Спектральний аналіз відрізняється від лінійного передбачення тим, що оцінки середнього значення усередненого шуму віднімаються з спектра, обчисленого по зашумленими даними. Найбільш часто використовуються два підходу до класифікації та розпізнавання:

- міра близькості параметрів (така функція називається метрикою);
- нейронні мережі.

Другий підхід не використовує допоміжних функцій, але моделює процес розпізнавання в біологічних системах. Цей підхід видається більш перспективним в даний час. У системах розпізнавання мови виділяються дві основні підсистеми:

- підсистема попередньої обробки мовних сигналів;
- підсистема класифікації мовних сигналів.

**Виклад основного матеріалу.** Нехай існує мовний сигнал як вхідні дані нейронної мережі. Після обробки звукових даних отримано масив сегментів сигналів. Кожен сегмент відповідає набору чисел, що характеризують амплітудні спектри сигналу. Для підготовки до обчислення для сигналу виходу нейронної мережі необхідно записати всі набори чисел в таблицю, рядок якої – це набір чисел кожного кадру. Кількість вхідних і вихідних нейронів відомо. Кожен з вхідних нейронів відповідає одному набору чисел. А на вихідному шарі тільки один нейрон, вихід якого відповідає бажаному значенню розпізнавання сигналу.

Нехай  $x_q$  –  $i$ -е вхідне значення  $q$ -го набору чисел;

$u_i$  – вихід  $j$ -го нейрона шару;

$w_{ij}$  – ваговий коефіцієнт зв'язку, що з'єднує  $i$ -й нейрон з  $j$ -м нейроном;

$w_j$  – ваговий коефіцієнт зворотного зв'язку  $j$ -го нейрону;

$\beta_j$  – зміщення  $j$ -го нейрону шару.

Для обчислення виходу нейронної мережі необхідно виконати наступні послідовні кроки.

Крок 1. Ініціювати всі контексти всіх нейронів прихованого шару  $x_j=0$ .

Крок 2. Подати перший набір чисел на вхід нейронної мережі  $x_1=(x_{11}, x_{12}, \dots, x_{1n})$ . Обчислити для нього виходи прихованого шару.

$$y_j = f(\sum w_{ij} l_i = I x_i + \beta_j + w_j x_j) \quad (1)$$

де  $f(x)$  – нелінійна активаційна функція  $y_j = \Pi + e^{-\alpha S_j}$ .

Крок 3. Якщо поточний набір чисел не є останнім, то перейти до кроку 5, інакше до кроку 4.

Крок 4. Записати виходи нейронів прихованого шару на контексти  $x_j = u_j$ , де  $j = 1, \dots, J$ . Перехід до кроку 2 для наступного набору чисел.

Крок 5. Обчислити вихід нейрону вихідного шару.

$$y_k = f(\sum w_{jk} l_i = I y_j + \beta_k) \quad (2)$$

Розглянемо задачу, яка полягає в розпізнаванні чисел від 0 до 9. Для розпізнавання одного числа потрібно побудувати власну нейронну мережу. І так повинно побудувати 10 нейронних мереж. Надиктована база з 250 слів (числа від 0 до 9) з різними варіаціями вимови. База випадковим чином поділялася на дві рівні частини – навчальну і тестувальну вибірки.

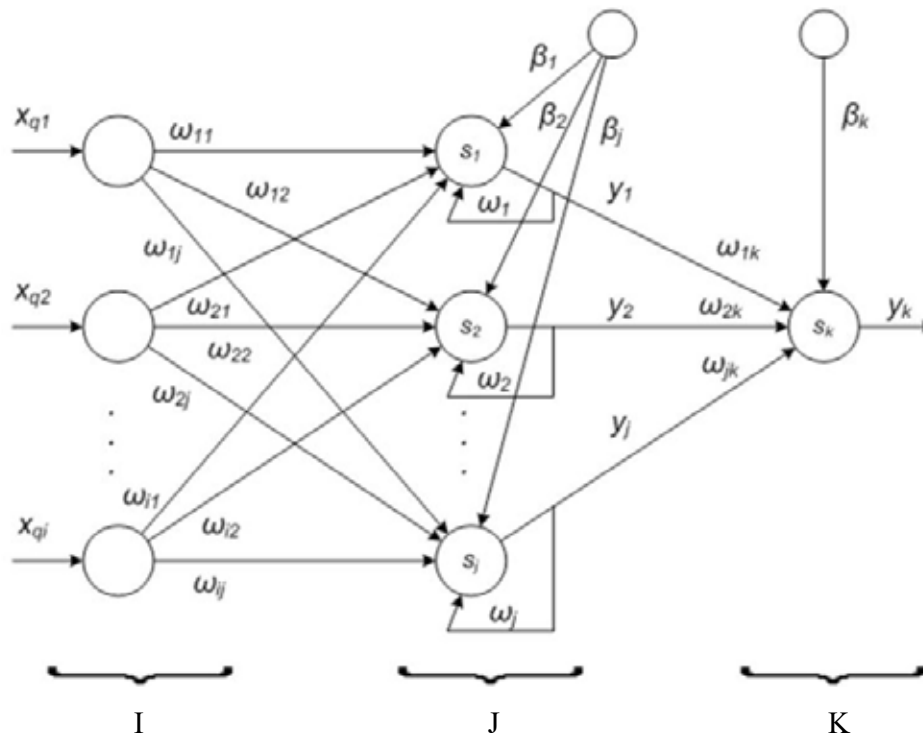


Рис. 1. Структура нейронної мережі із одним зворотнім зв'язком

Таблиця 1

**Опис набору ознак мовного сигналу**

Кадр	1-е значення	2-е значення	...	I-е значення
1-й кадр	$x_{11}$	$x_{12}$	...	$x_{1I}$
2-й кадр	$x_{21}$	$x_{22}$	...	$x_{2I}$
...	...	...	...	...
N-ий кадр	$x_{N1}$	$x_{N2}$	...	$x_{NI}$

I – кількість значень одного набору чисел, N – кількість наборів чисел (кадр сигналу після нарізки).

При навчанні нейронної мережі розпізнаванню одного числа, наприклад 5, бажаний вихід цієї нейронної мережі повинен бути одиницею для навчальної вибірки з числом 5, а інші – нулю. Навчання нейронної мережі здійснюється шляхом послідовного пред'явлення навчальної вибірки, з одночасним налаштуванням ваг відповідно до певної процедури, поки помилка налаштування по всій множині не досягне прийняттого низького рівня. Функція помилки в системі буде обчислюватися за наступною формулою:

$$E = 12N \sum (y_{ki} - d_i) N_i = 12 \quad (3)$$

де N – кількість навчальних вибірок, оброблених нейронною мережею прикладів;  $y_{ki}$  – реальний вихід нейронної мережі;  $d_i$  – бажаний (ідеальний) вихід нейронної мережі.

Для кожного слова з тестової вибірки реальні виходи обчислюються нейронними мережами розпізнавання різних чисел. Нейронна мережа, яка має максимальне вихідне значення, і є нейронною мережею розпізнавання даного слова. І слово, розпізнане нейронною мережею, є результатом розпізнавання [4].

**Проектування веб-додатку розпізнавання мови.** Проектування веб-додатку – це створення продуманої моделі ефективного веб-ресурсу на етапі підготовки до створення дизайну. Створення структури сайту, а саме – структури розділів і інтерфейсу, дозволяє оптимізувати процес розробки. Створений прототип дозволяє оцінити його відповідність завданням і при необхідності внести зміни в попередню модель. Таким чином, розробка дизайну і програмної частини відбуватиметься на основі прототипу, що дозволить уникнути несподіванок і виконати роботу максимально оперативно і ефективно, з мінімумом оптимізації

в процесі. Прототип створюється на основі технічного завдання, в якому відображені вимоги до сайту. Збір вимог – відповідальне завдання, від якої залежить доцільність і ефективність майбутнього проекту. Проектування інтерфейсу і структури сайту завжди починається з постановки завдань і розбору тематики [5].

Web Speech API дозволяє взаємодіяти з голосовими інтерфейсами у веб-додатках. Він складається з двох частин – SpeechSynthesis – перетворення тексту в мову та SpeechRecognition – асинхронне розпізнавання мови. Web Speech API дозволяє веб-програмам обробляти голосові дані. Доступ до мовлення здійснюється через інтерфейс SpeechRecognition, який забезпечує можливість розпізнавання контексту мовлення з аудіовходу (зазвичай за допомогою служби розпізнавання мови за умовчанням пристрою) і відповідати належним чином.

Як правило, використовується конструктор інтерфейсу для створення нового об'єкта SpeechRecognition, який має ряд оброблювачів подій для виявлення, коли мовлення здійснюється через мікрофон пристрою.

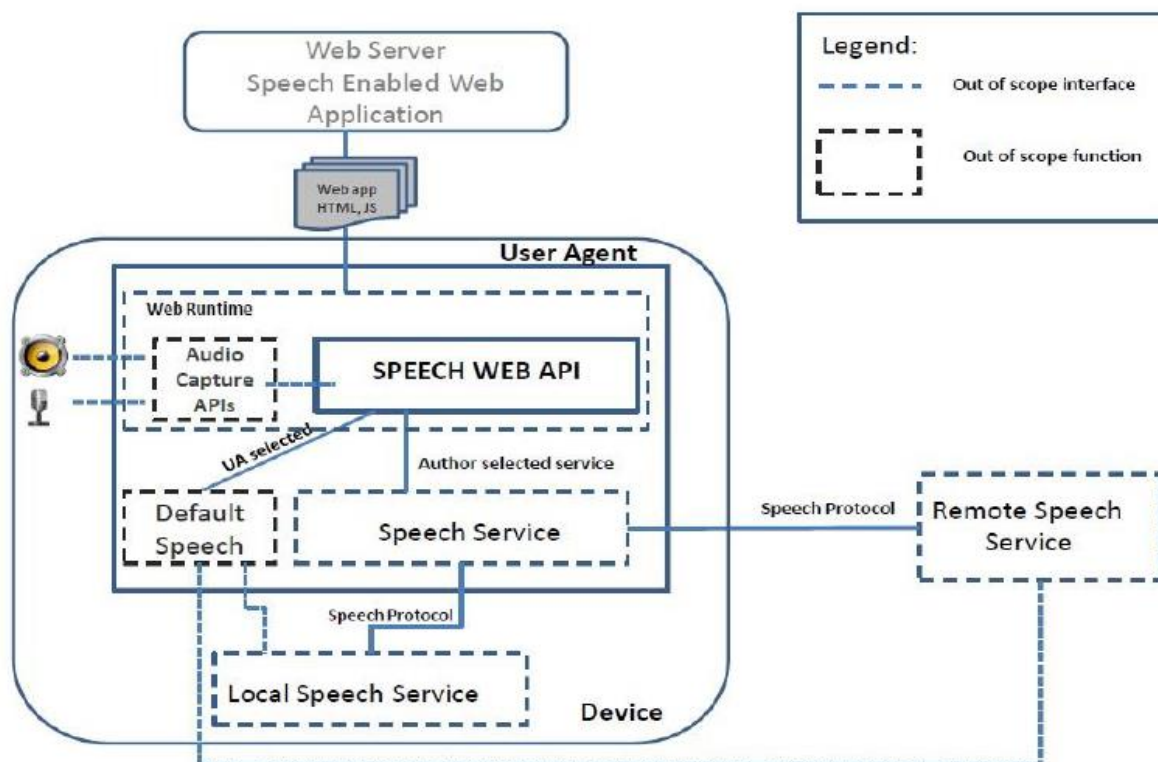


Рис. 2. Схема голосового розпізнавання Web Server

Інтерфейс SpeechGrammar представляє контейнер для певного набору граматики, який прилад повинен розпізнати. Граматика визначається за допомогою JSpeech Grammar Format (JSGF) [6].

Синтез мовлення здійснюється через інтерфейс SpeechSynthesis, компонент «текст-мовлення», який дозволяє програмам прочитати текстовий вміст (зазвичай за допомогою синтезатора рішень за замовчуванням пристрою). Різні типи голосів та текстів представлені об'єктами SpeechSynthesisVoice та SpeechSynthesisUtterance. Отримати голосове відображення можна, передаючи об'єкти методу SpeechSynthesis.speak() [7].

Голосове розпізнавання передбачає отримання мовлення через мікрофон пристрою, який потім перевіряється службою розпізнавання мов у списку граматики (в основному, словник, який окремо вказується в кожному додатку). Коли слово або фраза успішно розпізнаються, вони передаються як результат (або список результатів) у вигляді текстового рядку, з якими в результаті можуть бути ініційовані подальші дії.

Web Speech API має основний інтерфейс контролера – SpeechRecognition – плюс ряд тісно пов'язаних інтерфейсів для представлення граматики, результатів тощо. Як правило, стандартна система розпізнавання мовлення, доступна на пристрої, буде використовуватися для розпізнавання мови – більшість сучасні ОС мають систему розпізнавання мовлення для випуску голосових команд. Наприклад, Mac OS X, Siri на iOS, Cortana в Windows 10, Android Speech тощо [8].

Реалізувати голосове розпізнавання команд веб-додатком виконаємо за допомогою бібліотеки JavaScript Annyang.

Annyang – невелика бібліотека із відкритим вихідним кодом, важить 2 кілобайта та знаходиться у вільному доступі для використання у проектах відповідно до MIT ліцензії.

---

Розпізнавання мовлення відрізняється залежно від використовуваного протоколу:

https:// Один раз просить про дозвіл і запам'ятовує вибір.

http:// Просить багаторазового дозволу на кожну завантаження сторінки. Результати також повертаються значно повільніше в HTTP.

Фактична обробка мовлення та перетворення команд в текст виконується в хмарі, тому швидше з'єднання зумовлює оптимальнішу роботу додатку.

Коли розпізнавання мови відбувається в безперервному режимі, результати повертаються повільніше (браузер чекає, коли людина закінчить розмовляти, щоб побачити, чи є щось ще, що людина хоче додати).

Якщо вимкнути режим безперервного режиму, веб-переглядач повертає результат набагато швидше.

Щоб запустити annyang у неперервному режимі, необхідно вказати continuous: false в об'єкті параметрів, який приймає annyang.start(). Також варто ввімкнути autoRestart.

```
annyang.start({autoRestart: true, continuous: false});
```

Варто звернути увагу, що ці параметри вже є стандартними, якщо використовується HTTPS. Якщо використовується HTTP, за замовчуванням буде активовано безперервний режим (що призведе до повільного розпізнавання), щоб запобігти повторним повідомленням про безпеку [9].

Для створення голосового інтерфейсу будемо використовувати бібліотеку annyang.

Основні команди бібліотеки:

init(commands, [resetCommands=true]) – ініціалізація бібліотеки за допомогою додавання списку команд для розпізнавання.

start([options]) – команда, що вказує на початок голосового розпізнавання, тому ініціалізація команд має бути зроблена до початку розпізнавання.

Опції команди start:

autoRestart (boolean, default: true) – дозвіл перезапуску при виникненні проблем із розпізнаванням;

continuous (boolean) – дозвіл на ввімкнення постійного режиму розпізнавання;

paused (boolean, default: true) – запуск розпізнавання у режимі призупинення.

abort() – зупинити розпізнавання або вимкнути мікрофон.

pause() – призупинення розпізнавання команд (до виклику команди start() або resume()) без вимкнення движка SpeechRecognition в браузері чи мікрофону.

resume() – відновлює виконання команди зворотного виклику та розпізнавання.

debug([newState=true]) – виведення налагоджувальних повідомлень на консолі.

setLanguage(language) – встановлення мови розпізнавання.

addCommands(commands) – додавання команд, схожий на функцію init(), але не видаляє попередніх команд.

removeCommands([commandsToRemove]) – видалити існуючі команди, можна передавати одиночну фразу чи масив команд, щоб видалити всі попередні команди варто запустити функцію без параметрів.

addCallback(type, callback, [context]) – можливість отримання масиву із розпізнаних слів та близьких до них.

removeCallback(type, callback) – видалення колбеків із подій [9].

Створюємо скрипт для розпізнавання команд, наприклад для входу в систему.

```
if (annyang) {  
  var login = function(name){  
    var user = document.getElementById('user');  
    user.focus();  
    user.value = name;  
  };  
  var addPassword = function(pass){  
    var password = document.getElementById('pass');  
    password.focus();  
    password.value = pass;  
  };  
  var loginCommands = {  
    'username *name': login,  
    'password *pass': addPassword,  
    'sign in': welcome,  
  };  
  var welcome = function(){  
    signIN();  
  }  
  annyang.addCommands(loginCommands);  
}
```

---

```
annyang.start();
}
}
```

Перш за все, перевіряємо чи розпочалося розпізнавання і чи працює бібліотека, для цього виконуємо перевірку `if(annyang)`.

Далі створюємо функції до полів функції згідно з тим, що вимовив користувач. Створюємо перелік команд та ставимо їм у виконання функції відповідно до розпізнаної фрази. Наприклад, `*name` – означає параметр, що передається у функцію `login(name)` та буде встановлено у полі логін форми входу. Аналогічно виконується команда введення паролю. Якщо виконати команду `'sign in'`, вона запускає на виконання створену раніше функцію `signIN()`, яка перевіряє на наявність зареєстрованого користувача із введеними логіном та паролем.

Для переходу між сторінками застосовуємо `location.href`.

Створимо голосове розпізнавання для прокручування сторінки до початку, вверх та вниз. Для цього додаємо наступний скрипт:

```
var userCommands = {
  'scroll top': scrollTop,
  'scroll down': down,
  'scroll up': up,
};
var scrollTop = function() {
  console.log('scroll top');
  var top = Math.max(document.body.scrollTop, document.documentElement.scrollTop);
  if(top > 0) {
    window.scrollTo(0, -100);
    t = setTimeout('scrollTop()', 50);
  } else clearTimeout(t);
  return false;
}
var down = function() {
  $('html, body').animate({scrollTop: '+=300'}, 500);
}
var up = function() {
  $('html, body').animate({scrollTop: '-=300'}, 500);
}
```

Функції, що виконуються є звичайними у мові Javascript. Для додавання голосового інтерфейсу варто лише створити відповідну команду, якій передати або існуючу функцію, або передати активність певного елемента чи створити нову функціональну одиницю, яка вбудована в псевдоелементи CSS.

Голосові функції веб-додатку є аналогічними з функціями, що здійснюються за допомогою допоміжних пристроїв: клавіатури, миші.

**Висновки.** У результаті виконання роботи було:

1. Проаналізовано сучасні можливості голосового керування пристроями. Визначено підходи до виділення інформативних ознак, що описують мовний сигнал: метод лінійного передбачення та спектральний аналіз.

2. Досліджено моделі розпізнавання мови на основі штучних нейронних мереж. Розглянуто структуру нейронної мережі із одним зворотнім зв'язком. Встановлено, що навчання нейронної мережі здійснюється шляхом послідовного пред'явлення навчальної вибірки, з одночасним налаштуванням ваг відповідно до певної процедури, поки помилка налаштування по всій множині не досягне прийняттого низького рівня.

3. Спроектовано інформаційний кросплатформенний додаток з голосовим інтерфейсом. Всі можливі переходи та зміни станів системи включають в себе голосове управління.

#### Список використаних джерел:

1. Dong Yu, Li Deng. Automatic Speech Recognition: A Deep Learning Approach. – L.: Springer-Verlag London, 2015. 320 p.
2. Automatic Speech recognition: short introduction. URL: [https://www.esat.kuleuven.be/psi/spraak/demo/Recog/asr\\_intro.html](https://www.esat.kuleuven.be/psi/spraak/demo/Recog/asr_intro.html)
3. Автоматичне розпізнавання, розуміння та синтез мовленнєвих сигналів в Україні / Т.К. Вінцюк, М.М. Сажок, Р.А. Селюх, Д.Я. Федорин, О.А. Юхименко, В.В. Робейко. *Управляющие системы и машины*. 2018. № 6. С. 7–24.

- 
4. Глибовець М. М., Олецкий О.В. Штучний інтелект. Київ : «Кисво-Могилянська академія», 2002. 364 с.
  5. Home Assistant. URL: <https://home-assistant.io/>
  6. Introducing the Web Speech API. URL: <https://www.sitepoint.com/introducing-web-speech-api/>
  7. JavaScript: Web API читання тексту та розпізнавання голосу. URL: <https://archakov.im/post/javascript-web-api-recognition-and-speech-text.html>
  8. Understand the Smart Home Skill API. URL: <https://developer.amazon.com/docs/smarthome/understand-the-smart-home-skill-api.html#how-the-smart-home-skill-api-works>
  9. annyang! Tutorial. URL: <https://github.com/TalAter/annyang>

#### References:

1. Dong Yu, Li Deng. Automatic Speech Recognition: A Deep Learning Approach. – L.: Springer-Verlag London, 2015. 320 p.
2. Automatic speech recognition: short introduction. URL: [https://www.esat.kuleuven.be/psi/spraak/demo/Recog/asr\\_intro.html](https://www.esat.kuleuven.be/psi/spraak/demo/Recog/asr_intro.html)
- 3 Automatic recognition, understanding and synthesis of speech signals in Ukraine / T.K. Vintsyuk, M.M. Sazhok, R.A. Selyukh, D.Ya. Fedoryn, O.A. Yukhymenko, V.V. Robeyko. *Control systems and machines*. 2018. No. 6. P. 7–24.
4. Hlybovets M. M., Oletskyi O. V. Artificial Intelligence. Kyiv: "Kyiv-Mohyla Academy", 2002. 364 p.
5. Home Assistant. URL: <https://home-assistant.io/>
6. Introducing the Web Speech API. URL: <https://www.sitepoint.com/introducing-web-speech-api/>
7. JavaScript: Web API text reading and voice recognition. URL: <https://archakov.im/post/javascript-web-api-recognition-and-speech-text.html>
8. Understand the Smart Home Skill API. URL: <https://developer.amazon.com/docs/smarthome/understand-the-smart-home-skill-api.html#how-the-smart-home-skill-api-works>
9. annyang! Tutorial. URL: <https://github.com/TalAter/annyang>