НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ІМЕНІ ІГОРЯ СІКОРСЬКОГО
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ
КАФЕДРА АКУСТИЧНИХ ТА МУЛЬТИМЕДІЙНИХ ЕЛЕКТРОННИХ СИСТЕМ

«На правах рукопису»
УДК 621.004.522

«До захисту допущено»
Завідувач кафедри
С. А. Найда
«_07__» червня 2021 р.

Дипломна робота бакалавра

зі спеціальності (спеціалізації) 171 Електроніка

на тему: «Знаходження дефектів у масиві мікрофонів за допомогою детектору голосової активності»

Виконав: студент IV курсу, групи ДГ-71

Грідін Андрій Євгенович

(пі

Керівник доц. каф. АМЕС к.т.н. доц. Богданов О. В. __

(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____

(назва розділу) (посада, науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент доц. каф. КЕОА к.т.н. доц. Варфоломєєв А. Ю.

(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань
Студент

Київ – 2021 р.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Електроніки

Кафедра Акустичних та мультимедійних електронних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність (спеціалізація) 171 – Електроніка

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____С. А. Найда

«_07__» червня 2021 р.

ЗАВДАННЯ
на дипломну роботу студенту

Грідіну Андрію Євгеновичу

1. Тема роботи: Знаходження дефектів у масиві мікрофонів за допомогою детектору голосової активності

керівник роботи Богданов Олексій Вікторович, доц. к.т.н.

затверджені наказом по університету від «24» травня 2021р. № 1316-с

2. Строк подання студентом роботи «01» червня 2021 року

3. Вихідні дані до проекту (роботи): використовувати програму та функції для знаходження дефектних мікрофонів в інтелектуальній системі.

4. Зміст дипломної роботи (перелік завдань, які потрібно розробити):
1) Актуальність використання детектора голосової активності;
2) Історія виникнення і стандарти детекторів голосової активності;
3) Аналіз існуючих алгоритмів VAD та їх порівняння;
4) Принцип детектування;
6) Принципова робота детектора в інтелектуальних системах;
7) Приклад технологій з використанням інтелектуальних мереж

5. Перелік графічного (ілюстративного) матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) Презентація

6. Консультанти розділів проекту (роботи)*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 02 «лютого» 2021 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Огляд літератури	03.02 – 9.02	Виконано
2	Написання вступу	10.02 – 16.02	Виконано
3	Написання 1-го розділу	17.02 – 01.03	Виконано
4	Написання 2-го розділу	02.03 – 29.03	Виконано
5	Проведення дослідження та оформлення результатів	30.03 – 03.05	Виконано
6	Написання 3-го розділу	04.05 – 24.05	Виконано
7	Написання реферату та оформлення всієї роботи	25.05 – 01.06	Виконано

Студент  А. Є. Грідін

Керівник проекту (роботи)  О. В. Богданов

ЗМІСТ

РЕФЕРАТ

Грідін, А. Є. *Знаходження дефектів у масиві мікрофонів за допомогою детектору голосової активності* : Дипломна робота бакалавра : 171 Електроніка. – Київ 2021. – 50 с.

Ключові слова: мовні кодеки, точки бездротового доступу, інтелектуальні системи, MATLAB®, пристрої введення, автоматичне розпізнавання мови, аудіосистеми, масиви мікрофонів, мікрофони, розумні будинки.

В даній роботі було проведено аналіз наявної інформації на тему використання детектору голосової активності для знаходження дефектів в масиві мікрофонів. На основі цього було зроблено висновок, що технології з використанням голосових помічників доволі розповсюджена і продовжує поширюватися, так як це дозволяє користувачам зручно взаємодіяти з системами без використанням прямого вводу даних. Проте через кількість даних пристроїв з голосовим способом вводу з’являється проблема їх обслуговування і перевірки.

Для дослідження були вивчені існуючі кодери, які використовуються в мобільному зв’язку, їх властивості, методи обробки, принципи роботи і недоліки. Для відтворення роботи детектора використовувався програмний пакет MATLAB®.

Була представлена робота із програмним доповненням для системи розумного будинку Amazon™ Dot, як приклад використання програмного забезпечення для роботи із одним елементом масиву мікрофонів.

ABSTRACT

Hridin Andrii. *Detection of defects in an array of microphones using a voice activity detector* : Bachelor Thesis : 171 Electronics. – Kyiv 2021. – 50 p.

Keywords: speech codecs, wireless access points, intelligent systems, MATLAB®, input devices, automatic speech recognition, audio systems, microphone arrays, microphones, smart homes.

In this thesis, an analysis of available information on the use of a voice activity detector to find defects in the array of microphones was performed. Based on this, it was concluded that technology using voice assistants is quite common and continues to spread, as it allows users to easily interact with systems without the use of direct data input. However, due to the number of data devices with voice input, there is a problem with their maintenance and verification.

This study examined existing coders used in mobile communications, their properties, processing methods, principles of operation and shortcomings. MATLAB® software package was used to reproduce the detector.

Work with the software add-on for the Amazon ™ Dot smart home system was presented as an example of using software to work with one element of an array of microphones.

Вступ.
Актуальність використання детектора голосової активності

Розвиток автоматизованих систем контролю, робототехніки та кібернетики дав поштовх на створення інтелектуальних систем і мереж для приладів, які зв’язуються між собою. Ці прилади та системи обмінюються інформацією один з одним для отримання певного результату. Вони можуть представляти собою як смарт-динаміки Amazon Echo©, Apple HomePod©, Google Home©, Lenovo Smart Assistant©, та і групу автоматизованих роботів чи дронів Digit© для сервісного обслуговування або Anybotics© для автоматизованої перевірки рухомого складу поїздів. Їх основна частина обробки даних в даних пристроях направлена на виконання задач, поставлених користувачем.

Контроль багатьох із цих пристроїв виконується за допомогою голосових команд користувача, якому для виконання проставленої перед ним задач потрібна працездатність певних елементів інтелектуальної системи або кожного з них. Забезпечити справність всіх елементів вводу, а саме мікрофонів, в деяких випадках може зайняти певний час, що призвести до затримання користувача в необхідний момент. Причини несправності можуть бути як знаходження пристрою в середі надмірної вологи чи вібрації, так і високої чи низької температури або поломку кабелів, які з’єднують ці мікрофони.

Деякі з пристроїв використовують масив мікрофонів для отримання команди від користувача з напрямку надходження голосового сигналу, однак обробка даних займає певний час, як і обробка голосових команд, наданих користувачем, для виконання задач. Автоматизовані системи контролю використовують нейро-мережі, для відтворення отриманого сигналу в команду, а це, на сам перед, потребує прямого підключення до мережі інтернет, що в деяких випадках неможливо. Використання заздалегідь вбудованого коду в пристрій пришвидшить і спростити перевірку елементу вводу. За допомогою заздалегідь написаного коду, який за рахунок адаптування під необхідні потреби.

1. Історія виникнення і стандарти детекторів голосової активності

Voice activity detector (VAD) — детектор активності мови — технологія стиснення мовного сигналу, за рахунок кодування пауз. У телекомунікаційних системах зв'язку найбільш дорогим елементом не є станційні споруди: комутатори, підсилювальні пункти, системи енергозабезпечення тощо, а є лінійні споруди, що зв'язують елементи мережі. Системи телефонного зв'язку не є винятком. Тому ефективність системи зв'язку визначається в першу чергу ефективністю використання ліній зв'язку [1].

Для збільшення обсягів інформації, що передається застосовуються безліч різних методів, наприклад, частотне і тимчасове ущільнення сигналів. У системах голосового зв'язку, до яких відносяться і системи мобільного зв'язку застосовуються різні системи стиснення.

Мова як природне джерело інформації має надмірністю, тобто в ній міститься безліч даних, які не несуть смислове навантаження. У зв'язку з цим, було створено безліч різних алгоритмів, які усувають надмірність мови, намагаючись виділити тільки значущі параметри мови. Зазвичай одночасно застосовуються декілька технологій компресії мовних даних, і вони об'єднуються під загальним назву голосовий кодек або вокодер.

Найбільш поширеним способом стиснення мовних даних є видалення пауз між фразами, словами, окремими звуками. Як показали численні дослідження, в мові (монолозі) може міститися до 50% пауз, а в діалозі їх обсяг може досягати 70%. Якщо врахувати, що телефонне з'єднання — це, як раз, розмова двох осіб, то з'являється можливість стиснення в 2-3 рази без втрати якості. Саме на підставі цієї властивості і реалізований механізм детектора активності мови. [2]

Алгоритм VAD працює не сам по собі, а як одна з операцій в процесі кодування мовного сигналу перед його відправкою в телекомунікаційну систему. Зазвичай, наявність пауз визначається на основі аналізу оцифрованих пакетів мовних даних, які представляють собою відрізки сигналу.

Як саме визначити паузу, тобто підібрати критерій, який дозволив би з високою часткою ймовірності прогнозувати, що даний пакет містить паузу, а не мова — найскладніший аспект в алгоритмі VAD. Ціною невірно прийнятого рішення буде втрата частини мовних даних.

У найбільш простий реалізації наявність паузи в наборі цифрових відліків визначається на основі порівняння сумарної енергії пакета мовних даних з деяким граничним значенням, яке відокремлює паузу від пакета з голосом. В такому випадку необхідно підібрати поріг так, щоб не допустити занадто часто усунення помилкових пауз, що може привести до втрати корисних даних і погіршення характеристик якості обслуговування (Quality of Service), а з іншого боку запобігти численний пропуск пауз, що може послужити зниження ефективності алгоритму VAD [1].

Зазвичай, для визначення пауз, застосовується складний алгоритм, що враховує не тільки енергію пакета, а й енергію спектральних складових відрізка сигналу. Крім того, в розрахунок береться і швидкість зміни (наростання або убування) енергії даного відрізка з попередніми. Також у випадку зі складною шумовою ситуацією ефективність роботи VAD може бути забезпечена періодичною оцінкою параметрів фонового шуму.

На приймальній стороні, працює інша частина VAD, мета якої відновити вихідний сигнал. Суть відновлення полягає не просто в заповненні пауз відрізками з нульовою енергією. Як показали дослідження, людина асоціює тишу в динаміці свого телефону як пропажа зв'язку та створює дискомфорт. Тому паузи між голосовими відрізками заповнюють шумом.

Можливі два варіанти:

1. По-перше, шум може створюватися генератором білого шуму. Це найбільш ефективний спосіб, тому що в даному випадку від джерела передається тільки інформація про тривалість пауз.
2. В іншому випадку, пауза на передавальній стороні сильно стискається, але загальні параметри, що описують гучність, частоту і т.і.,

залишаються. На приймальній стороні генератор відтворює паузу на підставі цих додаткових даних. Цей варіант вимагає передачу додаткових обсягів інформації, тобто знижує загальну ефективність VAD, але з іншого боку дозволяє домогтися найбільшої природності голосу, що практично прибирає «сліди» роботи детектора активності мови.

На практиці, як правило, використовують другий варіант нехай більш витратний, але і більш комфортний.

Алгоритм VAD використовується практично у всіх телекомунікаційних системах, де передається мова в цифровому вигляді. Зокрема, він знайшов широке застосування в технології VoIP, ТМЗК, ISDN, безумовно, в сортових системах зв'язку починаючи з другого покоління. Збільшення ефективності роботи системи, доступне і використанням VAD дозволяє прогнозувати і подальше його застосування, а також продовження робіт з пошуку більш досконалого механізму детектування пауз в мові [1].

2. Аналіз існуючих алгоритмів VAD та їх порівняння

Щоб використовувати переваги стиснення мовних пауз, було запропоновано безліч алгоритмів VAD, деякі з них були відібрані організаціями зі стандартизації, включаючи: Міжнародний консультаційний комітет по телефонії і телеграфії (International Telecommunication Union — Telecommunication sector — ITUT), Європейський інститут по стандартизації в галузі телекомунікацій (European Telecommunications Standards Institute — ETSI), Асоціація телекомунікаційної промисловості США (Telecommunications Industry Association — TIA) і Альянс галузей електронної промисловості (Electronics Industries Alliance — EIA).

Міжнародний комітет ITU-T випустив кодери G.729 Annex B (G.729B) [3] і G.723.1 Annex A (G.723.1A) [4] в якості розширення для 8 кбіт/с G.729 [5] і 5.3 / 6.3 кбіт/с G.723.1 [6] кодерів мови для можливості виконання переривчастої передачі (Discontinuous transmission – DTX). Європейський інститут стандартизації ETSI рекомендував GSM-FR, -HR і -EFR методи детектування активності мови для європейських систем цифрового сотового зв'язку [7-9].

Потім ETSI представив ще два детектора активності мови: адаптивний багато швидкісний VAD варіант 1 (кодер AMR1) і варіант 2 (кодер AMR2) [10], з тим, щоб використовувати їх в мережах третього покоління мобільного зв'язку UMTS. Північноамериканські організації зі стандартизації TIA і EIA представили два алгоритму VAD: один для кодера IS-96 [11], а інший для кодерів IS-127 [12] і IS-733 [13] (алгоритми VAD для S-127 і IS-733, які мають однакову структуру).

У Таблиці 2.1 представлені стандартизовані алгоритми VAD, класифіковані за принципом того, що вони аналізують. Головним чином — це енергії піддіапазонів і спектральна форма сигналу [16]. Наприклад, алгоритми VAD, запропоновані TIA і EIA, використовують попереднє розбиття сигналу на невелику кількість піддіапазонів, в той час як алгоритм VAD для кодера IS-96 – аналізує загальну енергію сигналу. З іншого боку, алгоритм VAD кодерів IS-127 і IS-733 також розкладають сигнал, але тільки на два піддіапазону.

Таблиця 2.1 Класифікація стандартизованих методів VAD в залежності від способу аналізу сигналу (в дужках вказано кількість спектральних піддіапазонів)

Спосіб аналізу	VAD
Аналіз спектральної форми	GSM-FR, GSM-HR, GSM-EFR
Аналіз енергії піддіапазонів	IS-96, IS-127, IS-733, AMR1, AMR2
Інше	G.729B, G.723.1A

Традиційно методи VAD Європейського інституту стандартизації ETSI були засновані на більш точному аналізі — аналізі спектральної форми вхідного сигналу. Причиною цього є те, коли спектральні форми фону і вхідного сигналу не збігаються (наприклад, в разі активної мови), тому помилки кодування збільшується. Проте, в останньому стандарті AMR прийняті два види алгоритмів VAD, кожен з яких заснований на аналізі спектральної енергії піддіапазонів, а не на більш точному аналізі форми спектра. Стандартизовані методи VAD в кодерах G.729B і G.723.1A ведуть виявлення за допомогою чотирьох різних способів, включаючи як аналіз спектральної форми, так і аналіз енергії піддіапазонів.

2.1. Алгоритми VAD кодерів G.729B / G.723.1A

Як розширення до кодеру мови G.729 міжнародний комітет ITU-T SG16 випустив кодер G.729 Annex B з метою підтримки переривчастої передачі DTX допомогою детектування активності мови VAD, аналізу фонового шуму CNI і генерації комфортного шуму CNG. Кодер G.729B ділить мова на інтервали по 10

мс і виробляє рішення про наявність чи відсутність мови для кожного фрейму, оцінюючи при цьому чотири параметри [3, 5]:

- різниця енергій всього діапазону — $\Delta E_f = \overline{E_f} - E_f$,
- різниця енергій діапазону НЧ — $\Delta E_l = \overline{E_l} - E_l$,
- спотворення спектра — $\Delta LSF = \sum_{i=0}^{9} \left(\overline{LSF_i} - LSF_i \right)^2$,
- різницю частоти переходів через нуль $\Delta ZC = \overline{ZC} - ZC$

де E_f – енергія всього діапазону, E_l – енергія діапазону НЧ, LSF_i – і-а частота спектра сигналу і ZC – частота переходів через нуль вхідного сигналу, $\overline{E_f}$, $\overline{E_l}$, $\overline{LSF_i}$, $\overline{ZC}$ – параметри, що характеризують шум і оновлюються за допомогою аналізу фонового шуму.

Блок схема алгоритму VAD кодера G.729B представлена на Рис. 2.1 [5]. Вхідні параметри для аналізу VAD можуть бути отримані з вхідного сигналу або з проміжних значень мовного кодера. Потім розраховуються параметри різниці між параметрами вхідного сигналу і шуму ΔE_f , ΔE_l , ΔLSF , ΔZC .

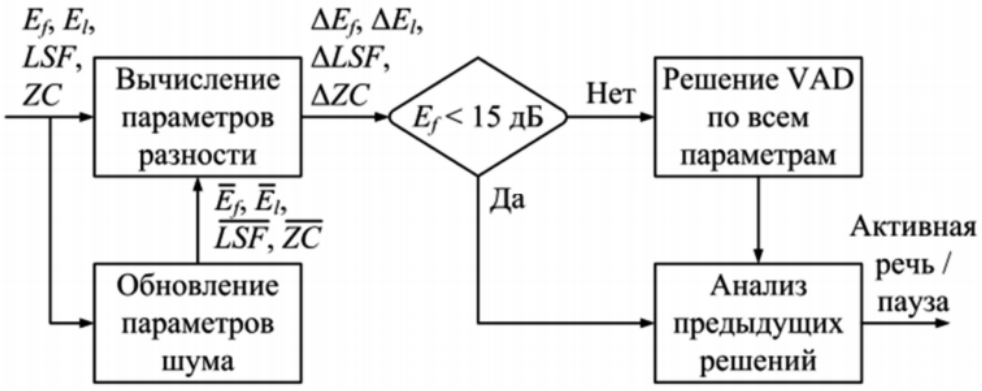

Рис. 2.1 Алгоритм VAD кодера G.729B

Рішення про наявність мови отримують шляхом аналізу інтервалів голосового сигналу за чотирма параметрами, які надходять на схему аналізу попередніх рішень. Блок поновлення параметрів шуму заснований на схемі авторегресії першого порядку. Вони оновлюються, якщо різниця енергії всього діапазону менше заданого фіксованого порога. Алгоритм VAD кодера G.723.1A

має аналогічну структуру. Відмінність полягає в тому, що в кодер G.723.1A ділить мова на інтервали по 30 мс.

2.2. Алгоритми VAD кодерів GSM-FR / HR / EFR

Алгоритми VAD кодеров GSM-FR / HR / EFR, запропоновані ETSI, мають схожу структуру. Тут передбачена залишкова енергія порівнюється з адаптивним порогом. Передбачена залишкова енергія обчислюється з використанням значень дійсної і згладженої автокореляції, які описують спектральні характеристики сигналу.

Передбачається, що якщо сигнал є тільки фоновим шумом, який зазвичай вважається стаціонарним, то середня спектральна форма буде подібна формі спектра поточного фрейма, що призведе до меншої залишкової енергії сигналу. Поріг рішення алгоритму VAD оновлюється тільки протягом інтервалів «немови», використовуючи найостанніші сигнали шуму, щоб відображати актуальні шумові характеристики. Блок схема алгоритмів VAD кодерів GSM-FR / HR / EFR представлена на Рис. 2.2 [9].

Рис. 2.2 Алгоритм VAD кодера GSM-EFR

2.3. Алгоритми VAD кодерів AMR

Кодер AMR1 розкладає вхідний сигнал на дев'ять нерівномірних піддіапазонів за допомогою банку фільтрів, де нижні смуги частот мають меншу смугу пропускання, а більш високі частотні діапазони — велику пропускну

здатність. Потім він обчислює енергію в кожному піддіапазоні, супроводжувану її відповідною оцінкою ЗСШ.

Енергія фонового шуму, використовувана в розрахунках ЗСШ, обчислюється адаптивним методом на основі авторегресивної моделі першого порядку і внутрішньої логічної схеми VAD. Нарешті, рішення VAD приймається шляхом порівняння суми ЗСШ піддіапазонів з адаптивним порогом і надходить на схему аналізу попередніх рішень. Блок схема алгоритму VAD кодера AMR1 представлена на Рис. 2.3 [10].

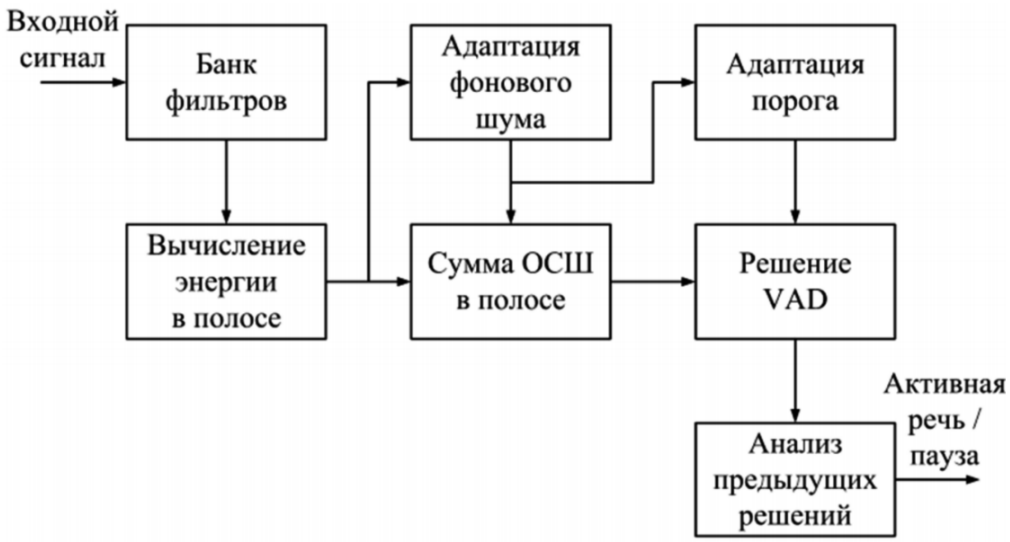

Рис. 2.3 Алгоритм VAD кодера AMR1

Структура кодера AMR2 схожа на AMR1 в тому, що детектування мовної активності здійснюється з використанням інформації про енергію в піддіапазонах разом з енергією фонового шуму. Однак алгоритм VAD кодера AMR2 трансформує вхідний сигнал в частотну область, використовуючи БПФ замість банку фільтрів, що застосовується в AMR1, і потім обчислює енергію в кожному піддіапазоні. Число смуг дорівнює 16, ширина смуг також є нерівномірною, як і в AMR1.

Згодом, за спектрами вхідного сигналу і фонового шуму, обчислюється ЗСШ для кожного піддіапазону. Енергія фонового шуму для кожної смуги змінюється під час інтервалів «немови» за допомогою авторегресивної схеми першого порядку. Щоб запобігти можливості виникнення надмірної чутливості до

умов нестаціонарного фонового шуму, AMR2 збільшує поріг прийняття остаточного рішення алгоритму VAD для високо флуктуючих сигналів, які оцінюються за дисперсії їх миттєвих міжкадрових ЗСШ. Крім того, адаптація рівня шуму шляхом вимірювання девіації спектра може проводитися неточно, якщо рівень енергій піддіапазонів змінюється швидко.

Таким чином, кодер AMR2 змінює поріг VAD адаптивним способом, орієнтуючись також на рівень сплеску і прийняті рішення по минулим інтервалах. Контроль за прийнятими рішеннями здійснюється шляхом вимірювання відносини пікового значення сигнал/шум до середнього, де середнє значення ЗСШ розраховується за допомогою авторегресивної адаптації зі збільшеним миттєвим ЗСШ. Іншими словами, для збільшення відносини пікового значення сигнал/шум до середнього, зменшуються розрахунки по минулим значенням і рівня сплеску, поряд зі збільшенням порогу алгоритму VAD. Блок схема алгоритму VAD кодера AMR2 представлена на Рис. 2.4 [11].

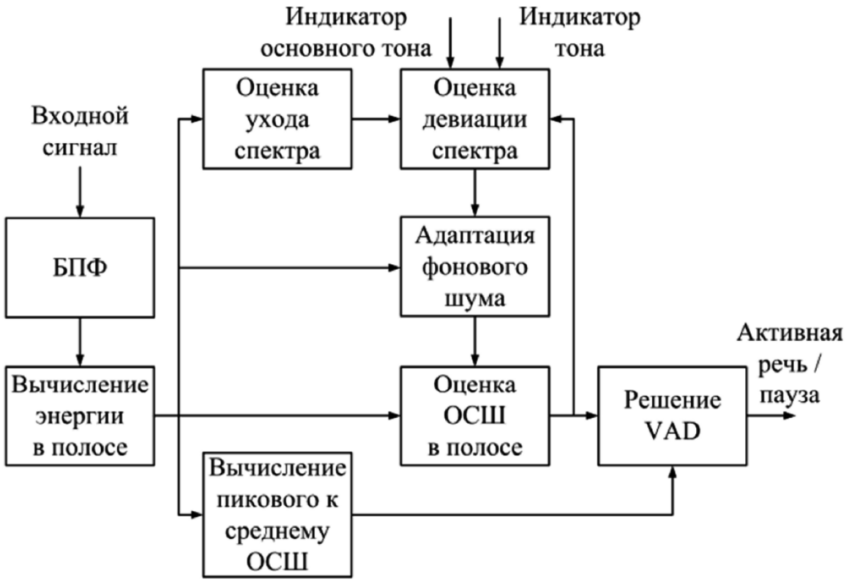

Рис. 2.4 Алгоритм VAD кодера AMR2

2.4. Алгоритми VAD кодерів IS-127/133

Засновані на CDMA цифрові системи мобільного зв'язку мають природну структуру для вбудовування алгоритму VAD, званого алгоритмом визначення швидкості (rate determination algorithm — RDA), який дає істотне збільшення

канальної ємності шляхом контролю потужності радіопередачі з метою ослаблення міжканального інтерференції.

Асоціації TIA / EIA представили два види алгоритмів RDA для стандартів кодеров IS-96 і IS-127, звані кодером з лінійним передбаченням з мультікодовим управлінням Qualcomm (Qualcomm code-excited linear prediction — QCELP) 8 кбіт/с і кодеком з розширеною змінною швидкістю (enhanced variable rate codec — EVRC), відповідно. У Північній Америці стандарт CDMA RDA IS-127 підтримує три швидкості: 1, 1/2 і 1/8. Інтервали, в яких міститься мова, кодуються зі швидкістю 1 або 1/2, а фоновий шум зі швидкістю 1/8. Алгоритм RDA кодера IS-733 називають 13 кбіт/с QCELP, він такий же, як і алгоритм RDA кодера IS-127.

В якості вхідних параметрів алгоритм RDA кодера IS-127 використовує енергії двох піддіапазонів зі збільшенням довгострокового передбачення. Перш за все, за допомогою авторегресивної моделі першого порядку, обчислюється згладжена енергія в піддіапазонах. Потім проводиться адаптація енергій сигналу і шуму в кожному піддіапазоні в залежності від посилення довгострокового передбачення. Іншими словами, енергія сигналу активно адаптується до поточного значення на вході, якщо коефіцієнт посилення передбачення щодо високий.

З іншого боку, якщо коефіцієнт посилення є невеликим, це дозволяє збільшити швидкість адаптації шуму. ЗСШ в кожному з двох піддіапазонів обчислюється з використанням інформації про енергії сигналу і шуму в кожній підполосі. Остаточна швидкість визначається шляхом порівняння відносин сигнал/шум зі значеннями адаптивних порогів, що залежать від рівня фонового шуму і ЗСШ попереднього фрейму, вступаючи потім на схему аналізу попередніх рішень. Блок схема RDA кодера IS-127 представлена на Рис. 2.5 [12].

Рис. 2.5 Алгоритм RDA кодера IS-127

2.5. Загальна порівняльна характеристика.

В [16] була проведена порівняльна оцінка п'яти стандартизованих алгоритмів VAD на предмет кількості помилок виявлення для інтервалів активної мови і паузи. Тестовим сигналом була мова тривалістю 96 секунд, перетворена за допомогою модифікованої системи проміжного відгуку, а потім змішана з транспортним шумом з ЗСШ: 5, 10, 15, 20 і 25 дБ. Інтервали активної мови і паузи були відзначені вручну.

Пропорції між неактивними і активними ділянками мови були 0,43 і 0,57, відповідно. Рішення алгоритмом VAD для кодерів G.729B і AMR2 приймається кожні 10 мс, для кодерів GSM-EFR, AMR1 і IS-127-кожні 20 мс. При незначній модифікації вихідного коду AMR2, результати можна отримувати кожні 10 мс, тому що в своїй основі AMR2 приймає рішення кожні 10 мс і потім повертає рішення по інтервалу тривалістю 20 мс, аналізуючи логічний комбінацію рішень за двома інтервалах тривалістю 10 мс.

Що стосується багатошвидкісного кодера IS-127, дві верхніх швидкості (1 і 1/2) застосовуються для кодування активної мови, а нижня швидкість (1/8) використовується для кодування «немови». Продуктивність роботи алгоритмів в середовищі транспортного шуму показана на Рис. 2.6 і Рис. 2.7 [16]. Осцилограми тестового сигналу і результатів роботи детекторів для ЗСШ 15 дБ представлені на Рис. 2.8 [16].

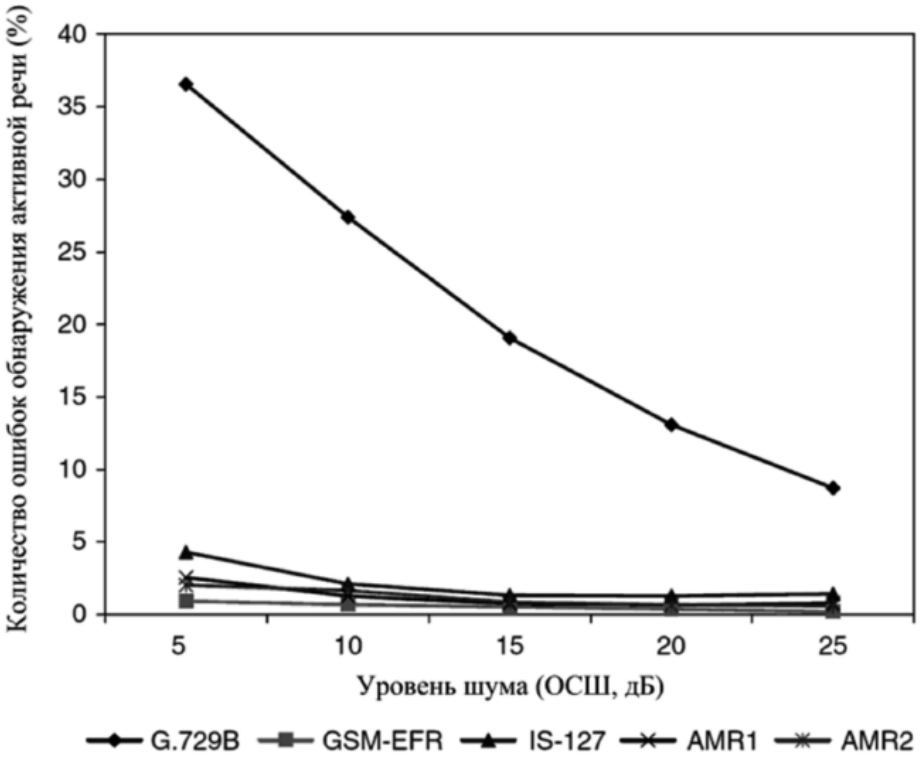

Рис. 2.6 Кількість помилок виявлення активної мови по відношенню до рівня транспортного шуму

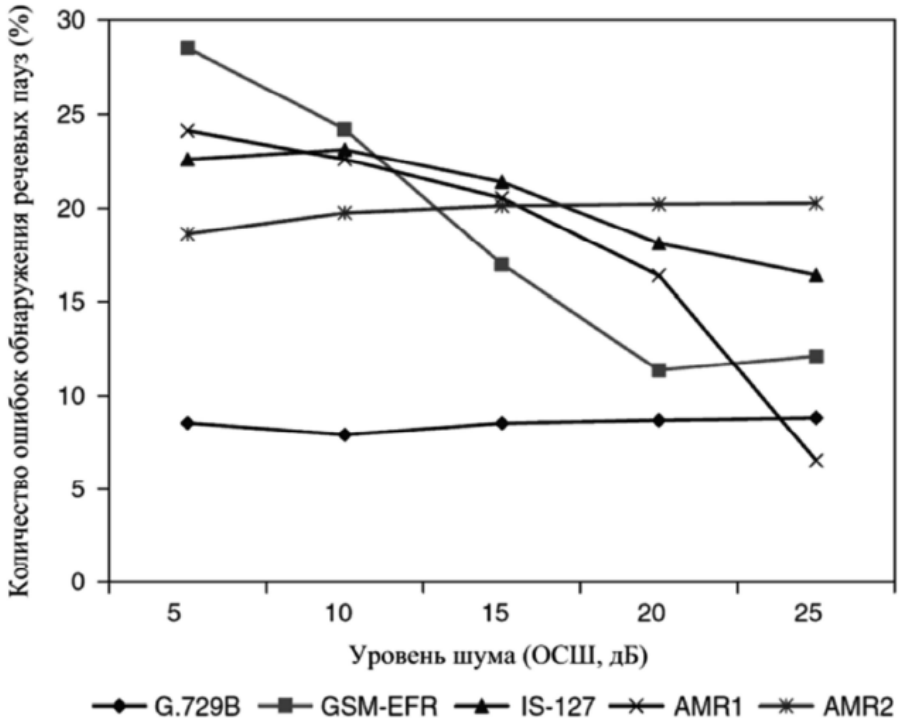

Рис. 2.7 Кількість помилок виявлення мовних пауз по відношенню до рівня транспортного шуму

Кодер G.729B показує хороший результат у порівнянні з іншими методами при виявленні мовних пауз. Однак він демонструє дуже високий рівень помилок детектування мови, що може спричинити різке обрізання сигналу мовлення.

Кодер IS-127 показує відносно високу кількість помилок при детектуванні активної мови в порівнянні з алгоритмами VAD, запропонованими ETSI.

Методи VAD Європейського інституту стандартизації ETSI, тобто алгоритми VAD кодеров GSM-EFR, AMR1 і AMR2, показують досить схожі результати у виявленні інтервалів активної мови, в той час як результати детектування інтервалів пауз сильно різняться. Кодер GSM-EFR демонструє найкращі результати для порівняно високого рівня ЗСШ (більше 15 дБ). Проте, кількість помилок виявлення інтервалів пауз при зменшенні рівня ЗСШ істотно зростає.

Кодер AMR2 показує щодо послідовні результати, не дивлячись на зміну рівня шуму, при виявленні ділянок пауз для мовного сигналу з транспортним шумом. За продуктивністю AMR1 знаходиться між GSM-EFR і AMR2. Як видно з наведених графіків, жоден метод не демонструє високу вірогідність і точності виявлення активної мови і мовних пауз одночасно. Таким чином, постає проблема поділу мовного сигналу на інтервали активної мови і мовних пауз з високим ступенем достовірності і найменшими втратами.

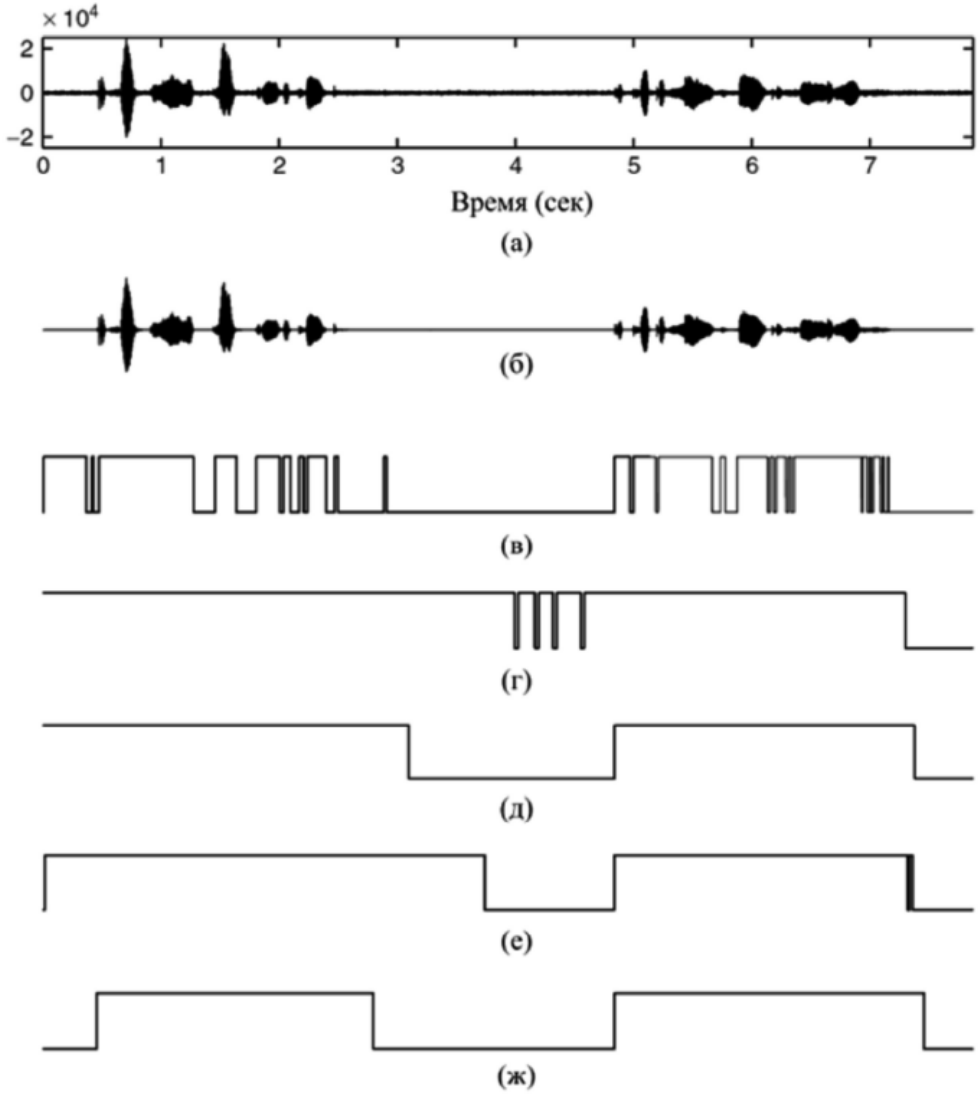

Рис. 2.8 Порівняння результатів роботи алгоритмів VAD для мовного сигналу з транспортним шумом (ЗСШ 15 дБ): а) вхідний зашумленний мовної сигнал;

ВИСНОВОК

В даний час продовжують активно розвиватися технології аналізу мовних сигналів, класифікації акустичних подій, фільтрації, а також голосові інтерфейси людино-машинного взаємодії, які вирішують завдання розпізнавання мови і голосового біометрії. Для відповідних систем важливу роль відіграють детектори акустичних подій.

Принцип детектування

У даній роботі запропонований детектор голосової активності, що забезпечує істотне підвищення точності правильного поділу мовних сигналів на періоди активної мови і паузи.

Структурна схема детектора активності мови зображена на Рис. 2.9 [14]. Мовний сигнал з виходу електроакустичного перетворювача посилюється селективним підсилювачем і подається на вхід суматора. На другий вхід суматора подається сигнал з виходу генератора вимірювального сигналу. Сумарний сигнал з виходу суматора надходить на вхід підсилювача-обмежувача 1, де відбувається посилення сигналу, а потім обмеження по амплітуді.

Рис. 2.9 Структурна схема детектора активності мови

Аналогічна операція проводиться над сигналом, що надходять з виходу селективного підсилювача на вхід підсилювача-обмежувача 2. Сигнал з виходу підсилювача-обмежувача 1 подається на перший вхід перемножувача. На другий вхід перемножувача подається сигнал з виходу підсилювача-обмежувача 2. Сигнал з виходу перемножувача надходить на вхід накопичувача-усереднювача (інтервал накопичення 10 мс), де відбувається обчислення сигналу, по амплітуді

якого приймають рішення про наявність періоду активного мовного сигналу або паузи в пороговому пристрої.

Значення порога обчислюється в схемі «обчислення порога» шляхом аналізу перших 150 мс від початку аналізу, тому що на цьому інтервалі мова зазвичай відсутня. Для проведення дослідження була обрана тестова фраза: «Продовження налагодження пристрою». На Рис. 2.10 [1] представлена осцилограма цієї фрази з додаванням транспортного шуму (ЗСШ 15 дБ), чистого мовного сигналу і сигналу з виходу детектора. Загальний час запису мовного сигналу зазначеної фрази склало 5 с, а сумарний час активних періодів — 2,64 с, що становить 52,8% часу обраного мовного сигналу. Кількість помилок виявлення активної мови склало 3,56%, кількість помилок виявлення мовних пауз — 1,61%.

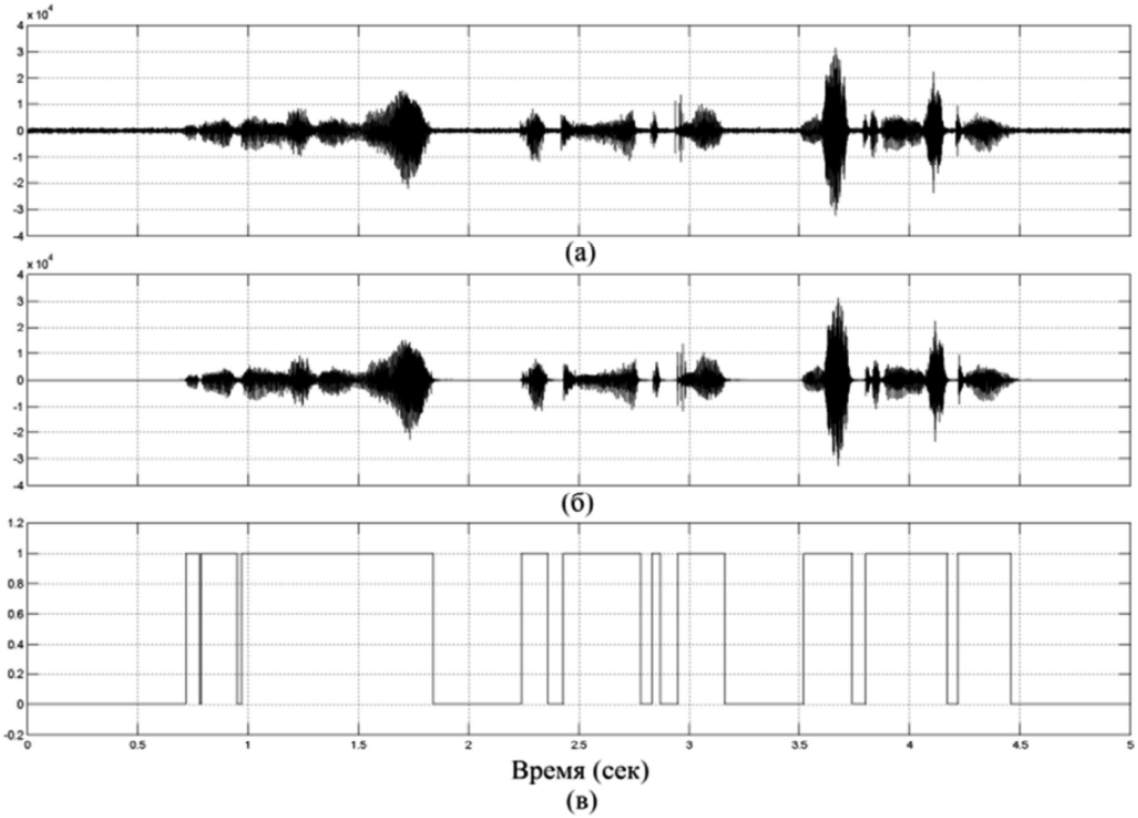

Рис. 2.10 Осцилограма тестової фрази з додаванням транспортного шуму (ЗСШ 15 дБ), чистого мовного сигналу і сигналу з виходу детектора

Для перевірки сприйняття мови на слух після обробки було зроблено запис мовного сигналу за допомогою блоку аудіозапису, керованого командами з детектора пауз. Інтервали, в яких присутня мова, залишалися без зміни. В

інтервали, в яких знаходилися паузи, записувався нуль. Якість отриманого мовного сигналу практично не відрізнялося від початкового. Слова були добре помітні, мова легко сприймалася на слух.

Детектування дефектів в мікрофонах

Детектори голосової активності можна використовувати для знаходження дефектів в масиві мікрофонів, так як це дозволяє зменшити час знаходження проблеми і знизити навантаження на процесор та елементи, які відповідальні за обробку мовного сигналу. Значна кількість систем з розпізнаванням голосу, які аналізують безпосередньо мовний сигнал і перетворюють оброблений сигнал в команду – не мають функцій перевірки самих елементів вводу, а саме мікрофонів.

Запропонований метод дозволяє використовувати обробку вхідних даних ближче до спростованих аудіо характеристик, як корисний (мова) та некорисний сигнал (шум). А, насамперед, системи розпізнавання голосу працюють на аналізі синтаксичних структур і побудові правильної команди для її виконання, що займає час для обробки даних та в більшості випадків потребує підключення до мережі Інтернет. Наприклад, при одночасній перевірці деякої кількості роботів з вбудованим комплексом розпізнаванням голосу користувач проговорює контрольну фразу і при отриманому сигналі з’являється перевищення значення шуму на одному з елементів масиву мікрофонів. При такій ситуації робот відправляє сигнал о непрацездатності його мікрофону. Аналогічну ситуації можна накласти і на системи розумних помічників Alexa©, Siri© чи Google Assistant©. Треба також розуміти, даний метод не є ідеальним, так як та же перевірка може проходити лише у середовищі з низьким рівнем зовнішнього шуму.

В даній роботі була створена модель по принципу детектора голосової активності, де використовувались вбудовані голосові доріжки для аналізу. Також приведена робота з пристроєм Amazon Echo Dot© та сервісами для налаштування власних команд і задач Alexa Skill©. Так як для реєстрації аккаунта повноцінного розробника коштує не підйомну для меня ціну та був створений аккаунт із обмеженими можливостями.

3. Створення моделі в MATLAB® та приклади для функцій

Виявлення голосової активності є важливим компонентом багатьох аудіосистем, таких як автоматичне розпізнавання мови та розпізнавання динаміків. Виявлення голосової активності може бути особливо складним у ситуаціях із низьким рівнем сигнал / шум (SNR), коли мова перешкоджає шуму.

У цьому прикладі використовуються мережі довгої короткочасної пам'яті (LSTM), які є різновидом рекурентних нейронних мереж (RNN), добре придатних для вивчення даних послідовності та часових рядів. Мережа LSTM може вивчати довгі короткочасні залежності між часовими кроками послідовності. Шар LSTM (lstmLayer) може дивитись на часову послідовність у прямому напрямку, тоді як двонаправлений шар LSTM (bilstmLayer) може дивитись на часову послідовність як у прямому, так і в зворотному напрямку. У цьому прикладі використовується двонаправлений рівень LSTM.

Цей приклад навчає двонаправлену мережу LSTM з виявленням голосової активності з послідовностями характеристик спектральних характеристик та метрикою гармонічного співвідношення. У сценаріях високого SNR традиційні алгоритми виявлення мови працюють належним чином. [18]

```
fs = 16e3;

[speech,fileFs] = audioread('Counting-16-44p1-mono-15secs.wav');

speech = resample(speech,fs,fileFs);

speech = speech/max(abs(speech));

sound(speech,fs)
```

Використовуємо функцію detectSpeech (Audio Toolbox), щоб знайти регіони мови. Функція detectSpeech правильно визначає всі регіони мови. [18]

```
[noise,fileFs] = audioread('WashingMachine-16-8-mono-200secs.mp3');
noise = resample(noise,fs,fileFs);

SNR = -20;
noiseGain = 10^(-SNR/20) * norm(speech) / norm(noise);

noisySpeech = speech + noiseGain*noise(1:numel(speech));
noisySpeech = noisySpeech./max(abs(noisySpeech));

sound(noisySpeech,fs)
```

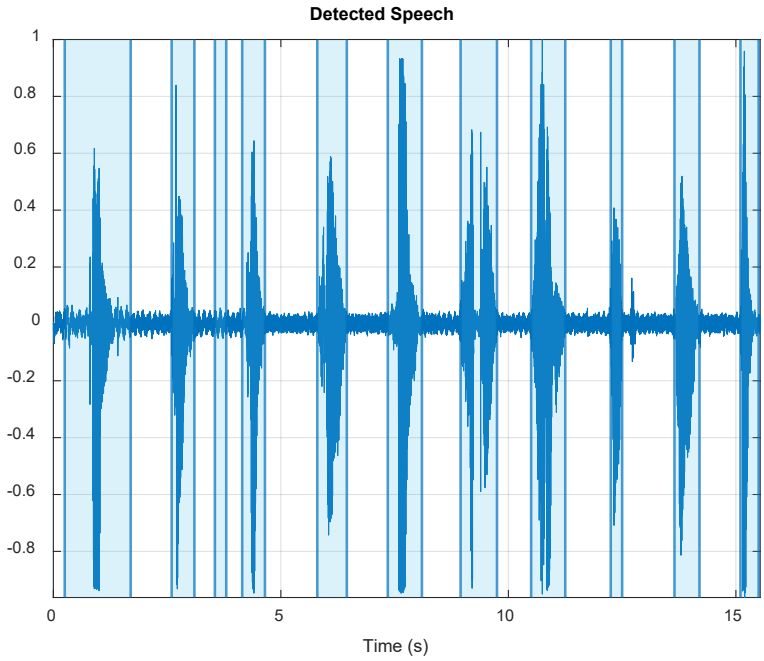

Рис. 3.1 Сигнал зі знайденими регіонами мови

Пошкоджує звуковий сигнал шумом пральної машини при SNR -20 дБ. Слухайте пошкоджене аудіо. [18]

```
[noise,fileFs] = audioread('WashingMachine-16-8-mono-200secs.mp3');
noise = resample(noise,fs,fileFs);

SNR = -20;
noiseGain = 10^(-SNR/20) * norm(speech) / norm(noise);

noisySpeech = speech + noiseGain*noise(1:numel(speech));
noisySpeech = noisySpeech./max(abs(noisySpeech));

sound(noisySpeech,fs)
```

Використовуємо команду detect Speech на зашумлений звуковий сигнал. Функції не вдається виявити мовні регіони, враховуючи дуже низький рівень SNR. [18]

```
detectSpeech(noisySpeech,fs,'Window',win)
```

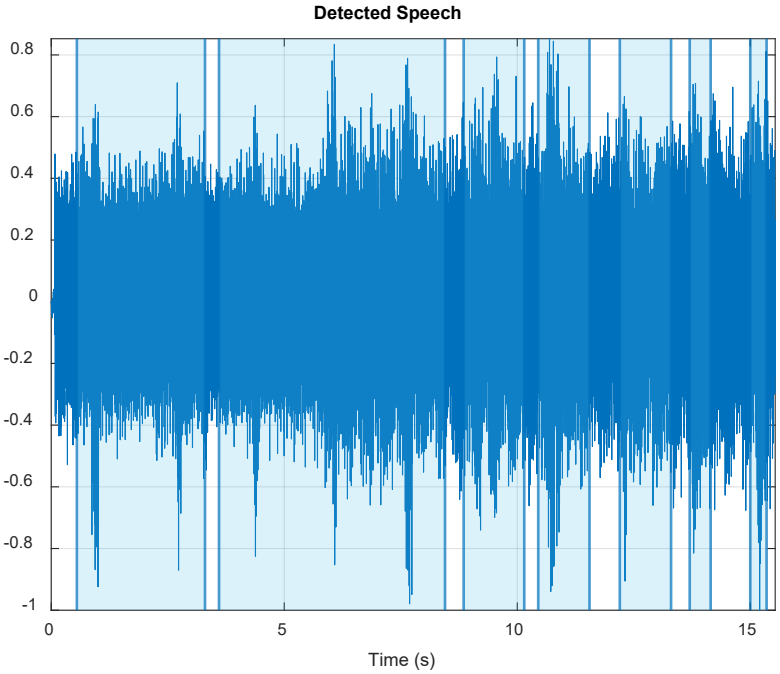

Рис. 3.2 Зашумлений сигнал

Завантажуємо попередньо навчений мережою та налаштований об’єкт audioFeatureExtractor (Audio Toolbox). Мережа була навчена виявляти мовлення в середовищах із низьким коефіцієнтом шуму, отримуючи функції, що виводяться з об’єкта audioFeatureExtractor. [18]

```
url = 'http://ssd.mathworks.com/supportfiles/audio/VoiceActivityDetection.zip';
 downloadNetFolder = tempdir;
 netFolder = fullfile(downloadNetFolder,'VoiceActivityDetection');
 if ~exist(netFolder,'dir')
 disp('Downloading pretrained network (1 file – 8 MB) ...')
 unzip(url,downloadNetFolder)
 end
load(fullfile(netFolder,'voiceActivityDetectionExample.mat'));
```

Витягнимо функції з мовних даних, а потім нормалізуємо їх. Зорієнтуємо функції так, щоб був час по стовпцях. [18]

```
features = extract(afe,noisySpeech);
features = (features – mean(features,1)) ./ std(features,[],1);
features = features';
```

Надамо ознаки через мережу виявлення мовлення, щоб класифікувати кожен вектор ознак як такий, що належить до кадру мови чи ні. [18]

```
decisionsCategorical = classify(speechDetectNet,features);
```

Кожне рішення відповідає вікну аналізу, проаналізованому за допомогою audioFeatureExtractor. Повторимо рішення так, щоб воно відповідало особистому аудіо зразками. Зробимо графік мови, мови з шумом та рішень VAD. [18]

```
decisionsWindow = 1.2*(double(decisionsCategorical)-1);
decisionsSample = [repelem(decisionsWindow(1),numel(afe.Window)), ...
  repelem(decisionsWindow(2:end),numel(afe.Window)-afe.OverlapLength)];
t = (0:numel(decisionsSample)-1)/afe.SampleRate;
plot(t,noisySpeech(1:numel(t)), ...
```

```
 t,speech(1:numel(t)), ...
 t,decisionsSample);
xlabel('Time (s)')
ylabel('Amplitude')
legend('Noisy Speech','Speech','VAD','Location','southwest')
```

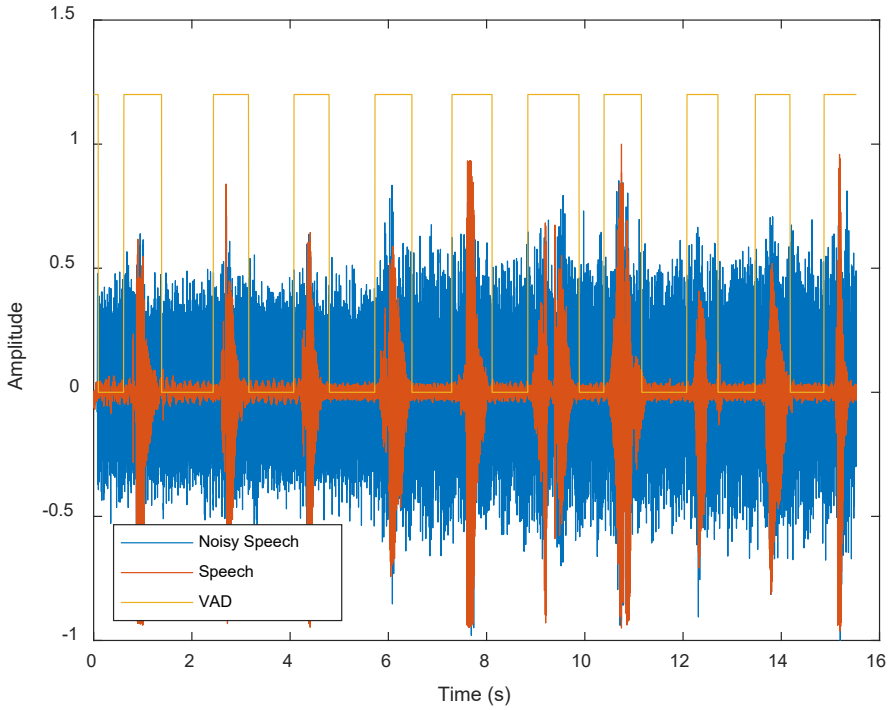

Рис. 3.3 Графік мови, мови з шумом та рішень VAD

Ми також можемо використовувати навчену мережу VAD у потоковому контексті. Щоб імітувати потокове середовище, спочатку треба зберегти мовні та шумові сигнали як файли WAV. Для імітації потокового введення ми зчитуємо кадри з файлів і змішуєте їх за потрібним SNR. [18]

```
audiowrite('Speech.wav',speech,fs)
audiowrite('Noise.wav',noise,fs)
```

3.1. Приклади використання функції detectSpeech.

Будування графіку виявлених регіонів мови.

Ця функція виявляє межі мови в звуковому сигналі.

Зчитуємо аудіосигнал. Закліпуємо звуковий сигнал на 20 секунд.

```
[audioIn,fs] = audioread('Rainbow-16-8-mono-114secs.wav');
audioIn = audioIn(1:20*fs);
```

Виконуємо detectSpeach. Не вказуйте вихідних параметрів для відображення графіку виявлених мовних областей. [19]

```
detectSpeech(audioIn,fs);
```

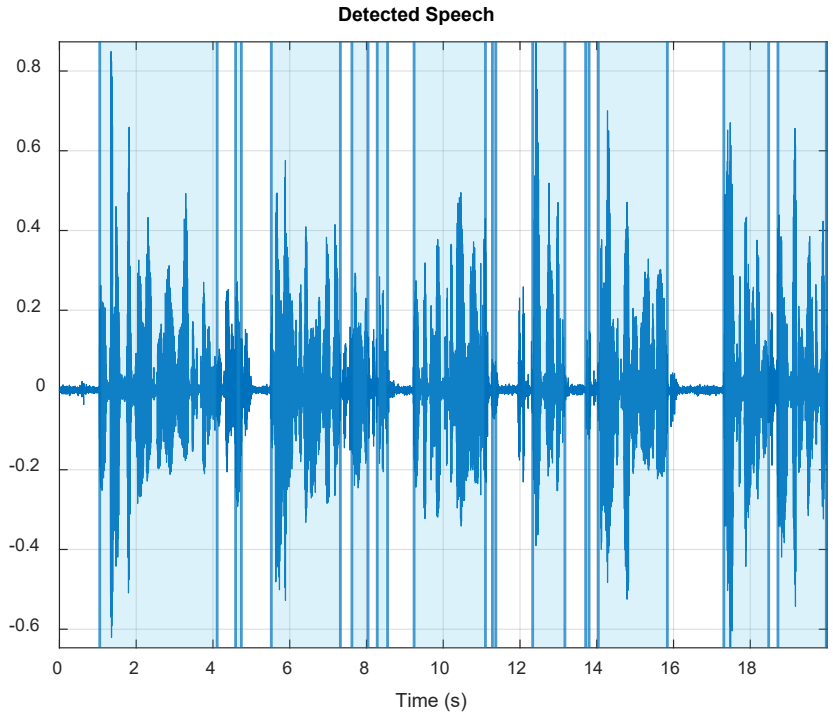

Рис. 3.4 Сигнал зі знайденими регіонами мови без вихідних параметрів для відображення графіку виявлених мовних областей

Ця функція використовує алгоритм порогового значення, заснований на енергетичному та спектральному розподілі на аналізу кадру. Ви можете змінити Window, OverlapLength та MergeDistance, щоб точно налаштувати алгоритм під ваші конкретні потреби. [19]

```
windowDuration = 0.074;
numWindowSamples = round(windowDuration*fs);
win = hamming(numWindowSamples,'periodic');
percentOverlap = 35;
overlap = round(numWindowSamples*percentOverlap/100);
mergeDuration = 0.44;
mergeDist = round(mergeDuration*fs);
detectSpeech(audioIn,fs,"Window",win,"OverlapLength",overlap,"MergeDistance",mergeDist)
```

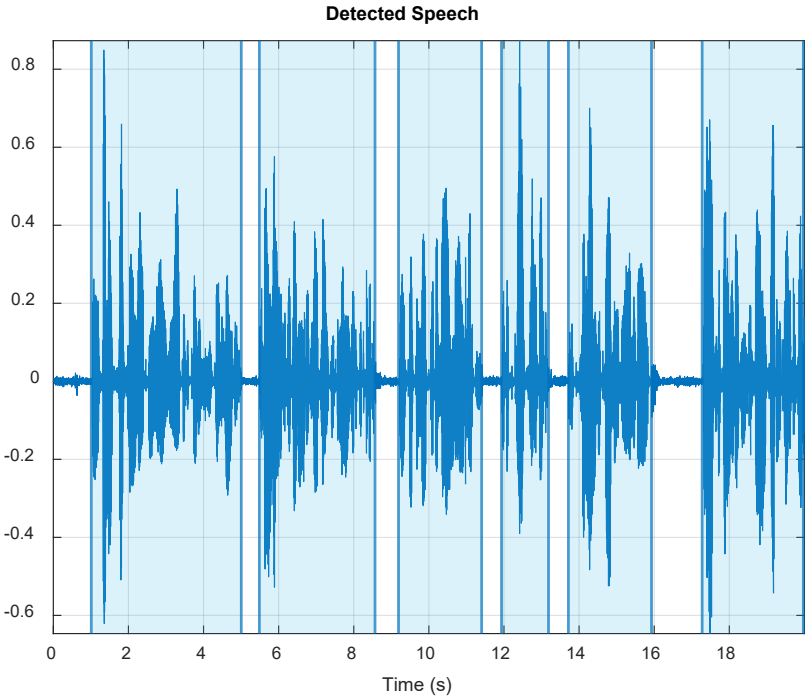

Рис. 3.5 Сигнал зі знайденими регіонами мови із вказаними вихідних параметрів для відображення графіку виявлених мовних областей

Видалення беззвучних регіонів з мовного сигналу.

Зчитуємо аудіофайл і прослуховуємо його. Будуємо спектрограму. [19]

```
[audioIn,fs] = audioread('Counting-16-44p1-mono-15secs.wav');
sound(audioIn,fs)
spectrogram(audioIn,hann(1024,'periodic'),512,1024,fs,'yaxis')
```

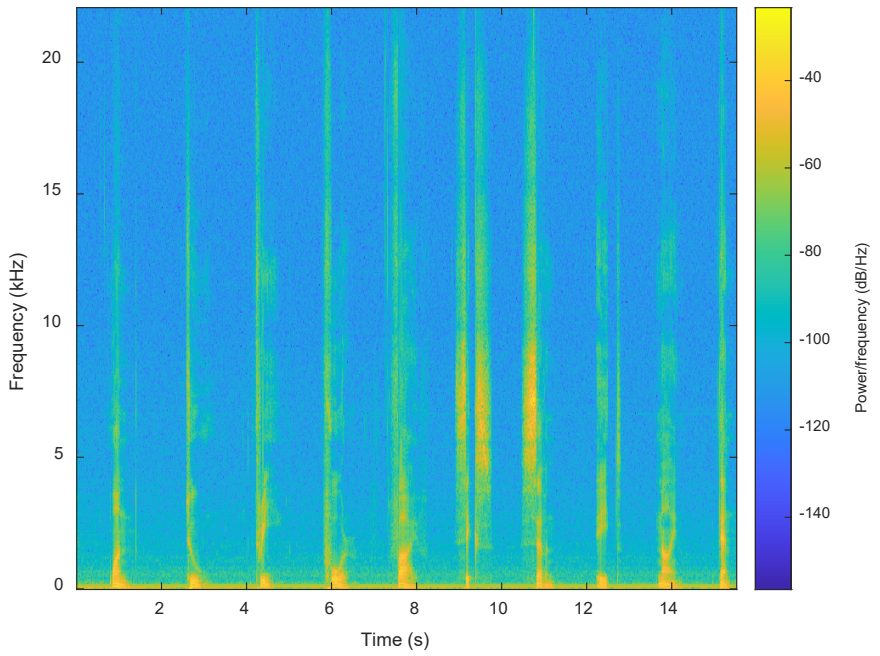

Рис. 3.6 Спектрограма голосового сигналу

Для програм машинного навчання часто потрібно захопити функції із аудіосигналу. Викличте функцію spectralEntropy на звуковому сигналі, потім побудуйте гістограму, щоб відобразити спектральний розподіл. [19]

```
entropy = spectralEntropy(audioIn,fs);
numBins = 40;
histogram(entropy,numBins,'Normalization','probability')
title('Spectral Entropy of Audio Signal')
```

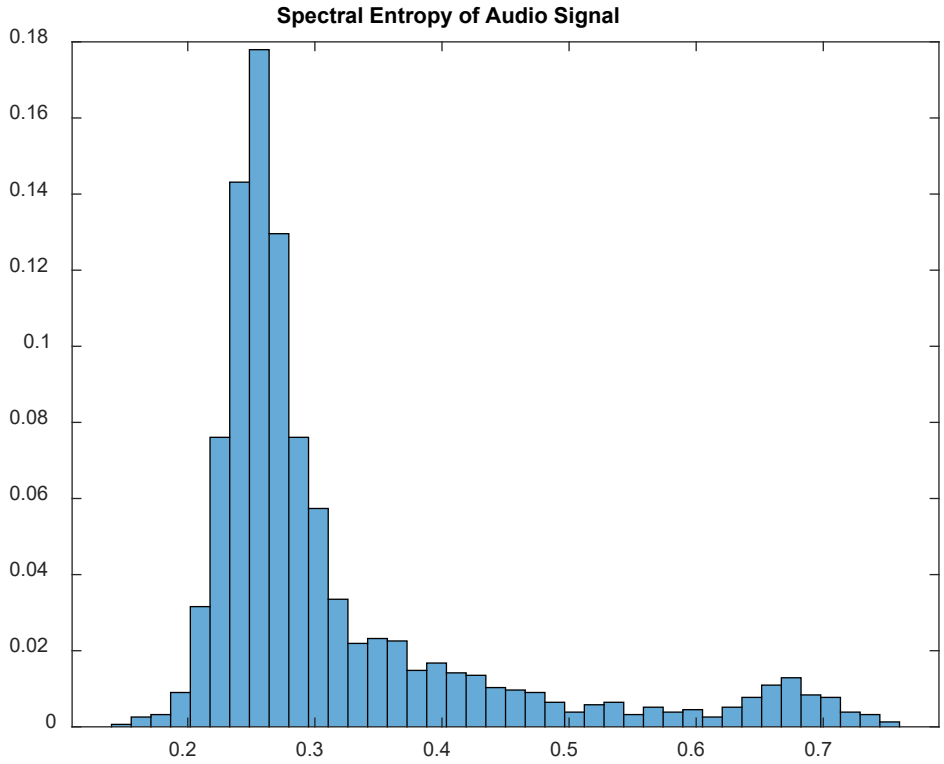

Рис. 3.7 Гістограма спектрального розподілу

Залежно від вашої програми, вам може знадобитися витягти спектральний розподіл лише з областей мови. Отримані статистичні дані більш характерні для динаміка і менш характерні для каналу. Використаємо detectSpeech на звуковий сигнал, а потім створимо новий сигнал, що містить лише області виявленої мови. [19]

```
speechIndices = detectSpeech(audioIn,fs);
speechSignal = [];
for ii = 1:size(speechIndices,1)
 speechSignal = [speechSignal;audioIn(speechIndices(ii,1):speechIndices(ii,2))];
end
```

Прослухаємо мовний сигнал і побудуємо графік спектрограми. [19]

```
sound(speechSignal,fs)
spectrogram(speechSignal,hann(1024,'periodic'),512,1024,fs,'yaxis')
```

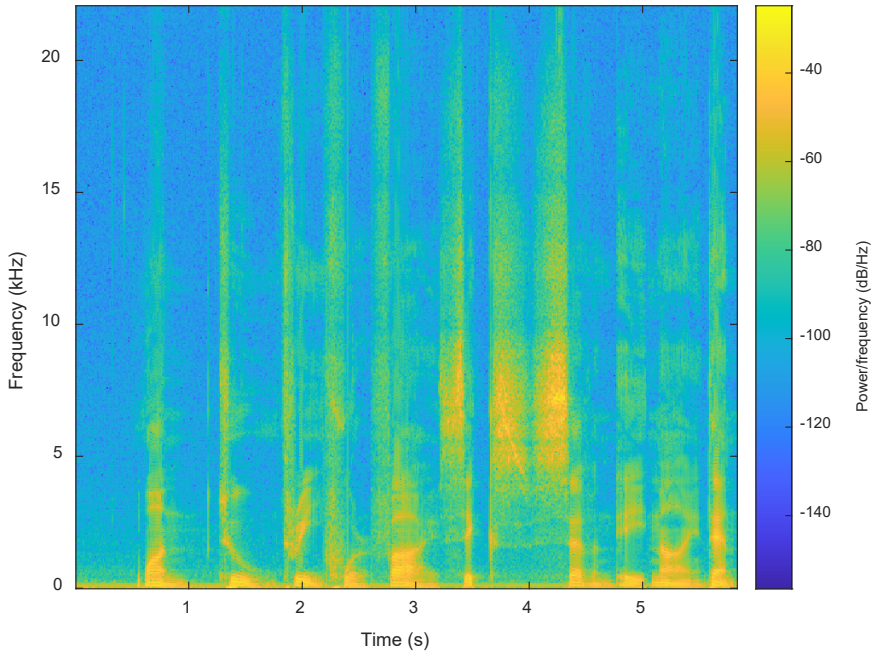

Рис. 3.8 Спектограма сигналу, що містить лише області виявленої мови

Застосуємо функцію SpectralEntropy для мовного сигналу, а потім побудуємо гістограму для відображення спектрального розподілу. [19]

```
entropy = spectralEntropy(speechSignal,fs);
histogram(entropy,numBins,'Normalization','probability')
title('Spectral Entropy of Speech Signal')
```

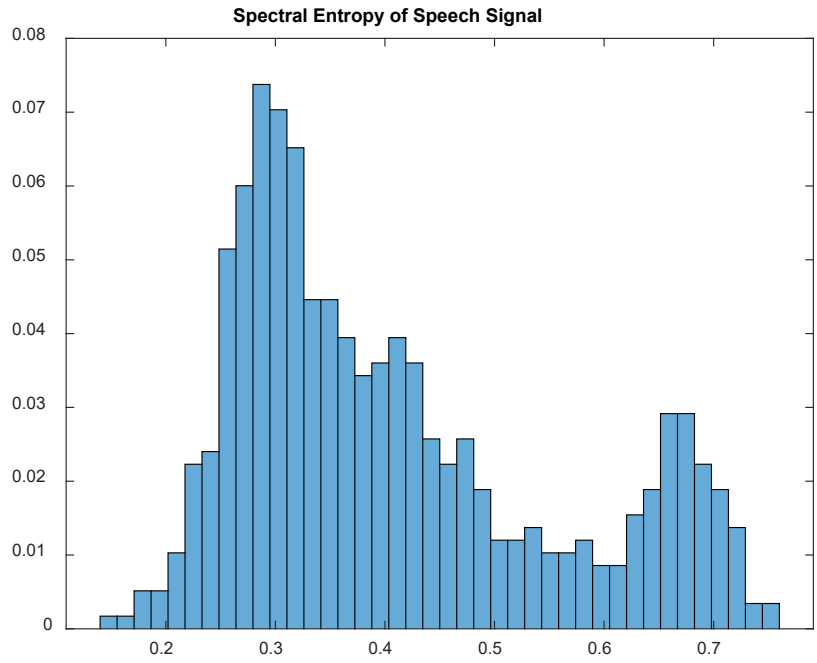

Рис. 3.9 Гістограма спектрального розподілу сигналу, що містить лише області виявленої мови

4. Принципова робота детектора в інтелектуальних системах

Тема розумних будинків, безсумнівно, перспективна . Проте для нашого часу це, на жаль, ця ідея лише тільки поширюється. Існує безліч факторів, які в деякій мірі можуть відштовхувати споживача від того, щоб використовувати даного типу технології в своєму будинку. Тут визначаються питання фінансової придатності: далеко не кожен громадянин може дозволити собі покупку будь-якого дорогого smart-пристрою.

4.1. Внутрішня структура

Потрібно зрозуміти, як ми в принципі можемо управляти оточуючими нас пристроями, навіть не підпадають під категорію розумних.

Здатність розумних колонок чути і розуміти співрозмовника на відносно великій відстані (кімната довжиною 6-8 метрів), незважаючи на сторонні шуми.

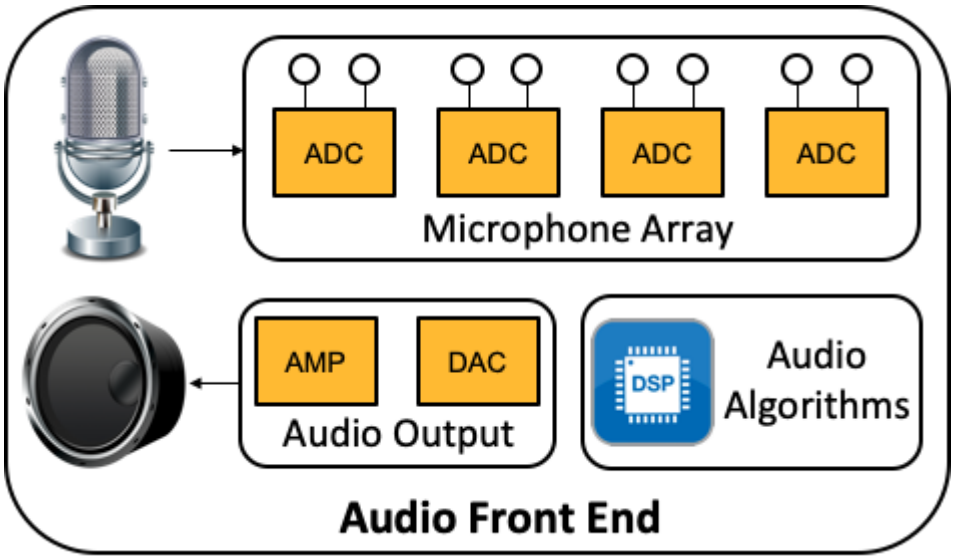

Рис. 4.1 Схема побудови мікрофонів та динаміків в пристрої Amazon Dot

На схемі Рис. 4.1 [17] ми бачимо цілий масив з чотирьох мікрофонів (характерно для Echo Dot 3, для інших моделей може відрізнятися), динамік і чіп з DSP-алгоритмами на борту.

З алгоритмів можна виділити наступні:

- wake word detection;
- audio feedback disposal;

- acoustic echo cancellation;
- beamforming.

Незважаючи на, здавалося б, ідеальне залізо, готові рішення теж не виняткові. З огляду на те, що розумні колонки націлені не тільки на домен smart home, розробники борються ще й з іншими проблемами, найбільш гострою серед яких, мабуть, є відсутність розуміння контексту оброблюваної інформації.

Для початку потрібно зрозуміти, як Alexa© інтерпретує призначені для користувача команди і що з ними відбувається далі.

Коли користувач висловлює намір взаємодіяти з будь-яким пристроєм, відбувається наступне:

1. Голосовий потік після попередньої обробки на залозі відлітає в Amazon® Cloud, де в кінцевому рахунку потрапляє на так званий Alexa© Skill;
2. Для Skill формується спеціалізований запит (директива), що містить інформацію про призначеному для користувача намір;
3. Директива відлітає на нашу функцію (back-end нашого Skill), де ми зобов'язані її обробити;
4. На цьому повноваження Amazon® закінчуються.

4.2. Alexa© Skill

Alexa© Skill – це, по своєї суті, програмне забезпечення для колонки. При цьому вся обробка проводиться на стороні Amazon. Існує кілька типів Skill, які ви можете розробляти. Але нам необхідний лише smart home skill.

Для старту вам знадобиться зареєструвати основний Amazon®-аккаунт і аккаунт розробника. Skill створюються саме за допомогою Dev-аккаунта. Але основний нам знадобиться для створення функції.

Після реєстрації йдемо в dev console та тиснемо Create Skill.

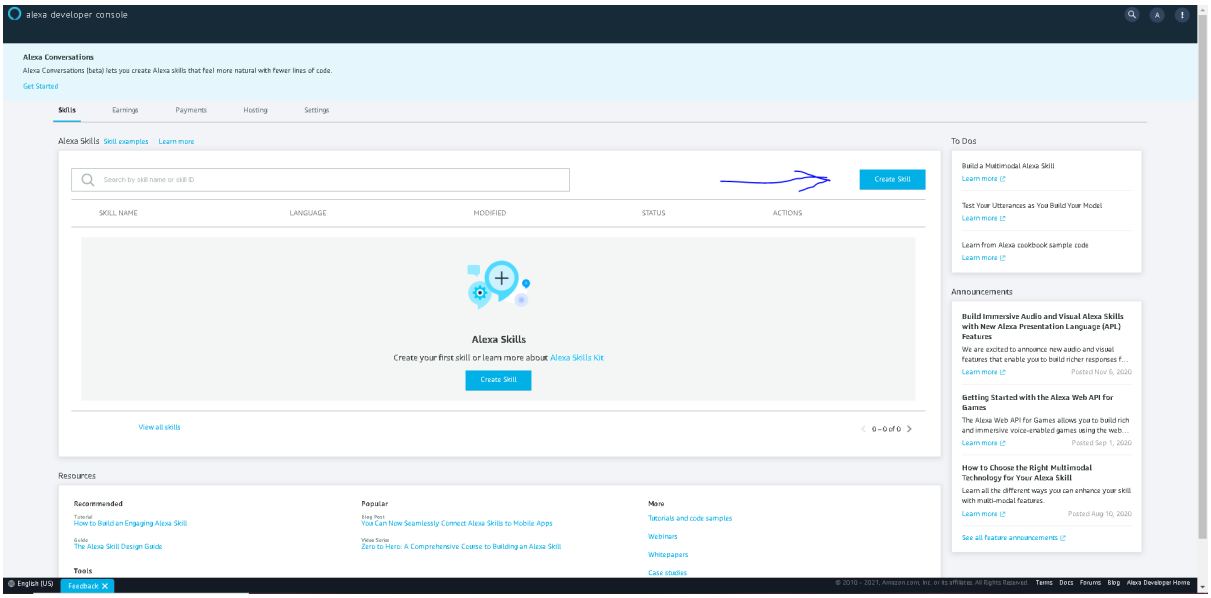

Рис. 4.2 Сайт Amazon Skill

На наступній сторінці даємо ім'я нашого Skilla і не забуваємо вибрати тип:

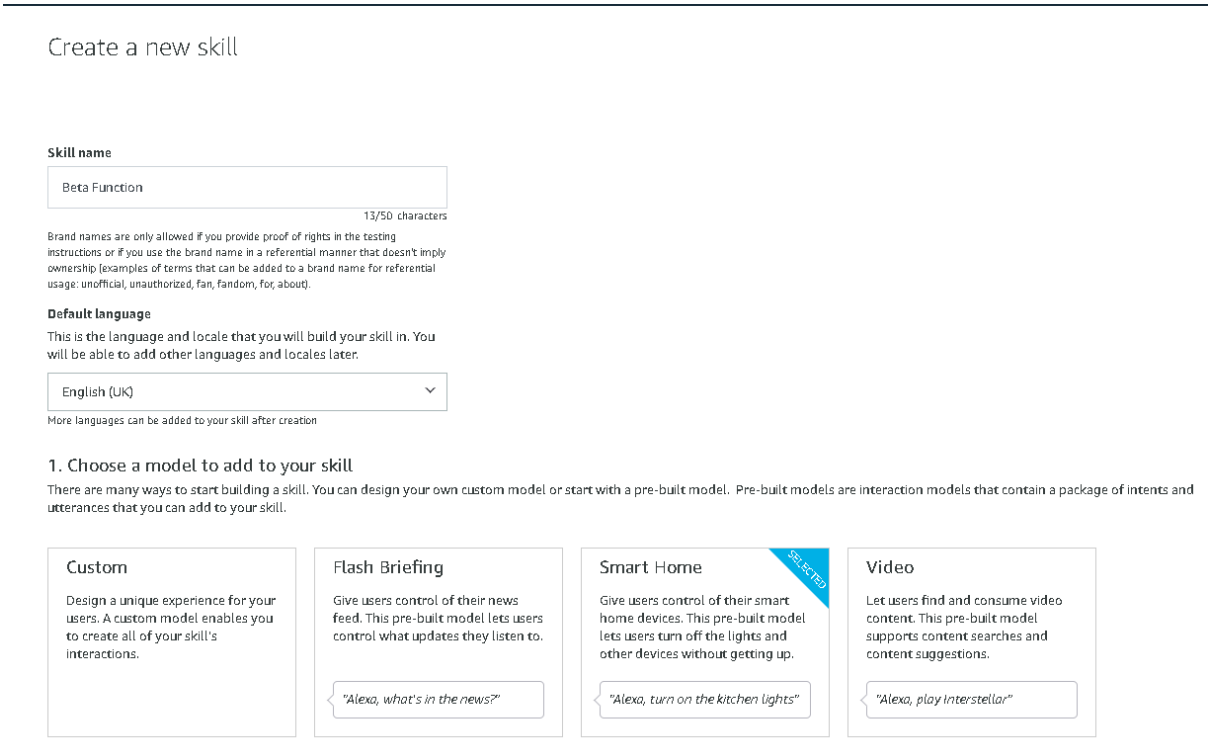

Рис. 4.3 Сайт створення Skill з налаштуванням назви, системної мови та моделі

Після створення Skill ви побачите наступну сторінку:

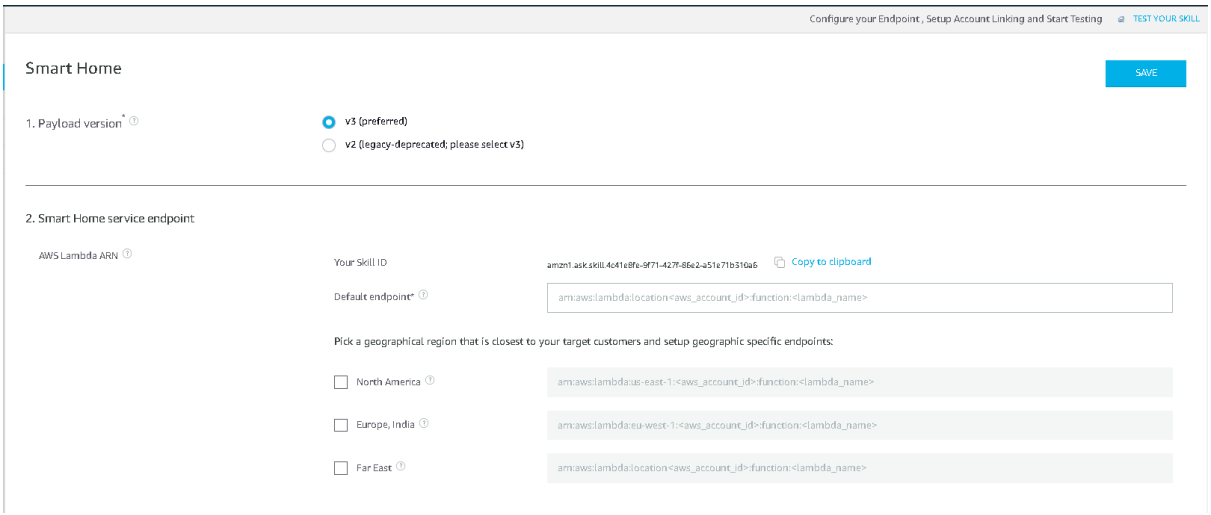

Рис. 4.4 Сайт з налаштуанням Skill та AWS Lambda

В якості back-end для нашого Skill ми можемо використовувати тільки AWS Lambda (на відміну від інших типів Skill). Прямуємо посиланням в самому низу — Setup Account Linking:

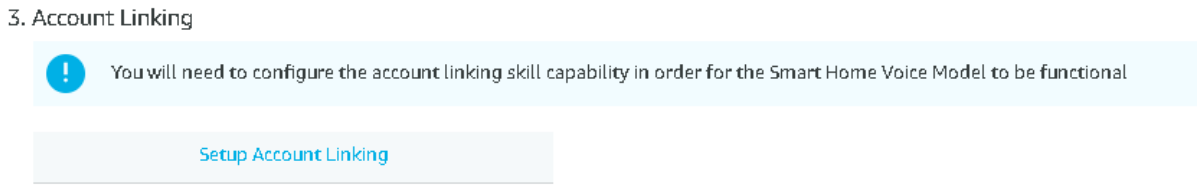

Рис. 4.5 Налаштування зв’язку облікового запису

Сторінка досить важлива, так як тут вам доведеться налаштувати доступ до свого oauth-провайдеру. Крок обов'язковий, без нього ви не зможете йти далі. Тут вам доведеться зробити вибір: використовувати або built-in-підтримку Amazon® Oauth, або якийсь існуючий сервіс по типу Auth0.

4.3. Amazon® Oauth

Йдемо на сторінку Login with Amazon і створюємо новий security profile.

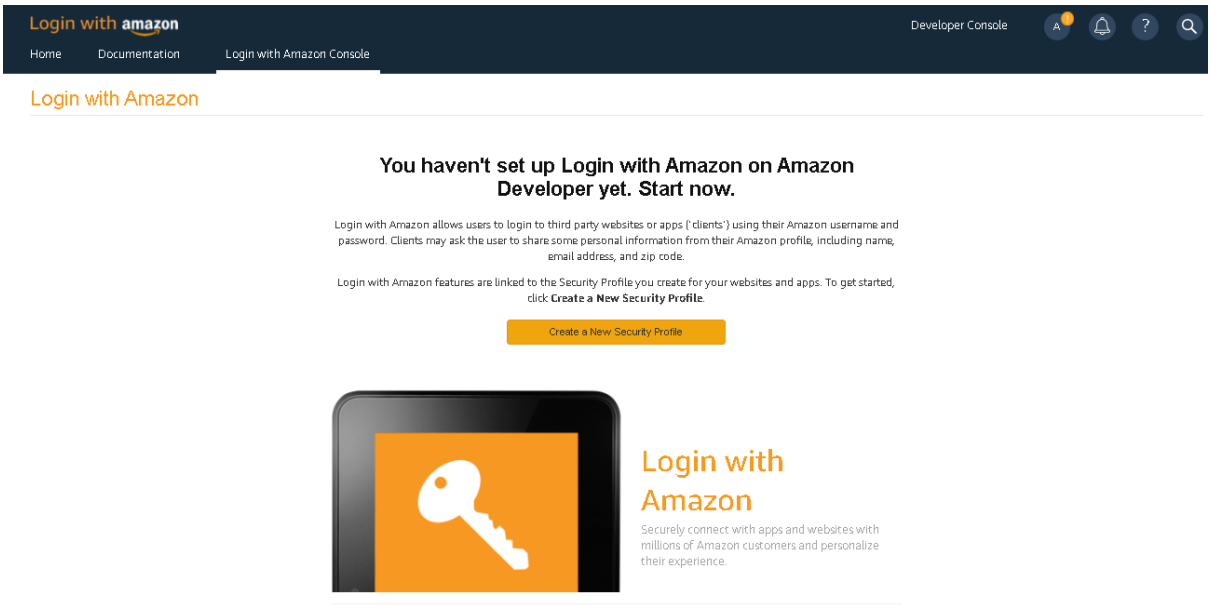

Рис. 4.6 Головна сторінка сайту Amazon Oauth

Заповнюємо обов'язкові поля (privacy notice URL для тестових цілей ставимо — http://example.com/privacy.html) і зберігаємо. На цьому етапі у нас вже згенеровані client id і secret, які знадобляться нам на Skill.

Відкриваємо Web Settings нашого профілю на редагування і оновлюємо поле Allowed Return URLs даними з поля Allowed Redirect URLs нашого Skill.

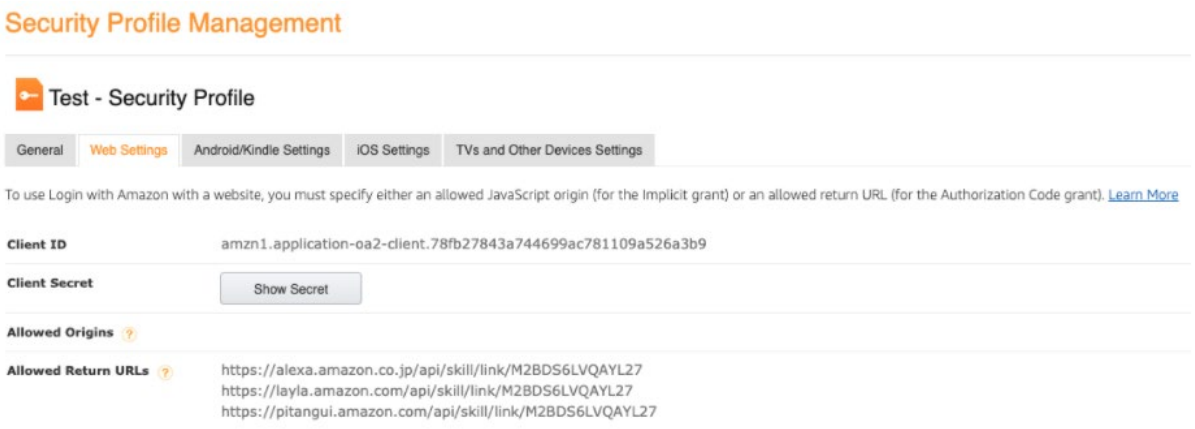

Рис. 4.7 Сайт Amazon Oauth з управління профілем безпеки

На самому ж Skill потрібно заповнити client id / secret згенерували значеннями з профілю:

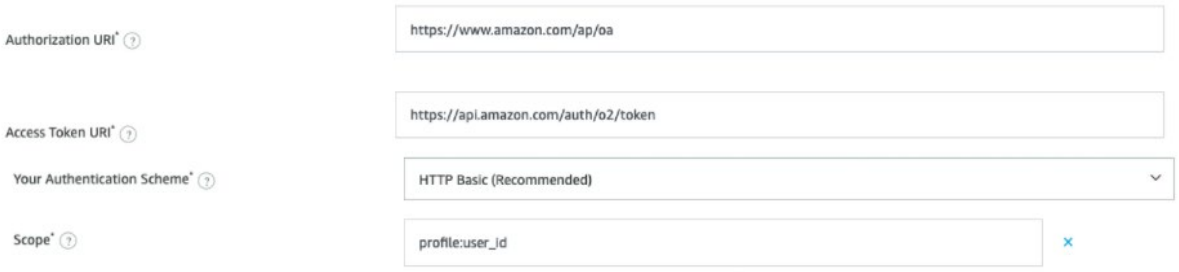

Рис. 4.8 Сайт Amazon Oauth налаштуванням аутентифікації

Інші поля заповнюємо. Зберігаємо і Oauth налаштований.

4.4. Auth0

Створюємо Auth0-аккаунт (краще скористатися варіантом з соцмережами). Потім створюємо новий Machine to Machine Application.

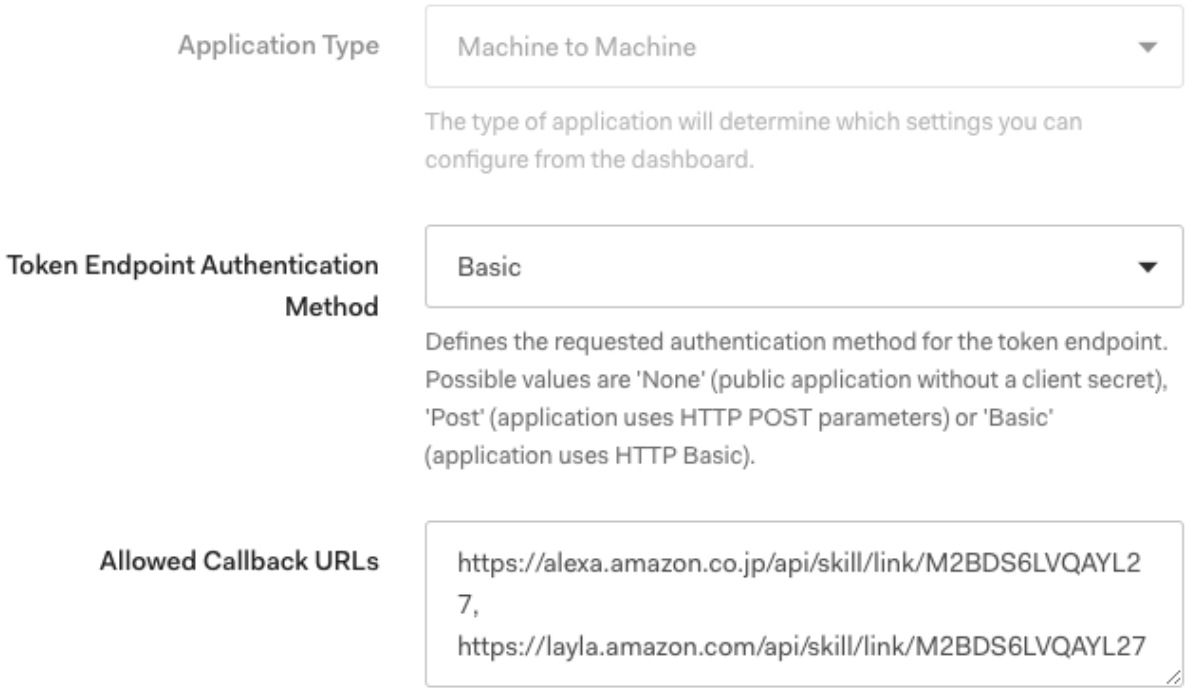

Рис. 4.9 Сайт Amazon Oauth Machine to Machine Application

Ну а на самому Skilla вже треба буде заповнити client id / secret, згенерований на Auth0. Решта поля доведеться заповнювати відповідно до вимог Auth0 і ім'ям вашого домену.

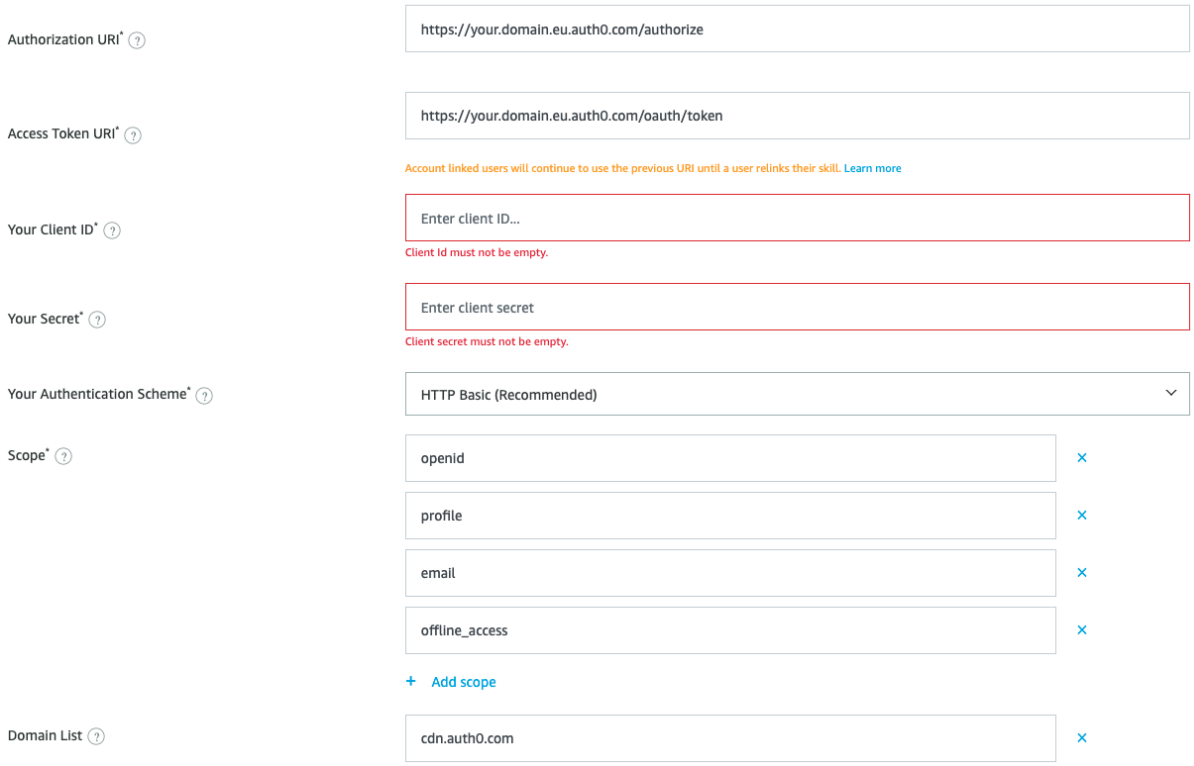

Рис. 4.10 Сайт Skill з налаштуванням зв'язку із Oauth

4.5. AWS Lambda

Як back-end для обробки запитів в Alexa© Skill нам знадобиться наша функція. Для її створення вам доведеться скористатися вже основним AWS-аккаунтом.

Рис. 4.11 Створення функції в AWS Lambda

Тут, задавши ім'я, все інше можна залишити за замовчуванням, якщо тільки вам не потрібен інший runtime. Ми будемо використовувати приклад лише на базі JS / TypeScript (можна вибрати іншу мову, але тоді доведеться адаптувати код).

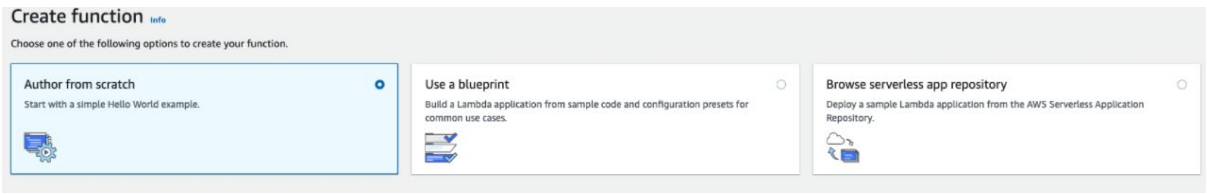

Рис. 4.12 Налаштування функції в AWS Lambda

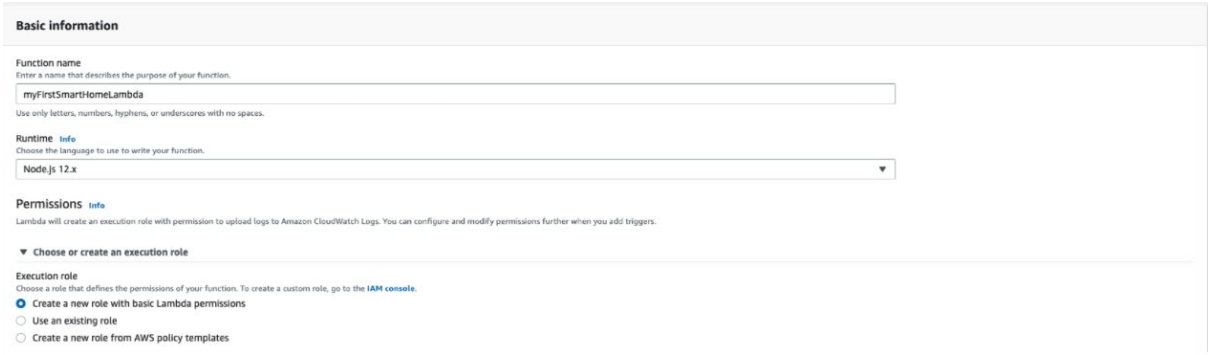

Рис. 4.13 Налаштування функції в AWS Lambda

Далі на головній сторінці нам потрібно додати Smart Home trigger. При цьому вам відразу доведеться вказати ID вашого Skilla, який можна скопіювати на головній сторінці:

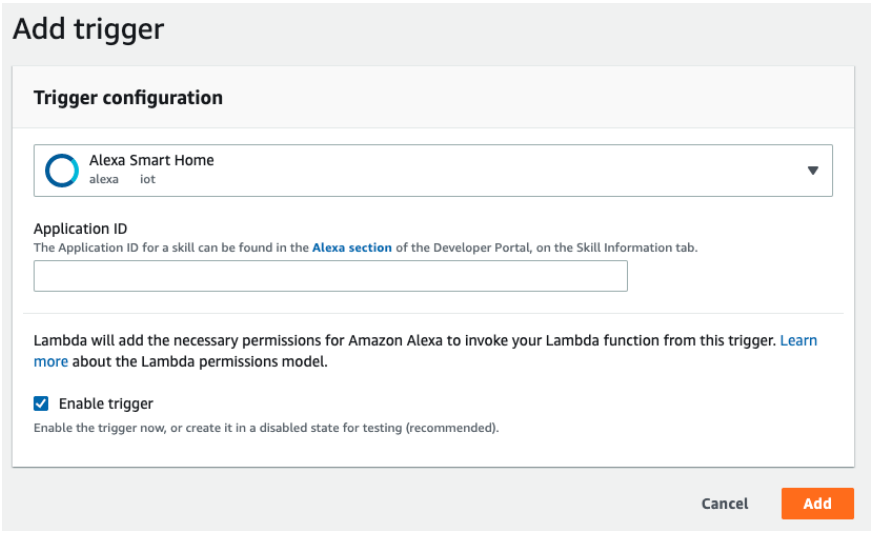

Рис. 4.14 Створення Smart Home trigger

Для того щоб тепер зв'язати функцію з нашим Skillom, необхідно скопіювати ARN:

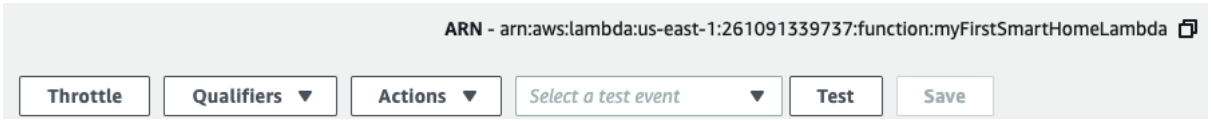

Рис. 4.15 Створення Smart Home trigger

Потім ARN треба вставити в поле Default Endpoint на головній, тобто в результаті у нас утворюється жорстка двосторонній зв'язок між Skilla і Lambda.

Отже, у нас все готово для розробки back-end.

4.6. Alexa© Smart Home Skill API

Нашому back-end (Lambda) знадобиться реалізувати обробники запитів, що приходять зі Skilla. Основна проблема тут в тому, що готового SDK під smart home skills поки ще не існує, тому багато речей доведеться описувати для своїх потреб.

Стосовно проблеми AWS Lambda. При можливості треба зменшувати використання його, тому що це мало не основна частина, де використовується значна частина процесорного часу для обробки (bottleneck), в контексті smart home skills. Це призводить до того, при довгому часі відсутності запитів до нього, пристрій встає в режим сну. Затримка першого запиту після режиму сну доволі велика, але всі наступні запити будуть проходити досить швидко.

Стосовно API: Amazon намагається зробити його гранулярним. Це, в деякому сенсі, дає певну гнучкість, але при великій кількості багатофункціональних пристроїв і через відсутність SDK сильно уповільнює процес реалізації.

Для quick start з лампочкою вам знадобиться реалізувати кілька ключових оброблювачів:

- Authorization — спрацьовує при активації користувальницького аккаунта (account linking).
- Discovery — викликається для запиту пристроїв, що знаходяться в локальній підмережі з колонкою (тут важливо швидко повернути список доступних, навіть віртуальних).
- ReportState — спрацьовує в разі необхідності запросити стан конкретного пристрою (актуально для мобільного додатка).
- PowerController — обробник команд включення / вимикання (спрацьовує за ключовими словами — on / off).

Коли ми просимо Alexa© знайти пристрої, в хмарі формується директива, яка містить різноманітну інформацію. Але для нас найбільш актуальними є Interface і Header Name. Ці два поля дають нам можливість однозначно

ідентифікувати обробник, який необхідно викликати у відповідь на який прилетів запит. У цьому конкретному прикладі ми зобов'язані повернути детальний список пристроїв (Endpoints). При цьому в числі endpoint-характеристик присутні два важливих поля: Endpoint ID (унікальний ідентифікатор пристрою) і Friendly Name (ім'я, на яке пристрій буде відгукуватися).

Коли ми вимовляємо «Alexa, light on», в хмарі відбувається пошук відповідності озвученого friendly name — light одному з ID раніше знайдених пристроїв (endpoint id). Таким чином, в директиві у нас прилітає вже ідентифікатор пристрою, а не його ім'я. При цьому інтерфейс визначається на підставі самої команди on, яка, згідно з документацією, прив'язана до PowerController. Header же спирається на саме значення: у випадку з on — це TurnOn, а для off було б TurnOff.

Слід також зазначити, що реалізація так званих контролерів безпосередньо залежить від тих властивостей, що ми повернули на фазі Discovery для кожного з пристроїв. Наприклад, світлодіодну стрічку ми можемо як вмикати / вимикати, так і змінювати її яскравість. А це означає, що в процесі пошуку такої стрічки ми повинні позначити підтримку PowerController і BrightnessController. Таким чином, коли користувач захоче включити стрічку, спрацює перший, змінити яскравість — другий.

ВИСНОВОК

Було проведено ознайомлення з сучасними системами голосового контролю построїв та запропонована ідея взаємодії з ними для знаходження дефекту мікрофона в них за допомогою детектора голосової активності і проведений аналіз наявної інформації .

На основі цього було зроблено висновок, що технології з використанням голосових помічників доволі розповсюджена і продовжує поширюватися, так як це дозволяє користувачам зручно взаємодіяти з системами без використанням прямого вводу даних. Поглиблення в проблематику цієї проблеми потребує подальшого вивчення систем голосового контролю та систем розумного дому.

Для дослідження були вивчені існуючі кодери, які використовуються в мобільному зв’язку, їх властивості, методи обробки, принципи роботи і недоліки. Для відтворення роботи детектора використовувався програмний пакет MATLAB® та наведені приклади використання деяких функцій для окремого огляду.

Була представлена робота із програмним доповненням для системи розумного будинку Amazon™ Dot, як приклад використання програмного забезпечення для роботи із одним елементом масиву мікрофонів.

ЛІТЕРАТУРА

1. Alto University Wiki. Voice activity detection (VAD). – 03.21.2021. — https://wiki.aalto.fi/pages/viewpage.action?pageId=151500905#space-menu-link-content
2. Летов, Игорь. Речевые кодеки. – 21.03.2021 г. — http://celnet.ru/vocod.php
3. ITU-T (1996) A silence compression scheme for G.729 optimised for terminals conforming to ITU-T V.70, ITU-T Rec. G.729 Annex B.
4. ITU-T (1996) Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s. Annex A: Silence compression scheme, ITU-T Rec. G.723.1 Annex A.
5. ITU-T (1996) Coding of speech at 8 kbit/s using conjugate structure algebraiccode excited linear prediction (CSACELP), ITU-T Rec. G.729.
6. ITU-T (1996) Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s, ITU-T Rec. G.723.1.
7. ETSI (1998) Digital cellular telecommunications systems (phase 2+); Voice activity detector (VAD) for full rate speech traffic channels, GSM 06.32 (ETSI EN 300 965 v7.0.1).
8. ETSI (1999) Digital cellular telecommunications systems (phase 2+); Voice activity detector (VAD) for full rate speech traffic channels, GSM 06.42 (draft ETSI EN 300 973 v8.0.0).
9. ETSI (1997) Digital cellular telecommunications systems; Voice activity detector (VAD) for enhanced full rate (EFR) speech traffic channels, GSM 06.82 (ETS 300 730), March.
10. ETSI (1998) Digital cellular telecommunications systems (phase 2+); Voice activity detector (VAD) for adaptive multi-rate (AMR) speech traffic channels, GSM 06.94 v7.1.1 (ETSI EN 301 708).
11. DeJaco P., Gardner W., and C. Lee (1993) ‘QCELP: The North American CDMA digital cellular variable rate speech coding standard’, in IEEE Workshop on Speech Coding for Telecom, pp. 5–6.

12. TIA/EIA (1997) Enhanced variable rate codec, speech service option 3 for wideband spread spectrum digital systems, IS-127.
13. TIA/EIA (1998) High rate speech service option 17 for wideband spread spectrum communication systems, IS-733.
14. M.Y. Appiah, M. Sasikath, R. Makrickaite, M. Gusaite, «Robust Voice Activity Detection and Noise Reduction Mechanism, Institute of Electronics Systems, Aalborg University.
15. X.L. Liu, Y. Liang, Y.H. Lou, H. Li, B.S. Shan, Noise-Robust Voice Activity Detector Based on Hidden Semi-Markov Models, Proc. ICPR'10, 81-84.
16. Kondoz A.M. Digital Speech. Coding for Low Bit Rate Communication Systems. – John Wiley & Sons, Ltd. 2004. – 442 p.
17. AIDC INTEL AI DEVCON. Designing Far-Field Speech Processing System with Intel and Alexa Voice Service. Amazon Dot build –in scheme with microphones. – 03.25.2021. — https://aidc.gallery.video/detail/video/5790045676001/designing-far-field-speech-processing-systems-with-intel-and-alexa-voice-service?autoStart=true&q=amazon
18. **The MathWorks, Inc.** Voice Activity Detection in Noise Using Deep Learning // MathWorks. - 04 12, 2021. - https://www.mathworks.com/help/audio/ug/voice-activity-detection-in-noise-using-deep- learning.html.
19. **The MathWorks, Inc.** detectSpeech // MathWorks. - 04 15, 2021. - https://www.mathworks.com/help/audio/ref/detectspeech.html.