

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

# **КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

**на тему:**

**«Інформаційна технологія машинного навчання  
системи керування протезом кісті руки з  
неінвазивною системою зчитування біосигналів»**

**Завідувач**

**випускаючої кафедри**

**Керівник роботи**

**Студента групи ІНм – 91н**

**Довбиш А.С.**

**Довбиш А.С.**

**Супруненко М.К.**

**СУМИ 2021**

(назва вузу)

Факультет ЕЛІП Кафедра Комп'ютерних наук

Спеціальність «Інформатика»

Затверджую:

зав.кафедрою \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

### ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Супруненко Микиті Костянтиновичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна технологія машинного навчання системи керування протезом кiстi руки з неінвазивною системою зчитування біосигналів.

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи) 26.01.2021 р.

3. Вхідні данні до проекту (роботи)

Вибірка міографічних даних , зібрана дослідниками з Познаньського університету

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми дослідження

2) Опис методів досліджень

3) Інформаційне, алгоритмічне та програмне забезпечення системи розпізнавання електронограм.

4) Додаток А. Лістинг програми.



## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	<i>Аналіз проблеми. Постановка задачі дослідження</i>		
2.	<i>Об'єктивна типологія психологічних станів людини</i>		
3.	<i>Інформаційно-екстремальна інтелектуальна технологія</i>		
4.	<i>Розробка інформаційного та програмного забезпечення системи комп'ютерної психодіагностики</i>		
5.	<i>Оформлення пояснювальної записки до дипломної роботи</i>		

Студент – дипломник

\_\_\_\_\_  
(підпис)

Керівник проекту

\_\_\_\_\_  
(підпис)

# РЕФЕРАТ

**Записка:** 74 стор., 19 рис., 3 табл., 1 додаток, 36 джерел.

Об'єктом дослідження є процес розпізнавання системою керування кінцівки протезу біосигналів з неінвазивною системою їх зчитування.

Предметом дослідження є моделі та методи машинного навчання системи керування протезом кінцівки руки з неінвазивною системою зчитування біосигналів.

В магістерській кваліфікаційній роботі розглядається задача розпізнавання електроміографічних біосигналів м'язів передплеччя руки для керування протезом кінцівок руки. Інформаційним синтез системи керування протезом кінцівок руки з неінвазивною системою зчитування біосигналів здійснюється шляхом інформаційно-екстремального машинного навчання, основою якого є максимізація інформаційної спроможності системи в процесі її навчання. Як параметри оптимізації розглядаються контрольні допуски на ознаки розпізнавання, а як критерій оптимізації параметрів машинного навчання – модифікована інформаційна міра Кульбака. Програмна реалізація алгоритму машинного навчання виконана на мові Python.

ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА  
ТЕХНОЛОГІЯ, МАШИННЕ НАВЧАННЯ, КАТЕГОРІЙНА  
МОДЕЛЬ, ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, СИСТЕМА  
КЕРУВАННЯ, ПРОТЕЗ КІСТІ РУКИ,  
ЕЛЕКТРОМІОГРАФІЧНИЙ ДАТЧИК

# ЗМІСТ

ВСТУП.....	3
1. АНАЛІЗ ПРОБЛЕМИ .....	4
1.1. Сучасний стан і тенденція розвитку інтелектуальних протезів руки.....	<b>Ошибка! Закладка не определена.</b>
1.2. Методи машинного навчання та розпізнавання сигналів.....	12
1.3. Формалізована постановка задачі дослідження .....	<b>Ошибка! Закладка не определена.</b>
2. ОПИС МЕТОДУ ДОСЛІДЖЕНЬ .....	<b>Ошибка! Закладка не определена.</b>
2.1. Основні положення інформаційно-екстремальної інтелектуальної технології класифікації .....	<b>Ошибка! Закладка не определена.</b>
2.2. Оцінка функціональної ефективності системи керування протезом руки, що навчається .....	<b>Ошибка! Закладка не определена.</b>
2.3. Алгоритми інформаційно-екстремального машинного навчання .....	<b>Ошибка! Закладка не определена.</b>
3. ІНФОРМАЦІЙНИЙ СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ПРОТЕЗОМ КІНЦІВКИ РУКИ.....	28
3.1. Формування вхідного математичного опису системи керування протезом руки з неінвазивною системою зчитування біосигналів .....	28
3.2. Категорійна модель машинного навчання системи керування протезом кінцівки руки .....	34
3.3. Короткий опис програмного забезпечення .....	42
3.4. Результати фізичного моделювання .....	46
ВИСНОВОК .....	<b>Ошибка! Закладка не определена.</b>
СПИСОК ЛІТЕРАТУРИ.....	<b>Ошибка! Закладка не определена.</b>

## ВСТУП

Незважаючи на значну кількість досліджень із вдосконалення взаємодії особи з інвалідністю з протезами, їх зручність, функціональність та поширеність активних протезів в побуті залишається на невисокому рівні. Основні напрямки вдосконалення - це швидкість, надійність управління при збільшенні можливих рухів. Для користувача важливі такі аспекти керованості: точність вибору руху, інтуїтивність керування та час реакції системи. Найбільший потенціал розвитку та природність забезпечує керування на базі міографії, як методу аналізу природних керуючих імпульсів нервової системи.

Об'єктом дослідження є процес розпізнавання системою керування кінцівки протезу біосигналів з неінвазивною системою їх зчитування.

Предметом дослідження є моделі та методи машинного навчання системи керування протезом кінцівки руки з неінвазивною системою зчитування біосигналів.

В попередніх працях з аналізу міографічних даних в рамках інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології) аналізу даних функціональна ефективність машинного навчання не досягала свого граничного максимального значення через зашумленість біосигналів та довільні початкові умови формування вхідного математичного опису. Крім того, попередні дослідження проводилися при малій потужності алфавіту класів розпізнавання, що обмежувало функціональні можливості системи керування протезом.

В магістерській роботі розроблено категорійні моделі і методи інформаційно-екстремального машинного навчання системи керування протезом кінцівки руки за ієрархічною структурою даних і оптимізацією параметрів формування вхідного математичного опису системи.

### 3 АНАЛІЗ ПРОБЛЕМИ

#### 1.1 Сучасний стан і тенденція розвитку інтелектуальних протезів

Сучасний розвиток протезів руки відбувається у напрямку підвищення їх функціональної ефективності шляхом використання методів аналізу даних [1–4], і тому сильно пов'язаний із подоланням зашумленості. Існуючі біонічні протези, керовані сигналами від пасивних електроміографічних сенсорів мають не достатньо високу точність через довільні фізичні умови зчитування даних для формування керуючих команд.

Мінімізувати несприятливі біофізичні чинники зашумлення на структурному рівні дозволяють інвазивні системи зчитування сигналів нервової системи [5], однак вони вимагають хірургічного втручання та мають високу вартість. Іншим способом подолання зашумленості є додавання до контуру керування, поряд із нервовою системою людини, більш детермінованого джерела даних для формування команд. Такий спосіб втілюють приклади протезів [6–8], що наділені тактильною функцією та здатні розпізнавати поверхню предмету, за рахунок даних від датчиків із відомими та сталими параметрами функціонування, а також системи з використанням оптичного трекінгу руху очей [9,10]. Більш перспективними є тактильні системи, бо вони не потребують додаткових пристроїв зчитування на інших частинах тіла окрім протезованої кінцівки, та мають потенціал для організації зворотнього зв'язку для нервової системи і, таким чином, повного сенсорного кола із протезом.

Розвиток протезобудування умовно можна поділити на три покоління:

Перше покоління - пасивних протезів, що відомі з XIV століття і не зазнавали суттєвих змін до другої половини XX століття. Під час цього періоду їх розвиток полягав у зменшенні ваги, збільшенні зносостійкості та кращому імітуванні зовнішнього вигляду їх живих прототипів. Створені протези були



механічними, як максимум керованими за допомогою безпосередньо м'язової сили, несли частіше за все косметичну функцію, були мало функціональними та мали низьку надійність через знос механічних складових.

Далі, із розвитком електроніки, з'являються перші активні протези, частіше за все напівмеханічні. Система керування таких протезів базується на схемі з невеликої кількості датчиків-елементів керування, що забезпечують зв'язок із користувачем, та електронних логічних елементів і відтворює декілька основних рухів. Частина рухів, власне, самого протезу, зазвичай була реалізована механічно. Функціональність таких протезів вже вища за пасивні, але все одно занадто низька, навіть для комфорту в побуті, а механічні частини негативно впливають на надійність.

Третє покоління пов'язано з розвитком інтегральних мікросхем та технологій їх масового виробництва, що призвело до появи відносно дешевих програмованих мікропроцесорів, з достатньо низьким енергоспоживанням для використання в автономних пристроях. Це, в свою чергу, призвело до певної уніфікації апаратної частини систем керування активного протезу. Нині така система містить датчики, що збирають керуючі сигнали з тіла оператора, АЦП, мікропроцесор, що виконує програму розпізнавання сигналів та драйвер електродвигунів. Акцент вдосконалення при цьому перемістився в програмну галузь, а головною метою стало вдосконалення алгоритмів розпізнавання керуючих сигналів від користувача. [11, 12].

Наведений нижче патент [12] є типовим зразком третього покоління.

Його спосіб керування відбувається наступним чином: отримується ЕМГ сигнал пацієнта за допомогою міоелектричного пристрою зчитування; виконується обробка ЕМГ-сигналу пацієнта та формується набір ознак ЕМГ-сигналу; передається набір ознак в систему управління інтелектуальною біонічною кінцівкою; визначається тип жесту; формується керуючий сигнал та передається на двигуни, що приводять у рух пальці біонічної кінцівки; отримується зворотний зв'язок від системи управління інтелектуальною біонічною кінцівкою за допомогою отримання інформації від зовнішніх датчиків.

Технічна проблема, розв'язувана в даному технічному рішенні, складається в здійсненні автономного повнофункціонального управління для виконання низькорівневих завдань маніпуляції інтелектуальним біонічним протезом кінцівки, яке дозволило б сприймати зазначений протез як природну кінцівку.

Ще одним додатковим технічним результатом є спрощення процедури встановлення і налаштування системи «протез-пристрій, що зчитує» за рахунок модульності системи - пристрій, що зчитує є портативним, і вимагає приблизного позиціонування при розміщенні, і легко знімається. Протез пов'язаний із зчитуючим пристроєм тільки бездротовим каналом передачі даних.

Зниження когнітивного навантаження на людину і підвищення ефективності керування протезом досягається за рахунок використання гібридної системи управління, яка поєднує в собі способи декодування електронейроміосигналів з елементами автономних роботизованих маніпуляцій із захоплення предмета при досягненні порогового значення відстані до нього за допомогою прийняття рішення, без участі користувача, про початок або припинення виконання захвату за допомогою інформації від додаткових датчиків (температури і відстані), і про силу стиснення предмета за допомогою інформації від датчиків.

Точність класифікації та запобігання помилкового спрацьовування протеза досягаються за рахунок одночасного формування максимального значення і середньої тривалості перевищення порогового значення нейроміосигнала у вікні даних в режимі реального часу, потім після центрування і нормалізації приймається рішення про виникнення керуючого сигналу.

Розташування мікроконтролера, виконаного з можливістю модульної збірки в протезі, дозволяє наблизити штучну кінцівку за вагою та розміром до природної. Використання зворотного зв'язку, що реалізується за допомогою датчиків температури, відстані, моменту і позиції, розташованих в протезі, дозволяє забезпечити використання протеза і надати максимальні можливості по управлінню.

На рис. 1.1 показана загальна структура біонічного протезу кінцівки руки третього покоління [12].

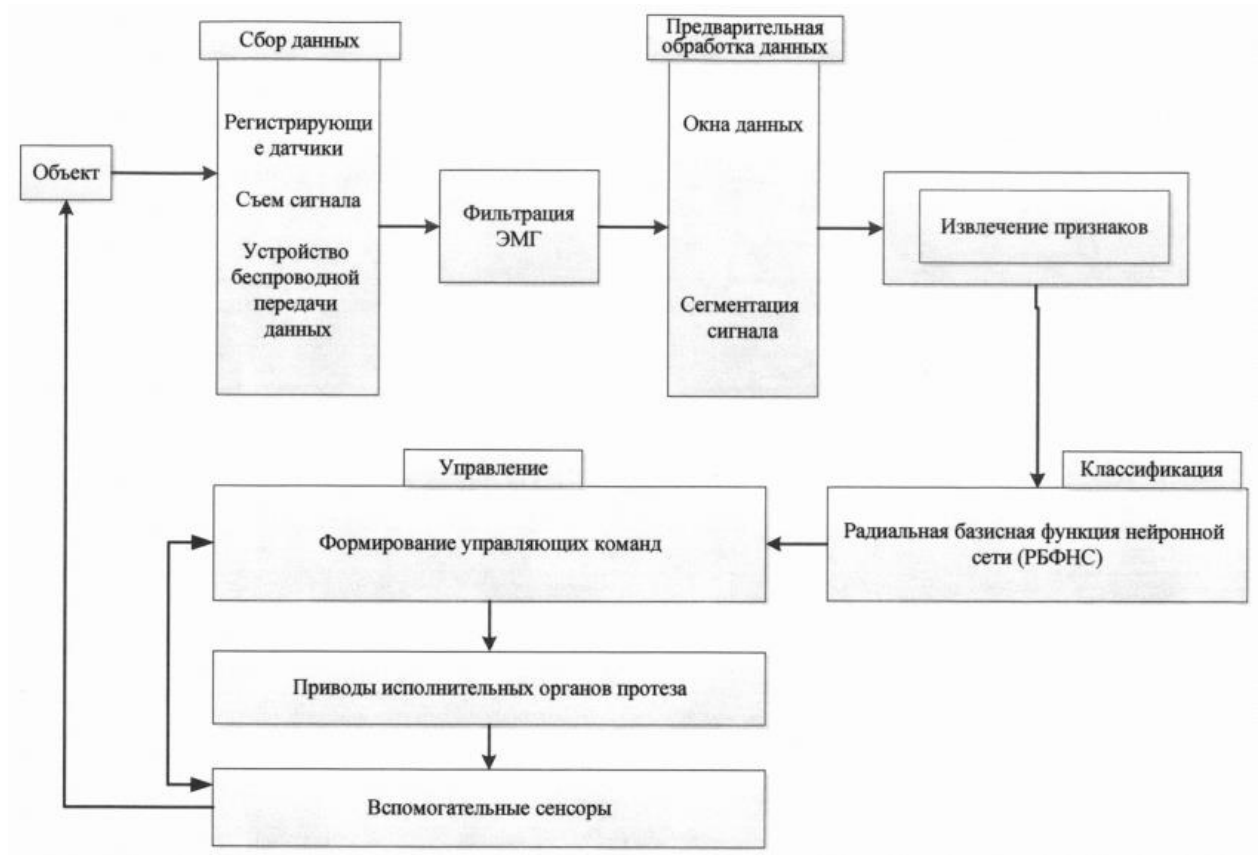


Рисунок 1.1 – Загальна структура біонічного протезу

На рисунку 1.1 розпізнавання біосигналів здійснюється штучною нейронною мережею Хеммінга, яка в процесі машинного навчання відновлює в радіальному базисі роздільну гіперповерхню. Система керування біонічного протезу через додаткові тактильні сенсори має зворотний зв'язок, який дозволяє корегувати керуючі команди з метою досягнення необхідної їх точності та оперативності.

На рисунку 1.2 показано структурну схему системи керування протезом кінцівки руки з неінвазивною системою зчитування біосигналів [12]/



Рисунок 1.2 – Блок-схема системы керування інтелектуальною біонічною кінцівкою руки

На рисунку 1.3 показано електроміографічну систему зчитування біосигналів [11].

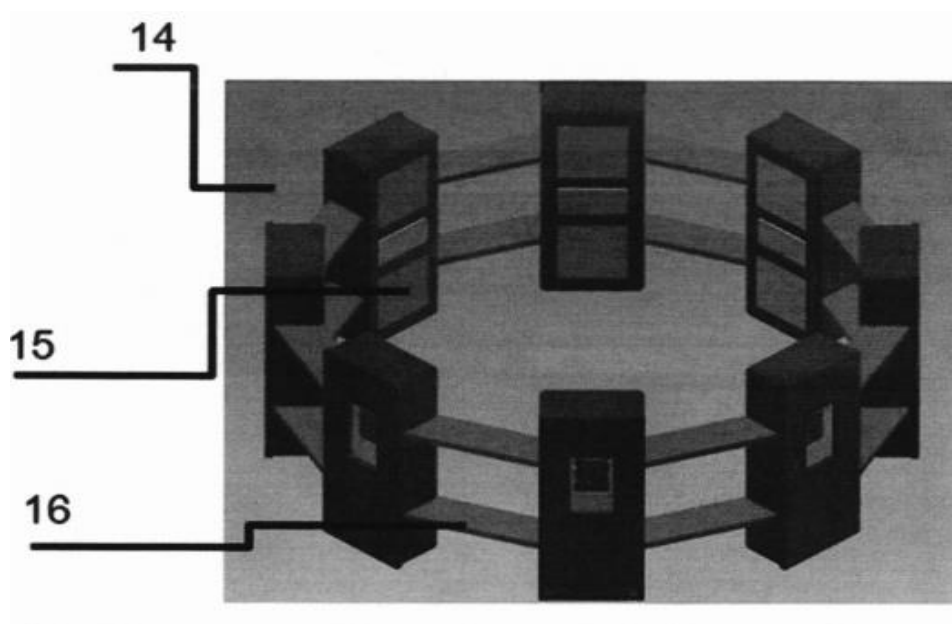


Рисунок 1.3 – Електроміографічна система зчитування біосигналів: 14 – система зчитування; 15 – датчик реєстрації біопотенціалів; 16 – еластична манжета

На рисунку 1.4 показано схему електроміографічного датчика [11], а на рисунку 1.5 – розміщення на протезі датчиків зворотного зв'язку.

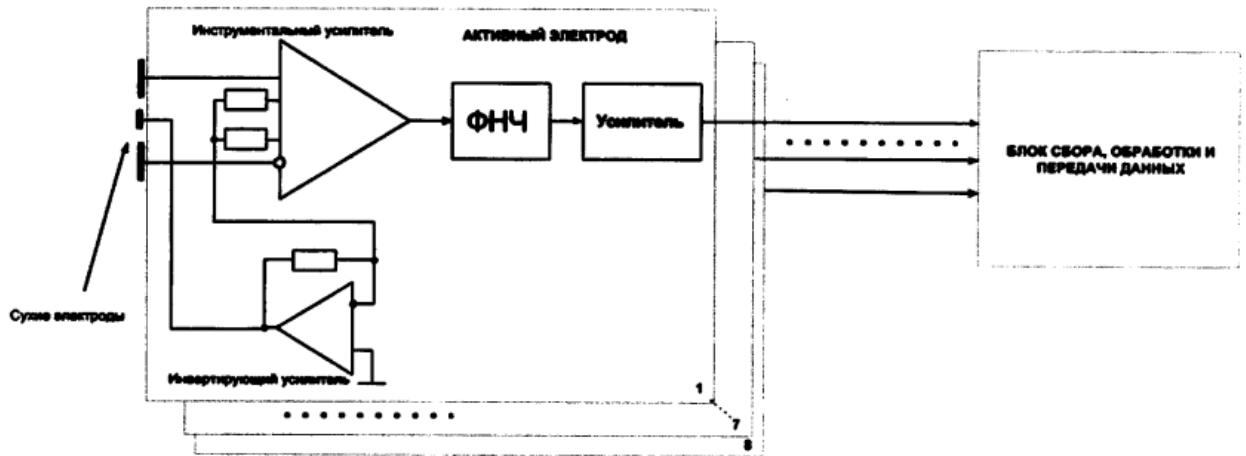


Рисунок 1.4 – Схема реалізації ЕМГ-датчика з сухим контактом.

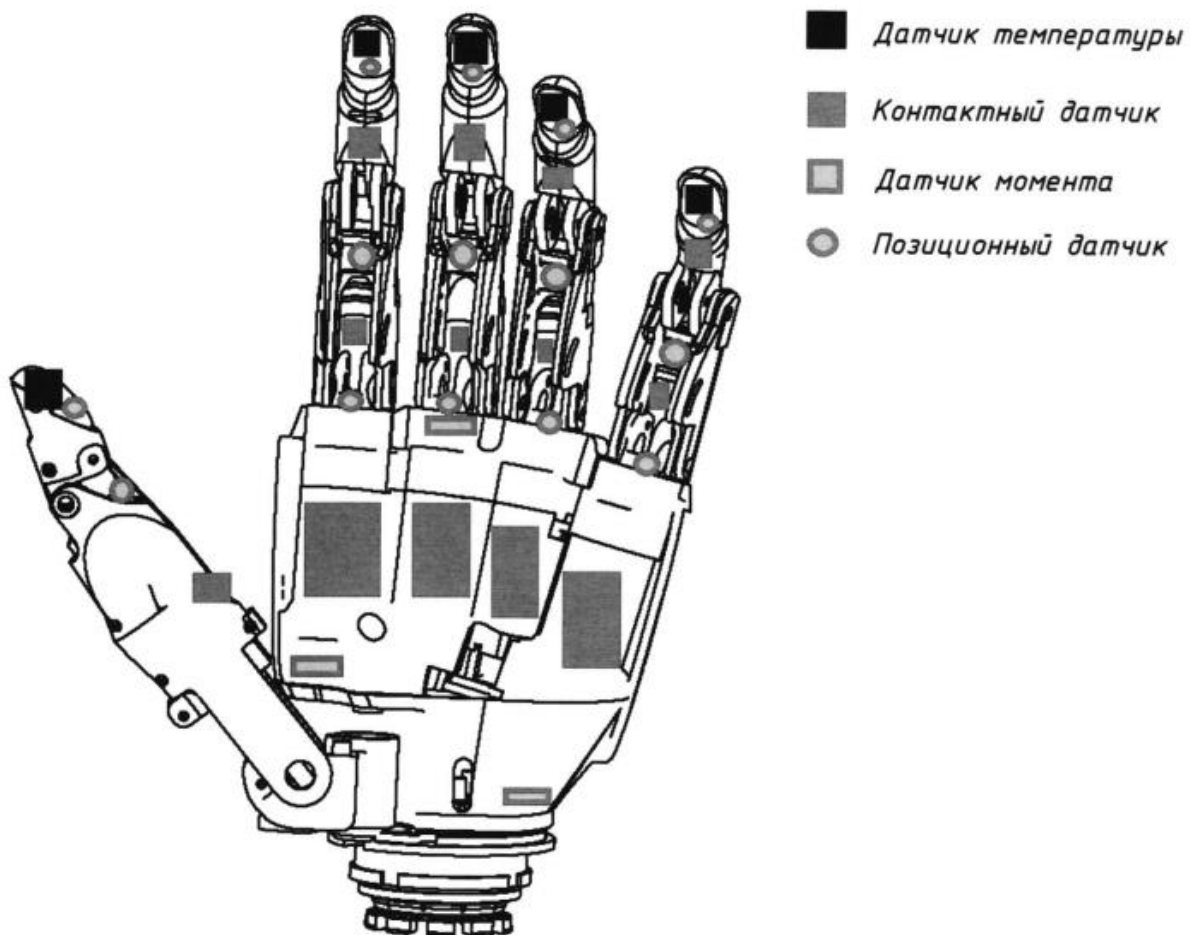


Рисунок 1.5 – Приклад розміщення датчиків зворотного зв'язку

На рис. 1.6 показано схеми приводу пальця протезу в прямому і зігнутому положеннях [11].

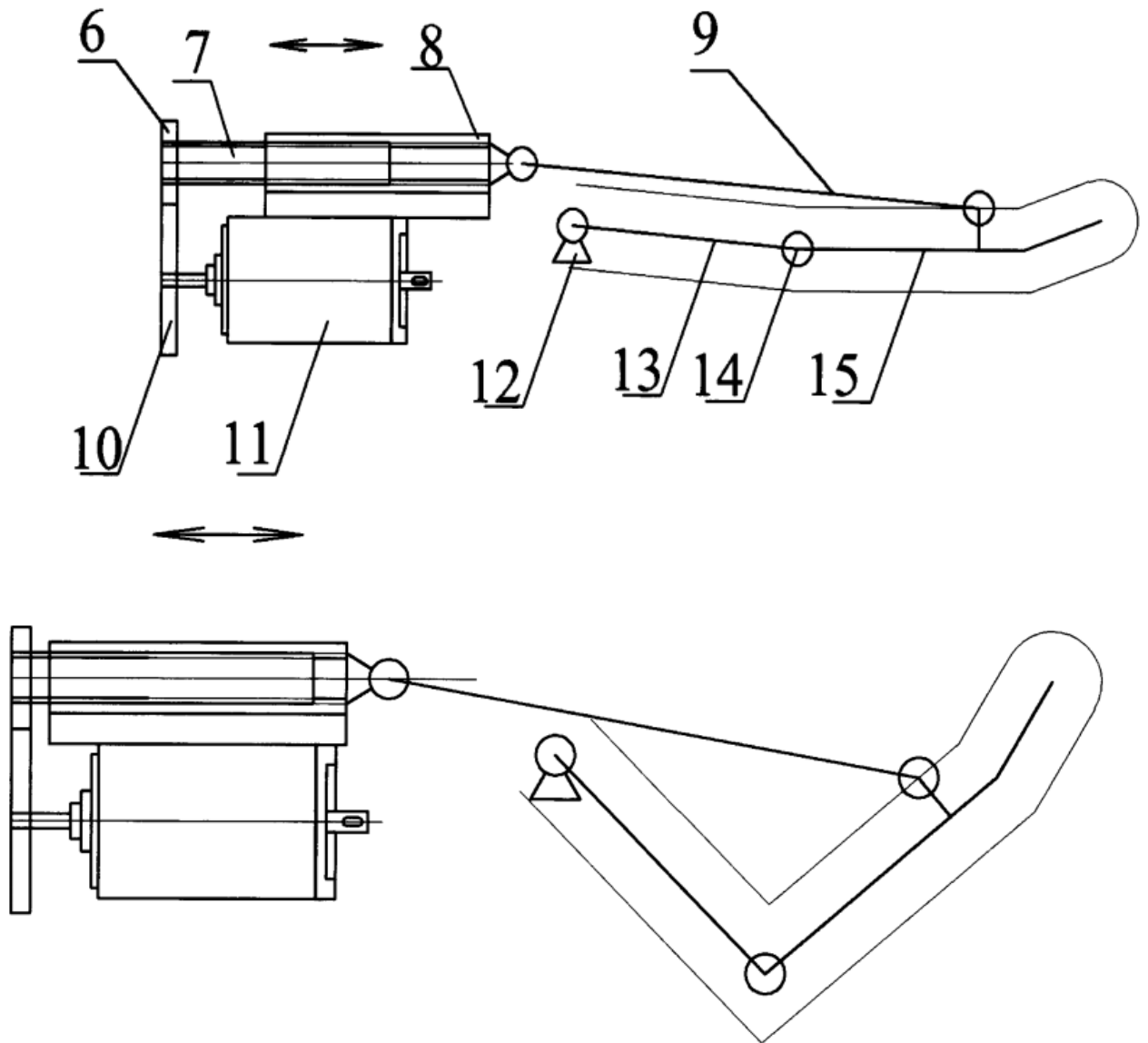


Рисунок 1.6 – Механізм приводу пальця в прямому (а) і зігнутому (б) стані;  
 6 – зубчате колесо; 7 – різбовий плунжер; 8 – гайка; 9 – важіль-привід;  
 10 – зубчате колесо велике; 11 – мікродвигун-редуктор; 12 – суглоб-вузол  
 повороту закріплений на п'ясті; 13 – перша фаланга пальця; 14 – суглоб-  
 вузол повороту пальця; 15 – друга фаланга пальця

На рис.1.7 показано структурну схему алгоритму обробки і розпізнавання біосигналу від системи зчитування [11].

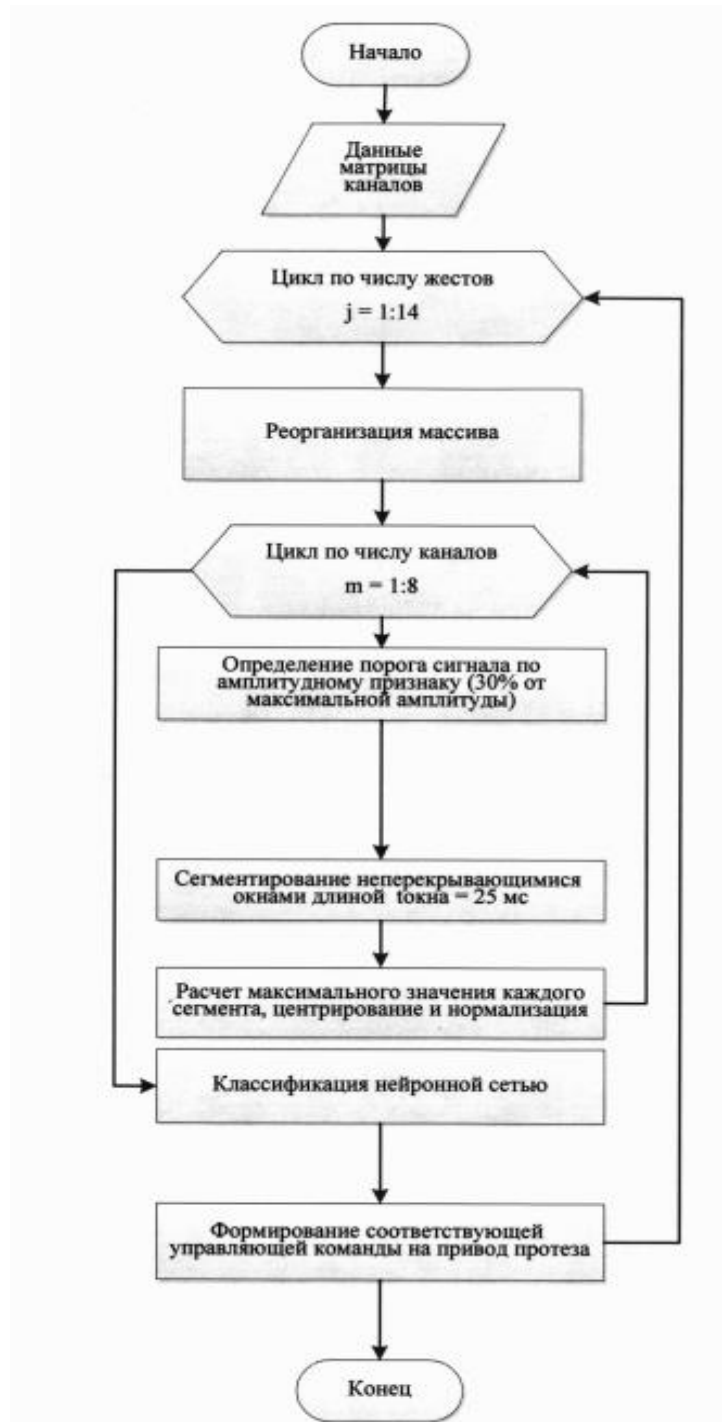


Рисунок 1.7 – Структурна схема алгоритму обробки біосигналу від системи зчитування

Схема протезу, показана на рисунку 1.2 обрана як аналог для системи керування протезом кiстi руки з неінвазивною системою зчитування біосигналів, що розробляється в даній роботі

## Методи машинного навчання та розпізнавання сигналів

Алгоритми машинного навчання для встановлення відображення між біосигналами та керуючими командами на основі нейронних мереж [12,13] та опорних векторів [14–17] через необхідну велику потужність словника ознак і суттєвий перетин фактор множин класів розпізнавання не дозволяють досягти достатньо високої точності визначення команд за біосигналами. Через це, розпізнавання рухів, наприклад, окремих пальців, навіть за умови відсутності пошкоджень м'язової тканини, має точність не вище 62% [16].

Оптимальні затримки контролера становлять 100 мс для швидких рухів і 125 мс для повільних, без погіршення показників протезування, тоді як затримки довше 200 мс негативно впливають на продуктивність користувача, та незважаючи на це, лабораторні дослідження розпізнавання образів зазвичай дають результати 300 мс. [18]

Експериментально було доведено [17], що в живих нейронних мережах працює навчання без нагляду в тому сенсі, що вага синапсу збільшується у відповідь на ко-активацію пре- та постсинаптичних нейронів [19]. Тому в статті [20] запропоновано використовувати штучну нейронну мережу Кохонена, яка реалізує SOM

Основною властивістю SOM є здатність обробляти дані на вхідному рівні, зберігаючи при цьому топологічні особливості простору реалізацій навчальної вибірки. Практичне значення цієї властивості передбачає можливість зменшення розмірності вхідних даних. Крім того, SOM може використовуватися як проміжний етап у системі класифікації даних [20]. Однак точність розпізнавання в досліді поступається навіть багат шаровому перцептроні і є недостатньою.

В статті [21] досліджується застосування для класифікації сигналів нечітких нейронних мереж, однак вони так само зазнають впливу багатовимірності. В роботах [22–24] для зменшення цього впливу використано екстрактори ознак з вхідних даних на основі штучних нейронних мереж.



Перспективним є використання методів інформаційно-екстремальної інтелектуальної технології класифікації даних, яка ґрунтується на максимізації інформаційної спроможності системи в процесі її навчання [1, 25–27].

Основною метою інформаційно-екстремального машинного навчання, є адаптація вхідного математичного опису системи до максимальної достовірності розпізнавання образів, як і в нейроподібних структурах. Але на відміну від них вирішальні правила будуються в рамках геометрично-топологічного підходу, що знімає проблему багатовимірності словника ознак.

В дослідженні [1] у рамках базового алгоритму ІЕІТ було розроблено СППР, що відзначилася адаптивністю та високою достовірністю розпізнавання поточного функціонального стану технологічного процесу.

В дослідженні [3] вдалося досягти прискорення обчислень алгоритму тонкого налаштування параметрів аналізу даних, завдяки використанню вирішальних правил побудованих на основі ІЕІТ. Це свідчить про високу узагальнюючу здатність та обчислювальну ефективність алгоритму.

В дослідженні [2] результатом машинного навчання системи керування протезом руки стали вирішальні правила, які дозволили в робочому режимі приймати в реальному темпі часу достовірні класифікаційні рішення. Це було досягнуто завдяки здійсненню паралельно-последовної оптимізації системи контрольних допусків на ознаки розпізнавання. Також було досліджено вплив контрольних допусків на функціональну ефективність навчання системи і встановлено, що інформаційний критерій оптимізації параметрів машинного навчання системи не досягає свого граничного максимального значення і синтезований класифікатор біосигналів не є безпомилковим за навчальною матрицею. Це означає що для підвищення функціональної ефективності доцільно впровадити оптимізацію додаткових параметрів функціонування системи керування.

У рамках інформаційно-екстремальної інтелектуальної технології розглядалася категорійна модель і алгоритм навчання системи підтримки прийняття рішень (СППР) для керування вирощуванням монокристалів із розплаву.

З метою підвищення достовірності та оперативності розпізнавання функціональних станів системи керування при збільшенні потужності алфавіту класів розпізнавання в процесі навчання СППР була застосована ієрархічна структура класів розпізнавання.

Алгоритм використовує обрану заздалегідь  $h$ -ярусну деревоподібну бінарну ієрархічну структуру класів розпізнавання. Береться усереднений за ярусами критерій функціональної ефективності (КФЕ) [4]

### 1.3 Формалізована постановка задачі дослідження

Інформаційний синтез здатної навчатися системи керування біонічною кінцівкою руки у рамках ІЕІ-технології визначається максимізацією інформаційної спроможності системи - найважливішої характеристики її функціональної ефективності, що визначає достовірність класифікаційних рішень. При цьому формалізована постановка задачі інформаційного синтезу системи керування, що навчається має наступну форму. Відомі алфавіт класів розпізнавання  $\{X_m^o / m = \overline{1, M}\}$ , які відповідають реалізованим рухам протезу, і багатовимірна навчальна матриця типу «клас-реалізація-властивість»  $\| \| y_{m,i}^{(j)} / i = \overline{1, N}; j = \overline{1, n} \|$ , де  $N$  – кількість ознак розпізнавання;  $n$  – обсяг вибірки. В матриці навчальної вибірки для певного класу рядки є його реалізаціями, які складаються з послідовності значень біосигналів, квантованих за часом, а стовпчик – множина довільних його реалізацій. Крім того, дано структурований вектор параметрів машинного навчання системи керування розпізнавати реалізації класу  $X_m^o$

$$g_m = \langle x_m, d_m, \delta_m, \theta \rangle, \quad (1.1)$$

де  $x_m$  – усереднений вектор-реалізація, вершина якого визначає геометричний центр гіперсферичного контейнеру класу  $X_m^o$ ;  $d_m$  – радіус контейнеру класу  $X_m^o$ ;  $\delta$  – параметр машинного навчання, який дорівнює половині симетричного поля контрольних допусків на ознаки розпізнавання.  $\theta$  – параметр квантування реалізацій з навчальної вибірки за рівнем сигналу.

На параметри навчання задано обмеження  $d_m < d(x_m \oplus x_c)$ , де  $d(x_m \oplus x_c)$  – кодова відстань між вектором  $x_m$  і усередненим вектором  $x_c$  сусіднього класу  $X_c^o$ , і  $\delta < \delta_H / 2$ , де  $\delta_H$  – нормоване поле допусків, яке задає область значень контрольного поля допусків на ознаки розпізнавання.

У процесі машинного навчання необхідно:

- визначити оптимальні значення координат вектору (1.1), які забезпечують максимум усередненого за алфавітом класів розпізнавання інформаційного критерію оптимізації параметрів навчання:

$$\bar{E} = \frac{1}{M} \sum_{m=1}^M E_m, \quad (1.2)$$

де  $E_m$  – максимальне значення інформаційного критерію оптимізації гіперпараметрів контейнеру класу  $X_m^o$  щодо відділення та покриття його реалізацій, обчислене в робочій (допустимій) області визначення його функції;

- за отриманими в процесі машинного навчання оптимальними (тут і далі в інформаційномірному розумінні) геометричними параметрами контейнерів класів розпізнавання побудувати вирішальні правила.

- в робочому режимі функціонування системи керування біонічної кінцівки, що відповідає циклічному повторенню етапу екзамену, необхідно реалізувати вирішальні правила на базі визначених на етапі машинного навчання

параметрів, класифікувати біосигнал, сформувати керуючий сигнал і подати його на виконавчий пристрій для реалізації відповідної функції.

## 1. 2 ОПИС МЕТОДУ ДОСЛІДЖЕНЬ

### 2.1 Основні положення інформаційно-екстремальної інтелектуальної технології класифікації

Сутність інформаційно-екстремального машинного навчання системи керування біонічним протезом руки полягає в адаптації вхідного математичного опису до максимальної достовірності класифікації біосигналів з метою формування відповідних керуючих команд для виконуючих пристроїв. При цьому трансформація вхідного апріорно нечіткого розподілу реалізацій образів в чітке здійснюється під час оптимізації за інформаційним критерієм гіперпараметрів контейнерів класів розпізнавання, які впливають на точнісні характеристики класифікаційних рішень. Оптимальні геометричні параметри контейнерів, одержані в процесі машинного навчання дозволяють на екзамені приймати рішення за відносно простими детермінованими вирішальними правилами, що важливо при реалізації алгоритмів прийняття рішень в реальному темпі часу.

Основним етапом формування вхідного математичного опису, здатної навчатися за ІЕІ-технологією, системи керування біонічною кінцівкою руки є створення тривимірної навчальної матриці  $\| y_{m,i}^{(j)} / m = \overline{1, M}; i = \overline{1, N}, j = \overline{1, n} \|$  типу «об'єкт-властивість». При цьому необхідно визначити:

- словник ознак і алфавіт класів розпізнавання;
- мінімальний обсяг репрезентативної навчальної матриці;
- нормовані допуски на ознаки розпізнавання.

Вхідний математичний опис матиме наступну теоретико-множинну структуру

$$\Delta_B = \langle G, T, \Omega, Z, Y, X; \Phi_1, \Phi_2 \rangle,$$

де  $G, T, \Omega, Z$  – відповідно простори вхідних сигналів (факторів), моментів часу одержання інформації; ознак розпізнавання; станів системи (визначає алфавіт класів розпізнавання);  $Y$  – структурована множина, відповідна вхідній багатовимірній навчальній матриці;  $X$  – бінарна навчальна матриця;  $\Phi_1 : G \times T \times \Omega \times Z \rightarrow Y$  – оператор формування матриці  $Y$ ;  $\Phi_2 : Y \rightarrow X$  – оператор трансформації вхідної навчальної матриці  $Y$  в бінарну матрицю  $X$ .

Згідно з принципом відкладених рішень побудова в процесі оптимізації параметрів машинного навчання безпомилкових за навчальною матрицею вирішальних правил в рамках ІЕІ-технології здійснюється за багатоциклічною ітераційною процедурою пошуку максимального граничного значення усередненого за алфавітом класів розпізнавання інформаційного критерію оптимізації параметрів машинного навчання:

$$\bar{E} = \max_{G_\xi \cap G_E} \frac{1}{M} \sum_{m=1}^M E_m, \quad (2.1)$$

де  $E_m$  – інформаційний критерій оптимізації параметрів навчання системи керування розпізнавати реалізації класу  $X_m^o$ ;  $G_\xi$  – допустима область значень  $\xi$ -ї ознаки розпізнавання;  $G_E$  – допустима область визначення функції інформаційного критерію.

На алгоритм навчання (2.1) накладаються наступні обмеження [4]:

$$\begin{aligned} & \left( \forall X_m^o \in \mathfrak{R}^{|M|} \right) \left[ X_m^o \neq \emptyset \right] \\ & \left( \exists X_m^o \in \mathfrak{R}^{|M|} \right) \left( \exists X_c^o \in \mathfrak{R}^{|M|} \right) \left[ X_m^o \neq X_c^o \rightarrow X_m^o \cap X_c^o \neq \emptyset \right] \\ & \left( \forall X_m^o \in \mathfrak{R}^{|M|} \right) \left( \forall X_c^o \in \mathfrak{R}^{|M|} \right) \left[ X_m^o \neq X_c^o \rightarrow \text{Ker} X_m^o \cap \text{Ker} X_c^o = \emptyset \right] \\ & X_m^o / m \neq c; m, c = \overline{1, M}, \end{aligned} \quad (2.2)$$

де  $\mathfrak{R}^{|M|}$  – в загальному випадку нечітке розбиття простору ознак на класи еквівалентності відповідні алфавіту станів;  $Ker X_m^o, Ker X_c^o$  – ядра класів  $X_m^o$  і  $X_c^o$ , які є найближчими сусідами;  $\Omega$  – простір ознак розпізнавання.

Таким чином, умови (2.2) допускають перетин класів розпізнавання, але не допускають перетину їх ядер, щоб не було поглинання одним класом розпізнавання найближчих своїх сусідів.

У бінарному просторі ознак  $\Omega$  формою оптимального контейнера класу розпізнавання є гіперпаралелепіпед. З метою узагальнення та зручності побудови такого контейнера припускається існування “псевдогіперсфери”, яка описує гіперпаралелепіпед, тобто містить усі його вершини. Це дозволяє далі розглядати такі параметри оптимізації контейнера в радіальному базисі простору ознак  $\Omega$ , як усереднений вектор  $x_m$  ознак класу розпізнавання  $X_m^o$ , вершина якого визначає геометричний центр контейнера, і радіус псевдосферичного контейнера, котрий визначається у просторі Хеммінга за формулою [2]

$$d_m = \sum_{i=1}^N (x_{m,i} \oplus \lambda_i) , \quad (2.3)$$

де  $x_{m,i}$  –  $i$ -та координата усередненого вектора  $x_m$ ;  $\lambda_i$  –  $i$ -та координата деякого вектора  $\lambda$ , вершина якого належить контейнеру класу розпізнавання  $X_m^o$ .

За ІЕІ-технологією відновлення в радіальному базисі простору ознак контейнера класу розпізнавання  $X_m^o$  здійснюється шляхом цілеспрямованого збільшення його радіусу  $d_m$  на кожному кроці машинного навчання.

Нехай класи  $X_m^o$  і  $X_c^o$  є найближчими сусідами, тобто мають серед усіх класів найменшу міжцентрову відстань  $d(x_m \oplus x_c)$ . Тоді за ІЕІ-технологією з

метою запобігання поглинання одним класом ядра іншого класу умови (2.2) доповнюються таким предикатним виразом:

$$\left(\forall X_k^o \in \tilde{\mathfrak{R}}^{|M|}\right)\left(\forall X_l^o \in \tilde{\mathfrak{R}}^{|M|}\right)\left[X_k^o \neq X_l^o \rightarrow (d_k^* < d(x_k \oplus x_l)) \& (d_l^* < d(x_k \oplus x_l))\right], \quad (2.4)$$

де  $d_k^*, d_l^*$  – оптимальні радіуси контейнерів класів розпізнавання  $X_k^o$  і  $X_l^o$  відповідно.

Алгоритм навчання за ІЕІ-технологією полягає в реалізації багатоциклічної ітераційної процедури оптимізації структурованих просторово-часових параметрів функціонування системи керування шляхом пошуку глобального максимуму усередненого за алфавітом  $\{X_m^o\}$  значення інформаційного критерію оптимізації параметрів машинного навчання. При цьому внутрішні цикли оптимізують фенотипні параметри навчання, які безпосередньо впливають на геометричну форму контейнерів класів розпізнавання та їх положення в бінарному просторі Хемінга. Приклад таких параметрів - це радіуси гіперсферичних контейнерів, та усереднені вектори реалізації класу розпізнавання, вершини яких визначають геометричні центри контейнерів. До генотипних відносять параметри навчання, які прямо впливають на розподіл реалізацій класу (наприклад, контрольні допуски на ознаки розпізнавання, рівні селекції координат еталонних двійкових векторів, параметри оптимізації словника ознак, плану навчання, параметри впливу середовища тощо).

Оптимізація кожного із цих параметрів дозволяє збільшувати максимальне значення інформаційного критерію оптимізації параметрів навчання, що підвищує повну ймовірність правильного прийняття рішень на екзамені.

Обов'язковою процедурою алгоритму навчання за ІЕІ-технологією є оптимізація контрольних допусків, величина яких безпосередньо впливає на значення відповідних ознак розпізнавання, а так само і на параметри розподілу реалізацій образу.



Детерміновані вирішальні правила будуються на основі одержаних в процесі інформаційно-екстремального навчання оптимальних, в інформаційно-номірному розумінні, геометричних параметрів контейнерів класів розпізнавання.

Одже, інформаційно-екстремальне машинне навчання може розглядатися як процес дефазіфікації вхідного нечіткого подання даних. Використання на екзамені побудованих в рамках геометричного підходу вирішальних правил дозволяє приймати високо достовірні рішення в реальному темпі часу. При цьому повна достовірність класифікатора наближається до максимальної асимптотичної, яка визначається функціональною ефективністю машинного навчання.

## 2.2 Оцінка функціональної ефективності системи керування протезом руки, що навчається

Центральним питанням інформаційного синтезу системи керування рухомим об'єктом є оцінка функціональної ефективності процесу машинного навчання, яка визначає максимальну достовірність рішень, що приймаються на екзамені. В ІЕІ-технології, як критерії оптимальності параметрів машинного навчання, можуть використовуватися різні функції, які відповідають властивостям інформаційних мір [1]

- інформаційна міра є дійсна і знакододатна функція від імовірності;
- кількість інформації для детермінованих подій ( $p_i = 1$  або  $p_i = 0$ ) дорівнює нулю;
- інформаційна міра має екстремум за значення ймовірності  $p_i = \frac{1}{m}$ , де  $m$
- кількість якісних ознак розпізнавання;
- сумісна інформаційна міра двох незалежних повідомлень дорівнює сумі їх відповідних інформаційних мір.

Найчастіше використовуються ентропійний критерій Шеннона та інформаційна міра Кульбака [1].

Модифікована інформаційна міра Кульбака, що запропонована в праці [7] розглядається як добуток логарифмічного відношення правдоподібності обчисленої на  $k$ -му кроці машинного навчання повної ймовірності правильного прийняття рішень  $P_{t,m}^{(k)}$  до повної ймовірності помилкового прийняття рішень  $P_{f,m}^{(k)}$  на їх різницю. Далі приведений модифікований критерій Кульбака

$$E_{K,m}^{(k)} = \log_2 \frac{P_{t,m}^{(k)}}{P_{f,m}^{(k)}} * [P_{t,m}^{(k)} - P_{f,m}^{(k)}] =$$

$$\left[ (D_{1,m}^{(k)}(d) + D_{2,m}^{(k)}(d)) - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)) \right] =$$

$$\log_2 \left( \frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * [1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))] \quad (2.5)$$

де  $\alpha_m^{(k)}(d)$  – помилка першого роду прийняття рішення на  $k$ -му кроці навчання;  $\beta_m^{(k)}(d)$  – помилка другого роду;  $D_{1,m}^{(k)}(d)$  – перша достовірність;  $D_{2,m}^{(k)}(d)$  – друга достовірність;  $d$  – дистанційна міра, яка визначає радіуси гіперсферичних контейнерів, побудованих в радіальному базисі простору Хеммінга.

Нормована модифікація критерію (2.5) подається у вигляді [2]

$$E_{K,m}^{(k)} = \frac{E_{K,m}^{(k)}}{E_{K,\max}^{(k)}}, \quad (2.6)$$

де  $E_{K,\max}^{(k)}$  – значення інформаційного критерію при максимальних значеннях першої та другої достовірностей та нульових помилках першого та другого роду.

Оскільки точнісні характеристики є функціями відстані вершин усереднених векторів ознак розпізнавання від геометричних центрів контейнерів відповідних класів розпізнавання, то критерії (2.5) і (2.5) в ІЕІ-технології слід розглядати як нелінійні та взаємно-неоднозначні функціонали від точнісних характеристик, що потребує знаходження в процесі машинного навчання робочої (допустимої) області визначення його функції.

Розглянемо процедуру обчислення модифікованого критерію Кульбака (2.54) для двохальтернативного рішення при рівноймовірних гіпотезах. Інформаційний критерій - функціонал від точнісних характеристик. Точнісні характеристики, при репрезентативному обсязі навчальної вибірки, можна оцінити у вигляді:

$$D_{1,m}^{(k)}(d) = \frac{K_{1,m}^{(k)}}{n_{\min}}; \quad \alpha_m^{(k)}(d) = \frac{K_{2,m}^{(k)}}{n_{\min}}; \quad \beta_m^{(k)}(d) = \frac{K_{3,m}^{(k)}}{n_{\min}}; \quad D_{2,m}^{(k)}(d) = \frac{K_{4,m}^{(k)}}{n_{\min}}, \quad (2.7)$$

де  $K_{1,m}^{(k)}$  – кількість подій, які означають належність реалізацій своєму класу  $X_m^o$ ;  $K_{2,m}^{(k)}$  – кількість подій, які означають неналежність реалізацій своєму класу  $X_m^o$ ;  $K_{3,m}^{(k)}$  – кількість подій, які означають належність чужих реалізацій класу  $X_m^o$ ;  $K_{4,m}^{(k)}$  – кількість подій, які означають неналежність чужих реалізацій класу  $X_m^o$ ;  $n_{\min}$  – мінімальний обсяг репрезентативної навчальної вибірки.

Розрахункова модифікація критерію Кульбака після відповідної підстановки оцінок (2.7) у вираз (2.5) приймає вигляд [1]

$$E_m^{(k)} = \frac{1}{n} \log_2 \left\{ \frac{2n + 10^{-r} - [K_2^{(k)} + K_3^{(k)}]}{[K_2^{(k)} + K_3^{(k)}] + 10^{-r}} \right\} [n - (K_2^{(k)} + K_3^{(k)})], \quad (2.8)$$

де  $10^{-r}$  – деякий  $\epsilon$  для уникнення поділу на 0. а умова  $1 < r \leq 3$  цілком достатня, за якої величина  $\epsilon$  не вплине на точність

Розглянемо схему обчислення змінних  $K_{1,m}^{(k)} - K_{4,m}^{(k)}$  у виразі (2.8). На рисунку 2.1' показано структуру навчальної матриці, яка послідовно складається з векторів структурованих ознак розпізнавання  $\{x_1^{(j)}\} \in X_1^o$  і  $\{x_2^{(j)}\} \in X_2^o$  відповідно.

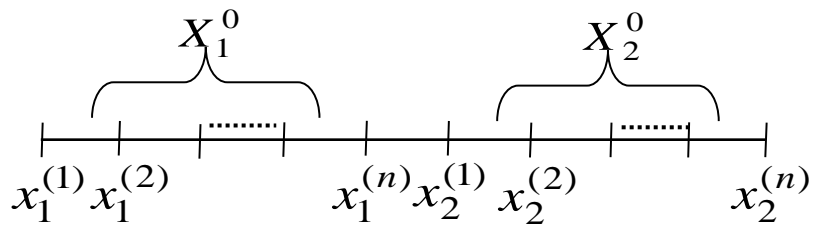


Рисунок 2.1 – Структура навчальної матриці

Алгоритм обчислення коефіцієнтів  $K_1^{(k)}$ ,  $K_2^{(k)}$ ,  $K_3^{(k)}$  і  $K_4^{(k)}$  має такий вигляд [7]:

$$(\forall X_1^o \in \mathfrak{R}^{/A/}) (\forall X_2^o \in \mathfrak{R}^{/A/}) [if x_1^{(j)} \in X_1 then K_1^{(k)} := K_1^{(k-1)} + 1 else K_2^{(k)} := K_2^{(k-1)} + 1];$$

$$(\forall X_1^o \in \mathfrak{R}^{/A/}) (\forall X_2^o \in \mathfrak{R}^{/A/}) [if x_2^{(j)} \in X_1 then K_3^{(k)} := K_3^{(k-1)} + 1 else K_4^{(k)} := K_4^{(k-1)} + 1].$$

Таким чином, наведений інформаційний критерій оптимізації параметрів машинного навчання системи керування протезом кінцівки руки в рамках ІЕІ-технології можна розглядати як узагальнену міру близькості класів розпізнавання, які відновлюються в процесі машинного навчання в радіальному базисі простору ознак розпізнавання Хеммінга.

### 2. 3 Алгоритми інформаційно-екстремального машинного навчання

У внутрішньому циклі багатоциклічної процедури оптимізації параметрів машинного навчання системи керування біонічним протезом реалізується так званий базовий алгоритм. Основними функціями базового алгоритму є обчислення на кожному кроці машинного навчання інформаційного критерію оптимізації параметрів машинного навчання і наближення до глобального максимуму критерію в області визначення його функції.

Як узагальнена схема базового алгоритму машинного навчання розглядається категорійна у вигляді орієнтованого графу, в якому множини відображаються одна на одну відповідними операторами перетворення інформації, що застосовуються в процесі навчання. На рис. 2.2 показано категорійну модель інформаційно-екстремального машинного навчання з оптимізацією геометричних параметрів гіперсферичних контейнерів класів розпізнавання [1].

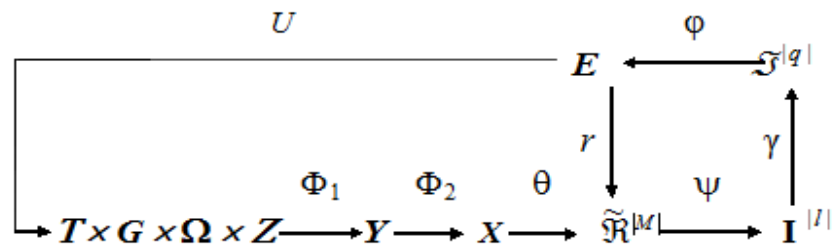


Рисунок 2.2 – Категорійна модель навчання системи керування

На рис. 2.2 оператор  $\theta: X \rightarrow^{[M]} \mathfrak{R}$  будує нечітке в загальному випадку розбиття  $^{[M]} \mathfrak{R}$  бінарного простору ознак на класи розпізнавання, а оператор класифікації  $\psi$  перевіряє основну статистичну гіпотезу про належність навчальної реалізації класу  $X_m^o$  і таким чином формує множину гіпотез  $I^{[l]}$ , де  $l$  – кількість статистичних гіпотез. Оператор  $\gamma$  шляхом оцінки прийнятих гіпотез формує множину точнісних характеристик  $\mathfrak{S}^{[q]}$ , де  $q = l^2$ , а оператор  $\phi$  обчислює множину значень інформаційного критерію  $E$ , який є функціоналом

від точнісних характеристик. Контур моделі, який замикається оператором  $r$ , реалізує ітераційний процес оптимізації геометричних параметрів розбиття  $^M/\mathfrak{R}$  шляхом пошуку глобального максимуму інформаційного критерію оптимізації параметрів машинного навчання в робочій (допустимій) області визначення його функції.

Процес навчання регламентується оператором  $U : V \rightarrow G \times T \times \Omega \times Z$ .

Таким чином, основною процедурою базового алгоритму інформаційно-екстремального навчання є обчислення на кожному кроці навчання інформаційного критерію оптимізації параметрів машинного навчання і організація пошуку глобального максимуму в робочій області визначення його функції.

Категорійну модель у вигляді орієнтованого графа відображень множин, що застосовується при функціонуванні системи розпізнавання в режимі екзамену, показано на рис. 2.3 [8].

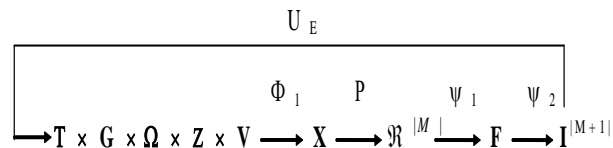


Рисунок 2.3 – Категорійна модель функціонування системи розпізнавання в режимі екзамену

У категорійній моделі (рис. 2.3) оператор  $\Phi_1$  відображає універсум випробувань на вибірку множини  $X$ , яка утворює екзаменаційну матрицю  $\|x_i^{(j)} / i = \overline{1, N}, j = \overline{1, n}\|$ , еквівалентну навчальній матриці за структурою і сформовану за оптимальними параметрами машинного навчання.

Алгоритм екзамену має такі вхідні дані:

□  $\{x_m / m = \overline{1, M}\}$  – масив еталонних двійкових векторів-реалізацій об-

разу, які визначають геометричні центри відповідних оптимальних контейнерів класів розпізнавання, побудованих на етапі навчання;

- $\{d_m\}$  – масив оптимальних радіусів, побудованих на етапі навчання відповідних контейнерів;
- $\{x_s^{(j)} / s = \overline{1, SMAX}; j = \overline{1, n}\}$  – масив двійкових векторів-реалізацій, що ідентифікуються, де  $SMAX$  – кількість реалізацій, що класифікуються;
- $\{\delta_{k,i} / i = \overline{1, N}\}$  – оптимальна система контрольних допусків на ознаки розпізнавання, визначена на етапі навчання.

Алгоритм екзамену у рамках ІЕІ-технології – прийняття детермінованих рішень в залежності від значень вирішальних правил, коефіцієнти яких сформовані на етапі машинного навчання. При цьому побудовані за результатами інформаційноекстремального машинного навчання вирішальні правила можна розглядати одночасно і як функції належності векторів ознак, що розпізнаються, відповідному класу розпізнавання із заданого алфавіту, і як характеристику близькості до ядра контейнера цього класу. Для гіперсферичного контейнера, наприклад, класу розпізнавання  $X_m^o$  вирішальне правило має вигляд [1]

$$\mu_m = 1 - \frac{d(x_m \oplus x^{(j)})}{d_m}. \quad (2.9)$$

де  $x_m$  – центральний вектор класу  $X_m^o$ , вершина якого визначає геометричний центр його оптимального контейнера;  $x^{(j)}$  –  $j$ -та реалізація образу, що розпізнається;  $d_m$  – оптимальний радіус гіперсферичного контейнера класу  $X_m^o$ .

Виводом алгоритму є повідомлення про належність вектора структурованих ознак, що розпізнається, деякому класу із сформованого на етапі машинного навчання алфавіту класів розпізнавання  $\{X_m^o\}$ .

### 3 ІНФОРМАЦІЙНИЙ СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ПРОТЕЗОМ КІНЦІВКИ РУКИ

#### 3.1 Формування вхідного математичного опису системи керування протезом руки з неінвазивною системою зчитування біосигналів

Вхідні дані отримано за допомогою трьох 8-електродних (загалом 24 електродів) браслетів, розміщених навколо передпліччя випробовуваних [35]. У набір даних включено 7 активних жестів та неактивний стан. Сумарна кількість повторень кожного жесту під час одного випробування - 20. Дані було зібрано для 44 учасників в ідентичній експериментальній процедурі, що повторювалася двічі (з тижневим інтервалом) і підвищила кількість повторень кожного жесту до 40. Усі жести виконувались в послідовній або повторюваній манері, тривалість жестів від однієї до трьох секунд. Таким чином представлений набір даних може використовуватися для розробки та перевірки інваріантних за користувачем методів класифікації, визначення стійкості ознак як до довгострокових змін сигналу ЕМГ, так і до неточної перестановки електродів. Використання розрідженої 24-електродної матрицеподібної конфігурації дозволяє розробляти методи, незалежні від розміщення електродів. Також наявні відеозаписи всіх випробувань з двох ракурсів з використанням бінокулярної камери глибини та HD-камери. Відеопотік використовувався для уточнення маркування жестів та виявлення можливих помилок суб'єктів. Набір даних putEMG доступний за адресою <https://www.biolab.put.poznan.pl/putemg-dataset/>.

З метою отримання набору даних putEMG була розроблена спеціальна експериментальна установка.



Установка була розроблена, щоб дозволити записувати активність м'язів передпліччя певного суб'єкту з широким діапазоном анатомічних особливостей, особливо округлості рук. Стенд збору даних putEMG видно на рисунку 3.1 [35].

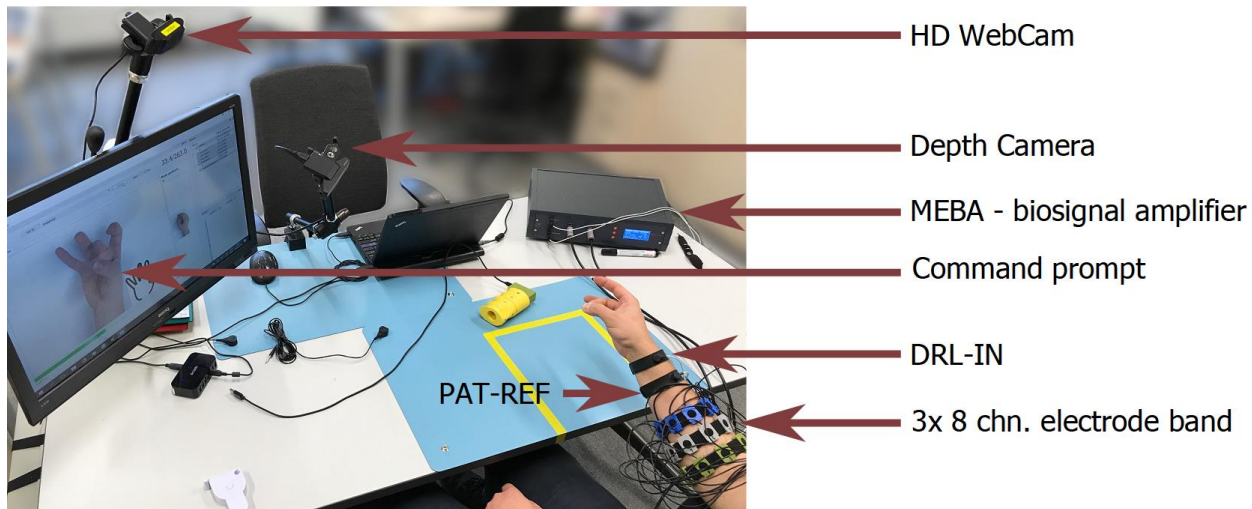


Рисунок 3.1 – Установка, використана під час отримання набору даних putEMG[35]

На рисунку 3.1 показано розміщення датчика sEMG стрічки та контрольних електродів, розташованих біля зап'ястя обстежуваного.

Для запису sEMG сигналу використовували універсальний настільний багатоканальний підсилювач біосигналу MEBA by OT Bioelettronica. Дані відбирали з частотою 5120 Гц за допомогою 12-бітного аналого-цифрового перетворювача (ADC) роздільна здатність і коефіцієнт підсилення 200. Крім того, вбудований аналоговий смуговий фільтр з пропускнуою здатністю від 3 до 900 Гц, застосовувався з метою усунення упередженості та запобігання згладжування. Сигнали були записані в монополярному режимі з електродами DRL-IN та Patient-REF, розміщеними щільно до зап'ястя руки що обстежувалась.

Набір даних putEMG складається із серії випробувань, які включають виконання 8 жестів руками (рис. 3.2). Вибір жестів базувався на попередніх дослідженнях, були обрані активні пози, що демонструють найвищу дискримінацію та найкращі підходи для зручного дизайну НМІ на основі SEMG. Набір складається з 7 активних жестів. Восьмий жест у наборі даних putEMG не працює, під час якого випробовуваних просили не рухати рукою, утримувати її стабілізованою та розслабляти м'язи. Період простою 3 с завжди розділяє виконання активних жестів.

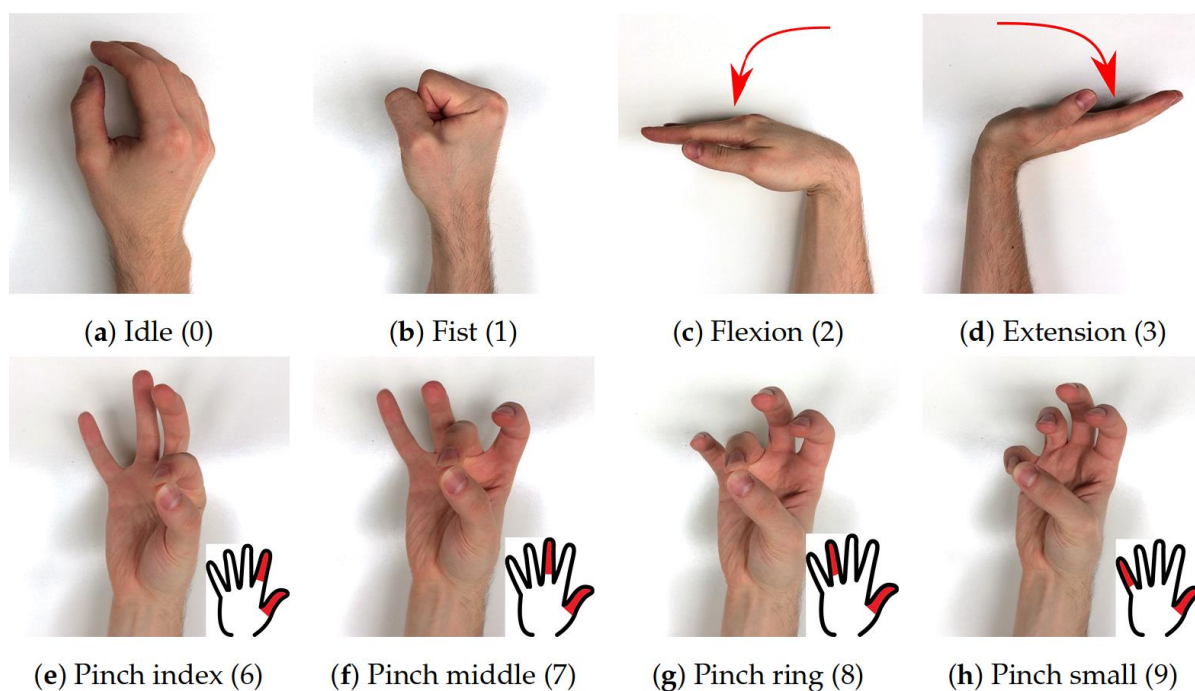
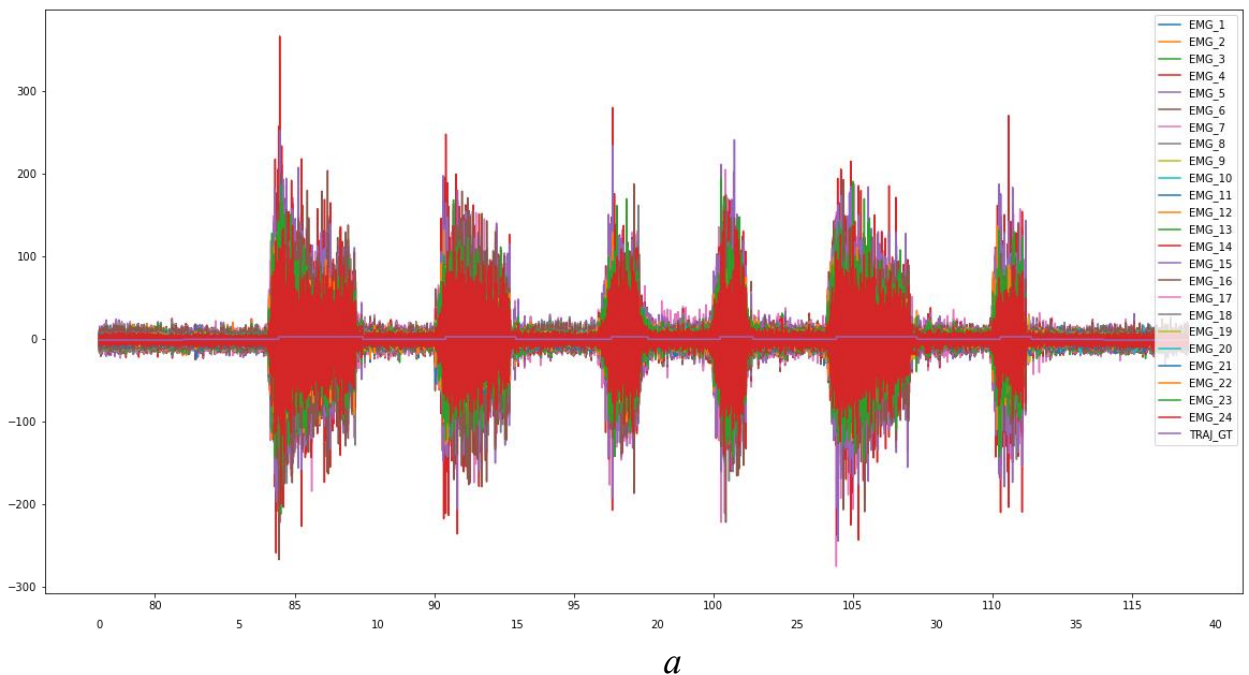


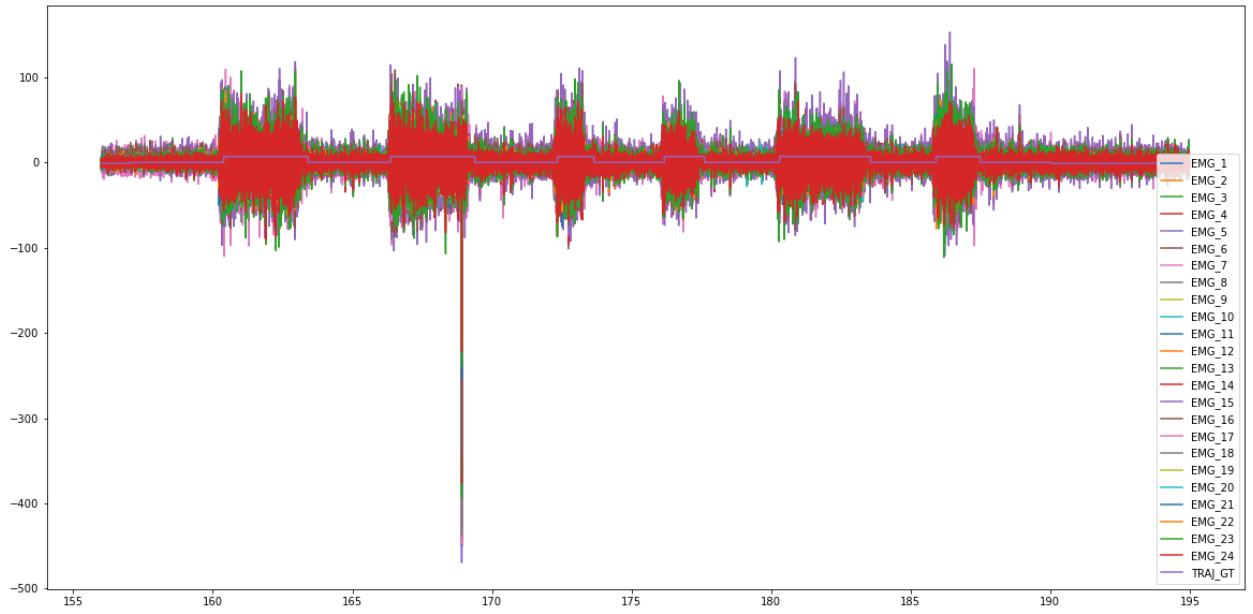
Рисунок 3.2 – Набір жестів PutEMG [35]

PutEMG включає 8 жестів; 7 активних жестів та бездіяльність, коли випробовуваних просили не рухати рукою; цифри в квадратних дужках означають маркер жестів, що використовується у файлах наборів даних putEMG. Виконувалося до 20 повторень кожного активного жесту в одному експерименті, проведеному для одного учасника. Кожен суб'єкт проводив експеримент двічі з принаймні тижневим поділом часу, збільшення кількості активних

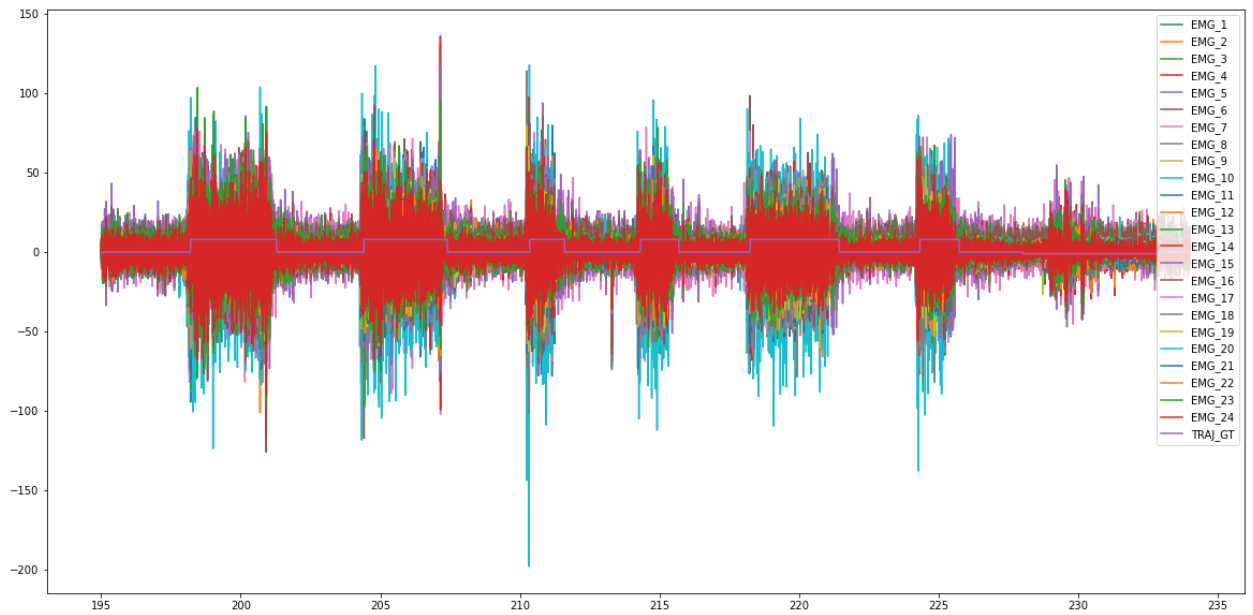
повторень пози до 40. Однак повторюваність розміщення смуги електродів є не повністю забезпечена і може дещо відрізнятись між експериментами.

Алфавіт машинного навчання згідно з рисунком 3.2 складався із шести класів розпізнавання: клас  $X_1^o$  – стисканням ладоні в кулак; клас  $X_2^o$  – згинання долоні; клас  $X_3^o$  – розгинання долоні; клас  $X_4^o$  – стискання в кільце великого і вказівного пальців; клас  $X_5^o$  – клас  $X_6^o$  – стискання в кільце великого і середнього пальців. Приклади біосигналів заданих класів розпізнавання представлені на рисунку 3.3 [35].





$\delta$



$\delta$

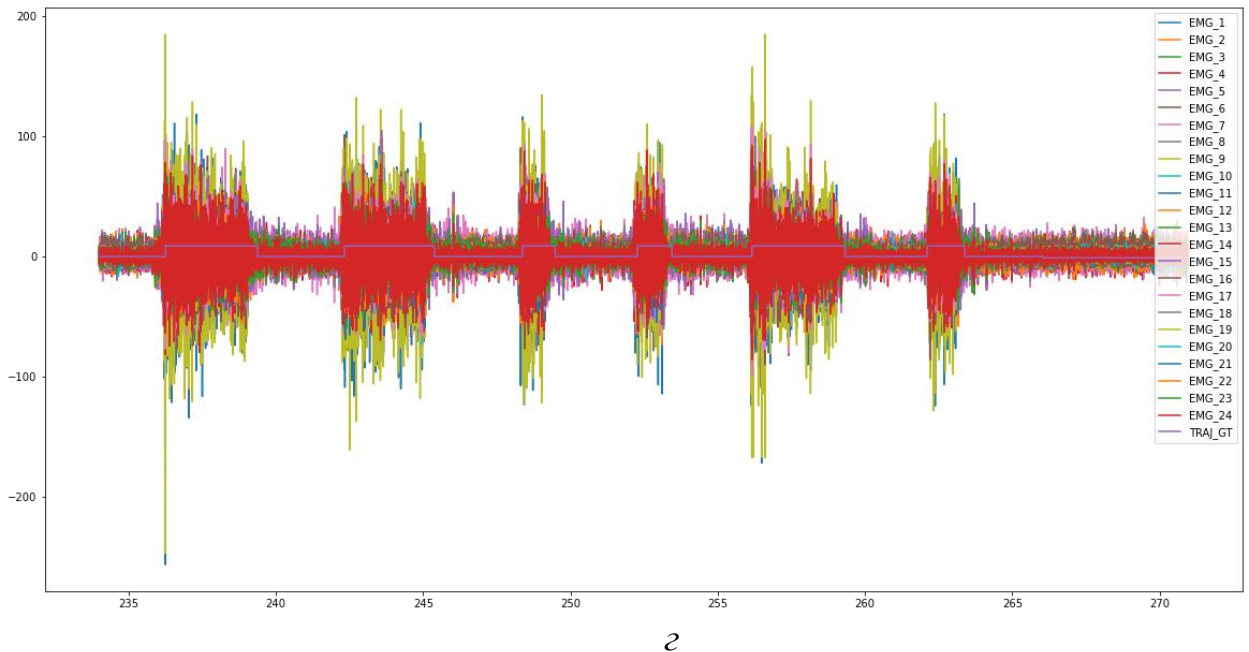


Рисунок 3.3 – Приклади біосигналів серії виконань певного жесту для заданого алфавіту класів розпізнавання: *a* – Кулак (клас  $X_1^o$ ); *б* – Флексія (клас  $X_2^o$ ); *в* – Екстензія (клас  $X_3^o$ ); *г* – Щипок (клас  $X_4^o$ );

Аналіз рис. 3.3 показує, що наявність в реалізаціях класів розпізнавання однакових ознак обумовлює їх перетин в просторі ознак. Крім того, біосигнали мають відносно великий шумовий фон. Ці фактори суттєво ускладнюють задачу розпізнавання біосигналів.

Формування вектору-реалізації відповідного класу розпізнавання здійснювалося шляхом квантування за часом біосигналу із періодом  $20\text{ ms}$  на заданому часовому інтервалі  $2\text{ s}$ . Тобто кожна реалізація образу складалася із 100 ознак розпізнавання. При цьому з метою фільтрації шуму квантування біосигналу починалося з моменту досягнення його амплітудою порогового значення, яке дорівнювало  $30\text{ mV}$ . У результаті для заданого алфавіту  $\{X_m^o / m = \overline{1,3}\}$  класів розпізнавання була сформована вхідна навчальна матриця у вигляді тривимірного масиву  $\{y_{m,i}^{(j)}\}$ , в якому кількість ознак розпізнавання дорівнювала  $N = 100$ , а кількість векторів-реалізацій для кожного класу

розпізнавання дорівнювала  $n = 40$ . Нормоване поле допусків на ознаки розпізнавання, яке задавало область значень відповідних контрольних допусків, дорівнювало  $\delta_H = 60 \text{ mV}$ .

### 3.2 Категорійна модель машинного навчання системи керування протезом кінцівки руки

Процес керування протезом, де керуючим сигналом виступають результати обробки міографічних даних, отриманих з неінвазивних датчиків є слабо формалізованим через об'єктивну невизначеність умов формування образів керуючих сигналів, викликану різницею в розміщенні датчиків, змінами вологості шкіри, втомою м'язів. Таким чином, категорійна модель інформаційно-екстремального навчання системи розглядається у вигляді орієнтованого графу, ребра якого характеризують оператори відображення множин. При цьому вхідний математичний опис матиме наступну структуру

$$\Delta_B = \langle G, G, T, \Omega, Z, Y, X; f_1, f_2, f_3 \rangle,$$

де  $G$  – множина вхідних факторів – біосигналів, які зчитуються системою;  $G$  – множина відфільтрованих біосигналів;  $T$  – множина моментів часу зняття інформації;  $\Omega$  – простір ознак розпізнавання;  $Z$  – простір можливих функціональних станів керованого процесу;  $Y$  – множина векторів реалізацій класів розпізнавання, яка утворює вхідну навчальну матрицю;  $X$  – бінарна навчальна матриця;  $f_1$  – оператор фільтрації біосигналів;  $f_2 : G \times T \times \Omega \times Z \rightarrow Y$  – оператор формування вхідної навчальної матриці  $Y$ ;  $f_3 : Y \rightarrow X$  – оператор формування бінарної навчальної матриці  $X$ .

У загальному випадку адаптація математичного опису системи на вході класифікатора за ІЕІ-технологією з метою побудови безпомилкових за

навчальною матрицею вирішальних правил полягає в багатоциклічній ітераційній процедурі оптимізації за інформаційним критерієм просторово-часових параметрів, які впливають на функціональну ефективність системи керування, що навчається.

Нехай дано алфавіт класів розпізнавання  $\{X_m^0 / m = \overline{1, M}\}$ , навчальна матриця типу “об’єкт-властивість”  $y_{m,i}^{(j)} i = \overline{1, N}, j = \overline{1, n}$ , де  $N, n$  – кількість ознак розпізнавання та реалізацій образу відповідно. Задано структурований вектор параметрів функціонування системи, що навчається розпізнавати реалізації класу  $X_m^0$ :

$$g = \langle x_m, d_m, \delta, \rho_m \rangle, \quad (3.1)$$

де  $x_m$  – усереднений структурований вектор ознак класу розпізнавання  $X_m^0$ ;  $d_m$  – радіус гіперсферичного контейнера класу розпізнавання  $X_m^0$ , який відновлюється в радіальному базисі простору ознак розпізнавання;  $\delta$  – параметр поля контрольних допусків на ознаку розпізнавання.

Параметр  $\delta$  дорівнює половині симетричного поля контрольних допусків на ознаки розпізнавання, як це показано на рисунку 3.4.

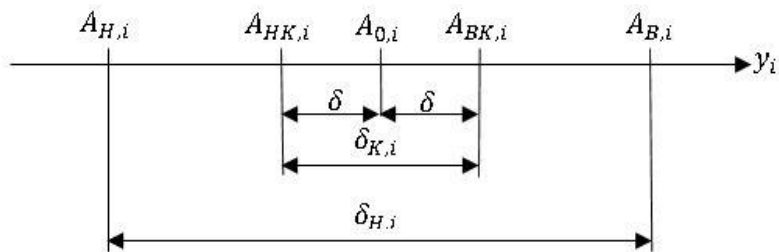


Рисунок 3.4 – Поле допусків на ознаку розпізнавання

На рисунку 3.4 прийнято такі позначення:  $A_{0,i}$  – номінальне значення ознаки  $y_i$ ;  $A_{H,i}$  – нижній нормований (експлуатаційний) допуск;  $A_{B,i}$  – верхній нормований допуск;  $A_{HK,i}$  – нижній контрольний допуск;  $A_{BK,i}$  – верхній контрольний допуск;  $\delta_{K,i}$  – поле контрольних допусків;  $\delta_{H,i}$  – поле нормованих допусків. .

У виразі (3.1) присутній додатковий параметр машинного навчання  $\rho_m$  – рівень селекції координат усередненого двійкового вектора ознак класу розпізнавання  $X_m^o$ . Згідно з працею [1] рівнем селекції називається рівень квантування дискрет полігона емпіричних частот потрапляння значень ознак розпізнавання у свої поля контрольних допусків. Полігон будується для кожного класу таким чином: по осі абсцис відкладаються ранги ознак розпізнавання, які відповідають номерам ознак у векторі-кортежі  $x_m^{(j)}$ , а по осі ординат – відносні частоти  $\omega_{m,i} = n_i / n$ , де  $n_i$  – кількість випробувань, при яких значення  $i$ -ї ознаки знаходиться в своєму полі контрольних допусків. За замовчуванням приймається рівень селекції  $\rho_m = 0.5$ . Оптимізація рівнів селекції координат усереднених двійкових векторів ознак дозволяє максимізувати середню міжкласову кодову відстань для заданого алфавіту, що згідно з максимально-дистанційним принципом теорії розпізнавання образів дозволяє підвищити повну ймовірність прийняття правильних класифікаційних рішень.

Категорійну модель інформаційного машинного навчання системи керування протезом руки з оптимізацією контрольних допусків на ознаки розпізнавання і рівнів селекції координат усереднених двійкових векторів ознак показано на рисунку 3.5.



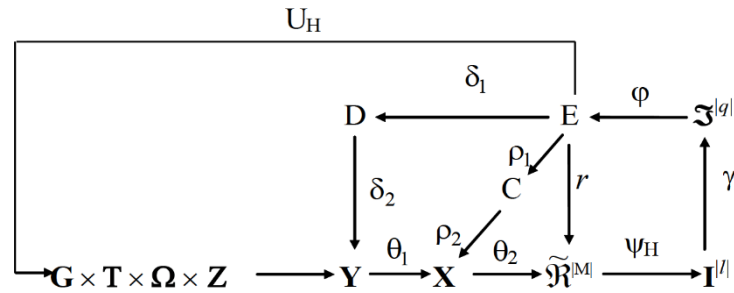


Рисунок 3.5 – Категорійна модель навчання з оптимізацією рівнів селекції координат двійкових усереднених векторів ознак розпізнавання

На рисунку 3.5 прийнято такі позначення:

$\theta$  – оператор відображення бінарних векторів-реалізації навчальної матриці  $X$  на розбиття  $\mathfrak{R}^{|M|}$  простору ознак на класи розпізнавання

$\psi$  - перевірка основної статистичної гіпотези про належність реалізацій відповідному класу розпізнавання. Формує множину статистичних гіпотез  $I^{|S|}$  в кількості  $s$

$\gamma$  - формування множини точнісних характеристик  $\mathfrak{S}^{|q|}$ , де  $q = s^2$ .

$\phi$  - обчислення множини  $E$  значень інформаційного критерію оптимізації параметрів навчання,

$r$  – оператор відновлення в радіальному базисі простору ознак контейнерів класів розпізнавання.

$D$  – терм-множина, система контрольних допусків, що є рівнями квантування ознак при формуванні робочої бінарної навчальної матриці, яка в процесі навчання адаптується для високої достовірності вирішальних правил.

Оператори контура категорійної моделі, показаного на рис. 3.6, оптимізують рівні селекції координат двійкових усереднених векторів ознак розпізнавання.

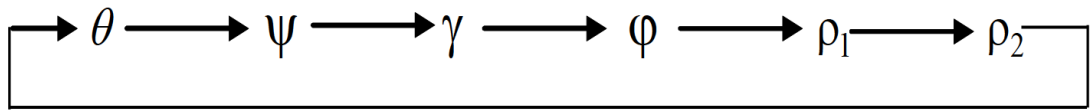


Рисунок 3.6 – Контур оптимізації рівнів селекції координат двійкових усереднених векторів ознак розпізнавання.

Таким чином, показана на рис. 3.5 категорійна модель може розглядатися як узагальнена структурна схема алгоритму інформаційно-екстремального навчання системи керування. При цьому згідно з принципом відкладених рішень для максимізації інформаційної спроможності системи керування може знадобитися оптимізація інших параметрів навчання. У цьому випадку категорійна модель буде мати додаткові контури оптимізації цих параметрів, у яких загальною буде терм-множина  $E$ , елементи якої обчислюються на кожному кроці машинного навчання.

### 3.3 Оптимізація параметрів навчання системи керування протезом кінцівки руки

Згідно з категорійною моделлю (рис. 3.5), інформаційно-екстремальний алгоритм машинного навчання системи керування біонічного протезу з оптимізацією системи контрольних допусків на ознаки розпізнавання подається у вигляді процедури пошуку глобального максимуму інформаційного критерію (2.8) в робочій області визначення його функції

$$\{\delta_{K,i}^* \mid i = \overline{1, N}\} = \arg \max_{G_\delta} \{ \max_{G_E \cap \{k\}} \bar{E}^{(k)} \}, \quad (3.2)$$

де

$$\bar{E}^{(k)} = \frac{1}{M} \sum_{m=1}^M E_m^{(k)}$$

усереднене за алфавітом класів розпізнавання значення інформаційного критерію оптимізації, обчислене на  $k$ -му кроці машинного навчання;  $E_m(d)$  – інформаційний критерій оптимізації параметрів навчання системи розпізнавати реалізації класу  $X_m^o$ ;  $G_\delta$  – допустима область значень параметра  $\delta$  поля контрольних допусків на ознаки розпізнавання;  $G_E$  – робоча (допустима) область визначення критерію  $E_m(d)$ ;  $\{k\}$  – множина значень радіусів гіперсферичних контейнерів класів розпізнавання.

Як інформаційний критерій оптимізації параметрів машинного навчання будемо використовувати модифіковану міру Кульбака (2.8).

Вхідною інформацією для алгоритму навчання є масив вхідної навчальної матриці  $\{y_{m,i}^{(j)}\}$  і система полів нормованих допусків  $\{\delta_{H,i}\}$  на ознаки розпізнавання, яка задає область значень відповідних контрольних допусків.

Розглянемо основні етапи реалізації алгоритму навчання системи керування з паралельною оптимізацією контрольних допусків на ознаки розпізнавання, за якою на кожному кроці навчання змінюються контрольні допуски для всіх ознак розпізнавання одночасно:

1. ініціалізація лічильника кроків зміни параметра  $\delta$  поля контрольних допусків на ознаки розпізнавання:  $\delta := 0$ ;
2.  $\delta := \delta + 1$ ;
3. обчислюються для всіх ознак розпізнавання нижні  $A_{HK,i}$  і верхні  $A_{BK,i}$  контрольні допуски на ознаки розпізнавання:

$$A_{HK,i} = \bar{y}_i - \delta; \quad A_{BK,i} = \bar{y}_i + \delta. \quad (3.3)$$

4. реалізується внутрішній цикл процедури (3.2), який полягає в обчисленні на кожному кроці зміни радіусів контейнерів класів розпізнавання інформаційного критерію оптимізації параметрів навчання, пошуку глобального максимуму інформаційного критерію і визначенні оптимальних радіусів контейнерів класів розпізнавання за процедурою

$$d_m = \arg \frac{1}{M} \sum_{m=1}^M E_m(d_m) ;$$

5. якщо  $\delta < \delta_H / 2$ , то виконується пункт 2, інакше – пункт 6;

6. обчислюється усереднене за алфавітом класів розпізнавання максимальне значення критерію  $\bar{E}$ ;

7. визначається оптимальний параметр поля контрольних допусків, який забезпечує максимальне значення усередненого критерію  $\bar{E}$ :

$$\delta = \arg \bar{E} ;$$

8. обчислюється за формулою (3.3) оптимальна система контрольних допусків на ознаки розпізнавання:

$$A_{HK,i} = \bar{y}_i - \delta; \quad A_{BK,i} = \bar{y}_i + \delta;$$

9. ЗУПИН.

Таким чином, параметрами інформаційно-екстремального навчання системи керування за процедурою (3.2) є оптимальні радіуси  $\{d_m\}$  гіперсферичних контейнерів класів розпізнавання, оптимальні усереднені вектори-реалізації  $\{x_m\}$  класів розпізнавання і оптимальний параметр  $\delta$  поля контрольних допусків на ознаки розпізнавання.

Отримані за результатами паралельної оптимізації контрольні допуски на ознаки розпізнавання є квазіоптимальними, оскільки вони на кожному кроці навчання змінювалися для всіх ознак на однакову величину.

Підвищення функціональної ефективності машинного навчання можна досягнути шляхом реалізації паралельно-послідовної оптимізації контрольних допусків на ознаки розпізнавання. При цьому отримані за результатами паралельної оптимізації квазіоптимальні контрольні допуски на ознаки розпізнавання розглядаються як стартові для послідовної оптимізації. Такий підхід дозволяє підвищити як достовірність прийняття рішень, так і оперативність алгоритму послідовної оптимізації, оскільки пошук глобального максимуму інформаційного критерію не виходить за межі робочої області визначення його функції.

Послідовна оптимізація контрольних допусків на ознаки розпізнавання в процесі машинного навчання здійснюється за процедурою наближення глобального максимуму інформаційного критерію оптимізації (2.) до граничного значення в робочій області визначення його функції.

За схемою реалізації послідовна оптимізація відрізняється від процедури паралельної оптимізації тим, що на кожному кроці машинного навчання послідовно змінюються контрольні допуски тільки для однієї ознаки розпізнавання. А оскільки контрольні допуски починають змінюватися для кожної ознаки при неоптимальних допусках інших ознак, то процедура (8) потребує декількох її прогонів до тих пір, поки значення інформаційного критерію перестануть змінюватися.

За отриманими в процесі машинного навчання оптимальними геометричними параметрами навчання будуються вирішальні правила для прийняття класифікаційних рішень при функціонуванні системи керування безпосередньо в робочому режимі. Для гіперсферичних контейнерів класів розпізнавання вирішальні правила мають вигляд [3]

$$(\forall X_m^o \in \mathfrak{R}^{M/})(x^{(j)} \in \mathfrak{R}^{M/}) [if (\mu_m > 0) \wedge (\mu_m > \mu_c) then x^{(j)} \in X_m^o], \quad (3.4)$$

де  $x^{(j)}$  – вектор-реалізація, що розпізнається;  $\mu_m, \mu_c$  функції належності реалізації, що розпізнається, контейнерам сусідніх (найближчих) класів розпізнавання  $X_m^o$  і  $X_c^o$  відповідно.

У виразі (3.4) відповідні функції належності для гіперсферичних контейнерів визначаються за формулою

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \quad (3.5)$$

де  $x_m^*$  – оптимальний усереднений двійковий вектор ознак;  $d_m^*$  – оптимальний радіус гіперсферичного контейнера

Таким чином, оптимізація системи контрольних допусків на ознаки розпізнавання полягає в організації пошуку в процесі машинного навчання глобального максимуму інформаційного критерію (2.5) в робочій (допустимій) області визначення його функції.

### 3.4 Короткий опис програмного забезпечення

Реалізацію функціональних операторів алгоритму машинного навчання та екзамену системи керування протезом кінцівки руки з неінвазивною системою зчитування біосигналів було здійснено на мові Python у середовищі Jupyter Lab, із використанням бібліотек numpy, matplotlib, pandas та numba.

Програмна реалізація містить:

- 1) Завантаження та візуалізацію вхідних даних
- 2) Різні сценарії предобробки даних та фільтрації шумів, що формують навчальну матрицю.

- 3) Відображення параметрів функціонування інтелектуальної інформаційної системи в процесі навчання в текстовому форматі;
- 4) Графічне відображення динаміки змін параметрів навчання в графічному вигляді після завершення навчання;
- 5) Алгоритм навчання оптимальних гіперпараметрів ієрархічного вирішального правила інтелектуальної системи керування, що наближує її інформаційну спроможність до максимальної асимптотичної. за паралельно послідовним алгоритмом навчання ІЕІТ із оптимізацією за рівнем селекції двійкових координат центральних векторів контейнерів класів розпізнавання.
- 6) Алгоритм екзамену за ІЕІТ.
- 7) Збереження винайдених оптимальних параметрів функціонування системи в файли.

Програма являє собою документ формату Jupyter Notebook, що дозволяє поєднувати в одному візуальному середовищі ділянки виконуваного коду, коментарі та інфографіку.

Таким чином програма має зручний та гнучкий інтерфейс із інтегрованими підказками для користувача, коментарями для глибшого розуміння функціонування скриптів програми та наочними візуалізаціями вхідних даних та параметрів в процесі навчання.

Після запуску файлу в середовищі Jupyter Notebook або Jupyter Lab, керування здійснюється редагуванням констант, що відповідають іменам файлів з вхідними даними і параметрам формування навчальної вибірки, та запуском ділянок-скриптів що реалізують певні функції програми. Кількість класів розпізнавання майже не обмежена, однак програма має деякі обмеження щодо об'єму вибірки вхідних даних та потужності словника ознак розпізнавання. Об'єм вибірки обмежується оперативною пам'яттю ЕОМ і разом з потужністю алфавіту ознак розпізнавання сильно впливає на час виконання програми. Таким чином великі вибірки та потужності спричиняють сильні незручності з великим часом навчання.

Більшість наявних сценаріїв предобробки даних націлена на вибірки з набору `putEMG`, однак самі алгоритми навчання та екзамену є використованими для будь-яких даних, що можуть бути зведені до масивів форми класи-реалізації-ознаки(3D) та вибірка-реалізація(1D-2D) відповідно.

Алгоритм є інваріантним до кількості ознак та реалізацій, і таким чином формує вектори оптимальних гіперпараметрів відповідно до розмірності вхідних даних.

Властивість адаптивності при оптимізації словника ознак розпізнавання задається сценаріями предобробки вхідних даних, які формують навчальну вибірку та мають запускатися на виконання перед запуском сценарію навчання. Сформовані навчальні вибірки із оптимізованим словником ознак розпізнавання зберігаються у вигляді масивів в оперативній пам'яті та можуть бути збережені в файли, назва яких відповідає змісту, завдяки призначенням для цього додатковим сценаріям. Програмна реалізація використовує наступні бібліотеки вказані в таблиці 3.1.

Таблиця 3.1 – Використані бібліотеки та їх призначення

№	Назва бібліотеки	Опис
1.	<code>pandas</code>	Зчитування та реорганізація даних.
2.	<code>numpy</code>	Математичні та логічні операції над масивами даних під час виконання алгоритму
3.	<code>matplotlib</code>	Візуалізація даних
4.	<code>numba</code>	Компіляція коду функцій на мові Python в код на мові C і далі в байткод, що пришвидшує їх виконання до 1000 разів.

У програмі було виконано наступні константи і змінні (табл. 3.2):



Таблиця 3.2 – Основні константи та змінні програми

Ім'я константи або змінної	Короткий опис
df	Змінна типу Pandas DataFrame, що дозволяє завантажити дані для навчання з файлу.
N=20	Кількість класів розпізнавання.
k=60	Кількість послідовних дискрет-замірів, з котрих екстрагуються ознаки.
Numpy.ndarray(dtype=float64) LearnArrs	Матриця релізацій підготована для навчання
Numpy.ndarray(dtype=float64) TestArrs	Матриця реалізацій підготована для тестування
Numpy.ndarray(dtype=float64) LearnGestArrs	Матриця реалізацій використовується та змінюється впродовж виконання навчання
Numpy.ndarray(dtype=float64) centers	Масив середніх за реалізаціями для кожного класу розпізнавання
List C	Список рівнів селекції
float64 k1bk float64 best_kulbk	Поточне та краще для певного базового класу значення КФЕ Кульбака
K2, K3	Значення точносних характеристик
Numpy.ndarray(dtype=bool) binary	Матриця двійкових реалізацій
bs_othr_cntr_dst bs_center_all_reals_dst	Кодові відстані між бінарними центрами базового та інших класів навчання, центром базового та реалізаціями всіх

Крім того було розроблено наступні процедури та функції (табл. 3.3):

Таблиця 3.3 – Основні функції програми

Назва функції	Короткий опис
heming(arrA, arrB)	Відстань Хемінга між двома реалізаціями
center_reals( bilinearized_center, binary)	Повертає масив відстаней від бінарного центру базового класу до всіх реалізацій навчальної вибірки.
k13(rds, h_cnr_rls)	Залежно від вхідних даних повертає значення коефіцієнту K1 або K3.
k24(rds, h_cnr_rls)	Залежно від вхідних даних повертає значення коефіцієнту K2 або K4.
kulbak(N,K2,K3)	Значення критерію Кульбака
kulbak_mean(radius, h_cnr_rls, z, N)	Середнє значення інформаційного критерію відносно всіх класів n
bilinearize (image, top, bottom)	Переведення сигналу в бінарний габарит відносно суми контрольних допусків
m_radius_opt(h_cnr_rls, r_max, z, N)	Оптимізація радіусу контейнеру бінарного класу.

### 3.5 Результати фізичного моделювання

Моделювання машинного навчання системи керування протезом кісті руки здійснювалося за двома схемами. У першому випадку (рис. 3.7) машинне навчання здійснювалося для заданого алфавіту з вхідними даними, сформованими при пороговому значенні біосигналу, яке дорівнювало 30 mV. У іншому

випадку машинне навчання системи керування здійснювалося з оптимізацією рівнів квантування біосигналів для кожного класу розпізнавання (рис. 3.8)

На рис. 3.7 показано графіки залежності усередненого за алфавітом класів розпізнавання інформаційного критерію (2.8) від параметра  $\delta$  поля контрольних допусків, отримані в процесі машинного навчання системи керування біопротезом з паралельною оптимізацією контрольних допусків при пороговому значенні біосигналу 30 mV. На рис. 3.7 кольоровою заливкою позначено робочі (допустимі) області визначення критерію (2.8), в яких перша і друга достовірності перевершують відповідно помилки першого і другого роду. При цьому оптимальне значення параметра  $\delta$  визначається максимальною величиною інформаційного критерію у всіх робочих областях.

Аналіз рисунка 3.7 показує, що у першому випадку машинне навчання характеризується невисокою функціональною ефективністю, що потребує збільшення його глибини шляхом оптимізації додаткових параметрів машинного навчання. Одним із таких параметрів оптимізації є рівні квантування біосигналів. В процесі машинного навчання рівні квантування біосигналів змінювалися з кроком 10mV. При цьому значення дискрет біосигналу, які знаходилися нижче рівня квантування, приймалися нульовим, а ті, що знаходилися вище рівня квантування приймалися одиничними. Таким чином, утворювалася так звана “зріджена” навчальна матриця, яка відрізнялася від початкової меншою кількістю одиниць. Аналіз рисунка 3.8 показує, що застосування “зрідженої” навчальної матриці для заданого алфавіту призвело до значного збільшення функціональної ефективності машинного навчання.

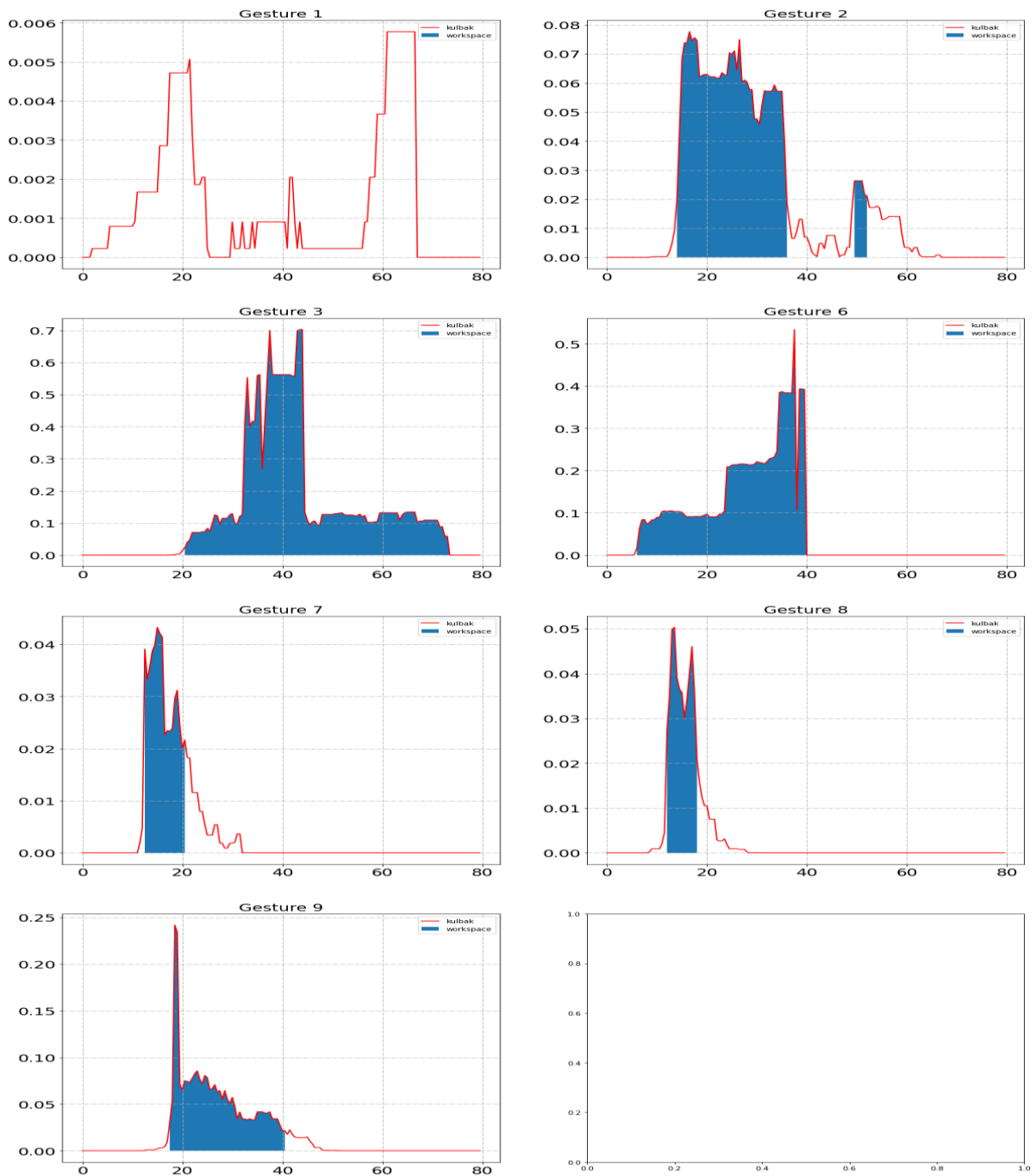


Рисунок 3.7 – Графіки залежності нормованого інформаційного критерію від

параметра поля контрольних допусків:  $a$  – клас  $X_0^o$ ;  $b$  – клас  $X_1^o$ ;  $v$  – клас

$X_2^o$   $z$  – клас  $X_3^o$ ;  $t$  – клас  $X_4^o$ ;  $d$  – клас  $X_5^o$   $e$  – клас  $X_6^o$

На рисунку 3.8 показано графіки залежності нормованого інформаційного критерію від параметра поля контрольних допусків при оптимальних рівнях квантування біосигналів.

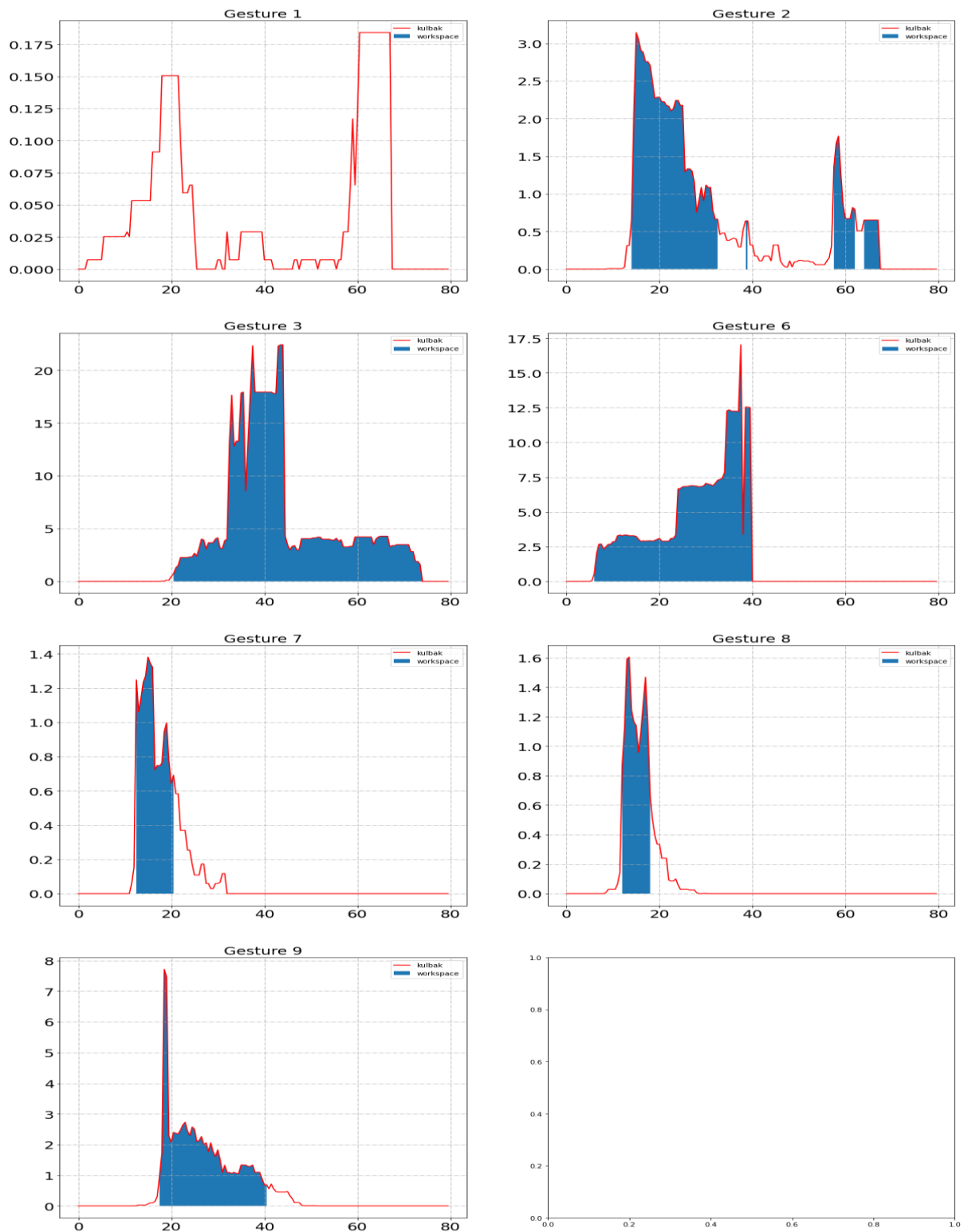


Рисунок 3.8 – Графіки залежності нормованого інформаційного критерію від параметра поля контрольних допусків при оптимальних рівнях квантування-біосигналів:  $a$  – клас  $X_0^o$ ;  $b$  – клас  $X_1^o$ ;  $v$  – клас  $X_2^o$ ;  $z$  – клас  $X_3^o$ ;  $r$  – клас  $X_4^o$ ;  $\partial$  – клас  $X_5^o$ ;  $e$  – клас  $X_6^o$

Для побудови вирішальних правил необхідно знання оптимальних в інформаційному розумінні геометричних параметрів контейнерів класів розпізнавання. На рисунку 3.10 показано графіки залежності інформаційного критерію (2.8) від радіусів контейнерів класів розпізнавання, одержаних при оптимальному параметрі поля контрольних допусків на ознаки розпізнавання.

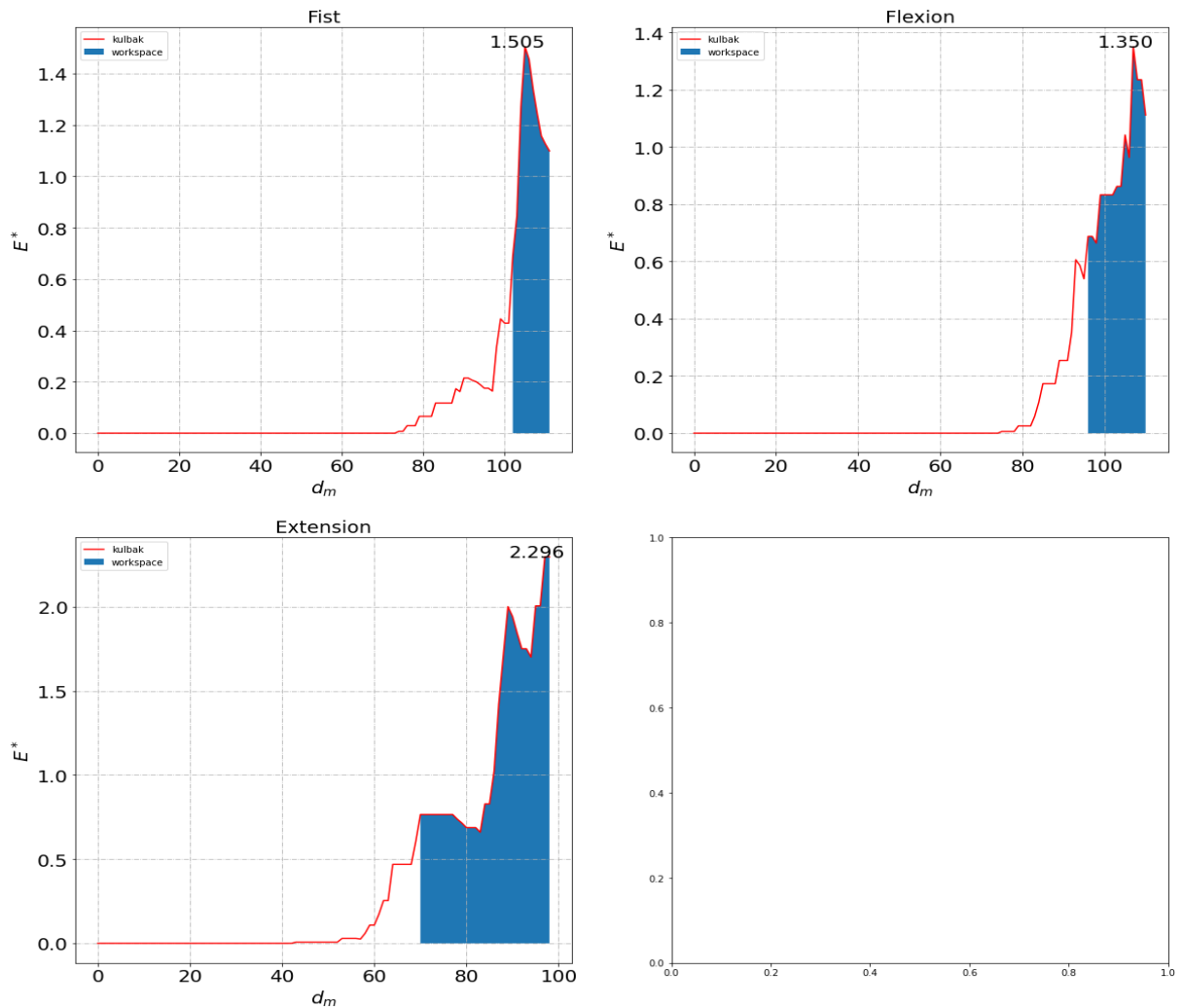


Рисунок 3.10 – Графіки залежності інформаційного критерію (2.8) від радіусів контейнерів класів розпізнавання:  $a$  – клас  $X_1^o$ ;  $b$  – клас  $X_2^o$ ;

$v$  – клас  $X_3^o$

Аналіз рис. 3.10 показує, що оптимальні значення радіусів контейнерів класів розпізнавання, визначені за результатами інформаційно-екстремального машинного навчання з паралельною оптимізацією контрольних допусків

на ознаки розпізнавання і оптимізацією рівнів квантування біосигналів для заданого алфавіту класів розпізнавання дорівнюють: для класу  $X_1^o - d_1 = 22$   
 $d_1 = 104$ (тут і далі в кодових одиницях), для класу  $X_2^o - d_2 = 108$  і для класу  
 $X_3^o - d_3 = 94$ .

Аналіз функціональної ефективності машинного навчання системи керування протезом кінцівки руки з паралельною оптимізацією контрольних допусків на ознаки розпізнавання і з оптимізацією рівнів квантування біосигналів для заданого алфавіту показав, що згідно з рисунком 3.7 середня значення інформаційного ненормованого критерію дорівнює 1,65, а згідно з рисунком 3.8 – 8,3. Таким чином суттєве підвищення функціональної ефективності машинного навчання майже у сім разів досягнуто за рахунок використання “зрідженої” навчальної матриці, що зменшує ступінь перетину класів розпізнавання в просторі ознак.

## 4 ВИСНОВКИ

1. Алгоритми машинного навчання для встановлення відповідності між біосигналами та командами на основі нейронних мереж та методу опорних векторів через багатовимірність словника ознак і суттєвий перетин класів розпізнавання не дозволяють досягти достатньо високої точності розпізнавання біосигналів. До того ж у великій кількості лабораторних досліджень розпізнавання образів час відклику систем на основі цих та інших поширених методів перевищує природний у 1,5 – 2,5 рази для повільних та швидких рухів відповідно.

2. Реалізовано метод інформаційно-екстремального машинного навчання системи керування протезом кісті руки з неінвазивною системою зчитування біосигналів, що дозволило надати системі керування властивості адаптивності до довільних початкових умов і гнучкості при перенавчанні системи через розширення алфавіту класів розпізнавання..

3. В даному дослідженні розроблено нову модифікацію алгоритму інформаційно-екстремального машинного навчання, яка підвищує точність багатокласового розпізнавання шляхом оптимізації рівня квантування біосигналів для заданого алфавіту класів розпізнавання, а також вперше на її основі реалізовано класифікатор міографічних даних для шести рухів.

4. Порівняно з попередніми роботами по даній темі, використано нову якісну, велику та різноманітну вибірку міографічних даних, зібрану польськими науковцями з Познаньського університету.



## 5 СПИСОК ЛІТЕРАТУРИ

1. Довбиш А.С. Основи проектування інтелектуальних систем : навчальний посібник / А. С. Довбиш. – Суми : Видавництво СумДУ, 2009. – 171 с.

2. Москаленко В. В. Вступ до інформаційного аналізу і синтезу інфокомунікаційних систем : навч. посіб. / В. В. Москаленко, А. С. Довбиш. – Суми : Сумський державний університет, 2016. – 226 с.

1. Довбиш А. С. Оптимізація ієрархічної структури даних інтелектуальної системи функціонального діагностування технічного стану складної машини / А. С. Довбиш, В. І. Зимовець, М. В. Бібик // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Харків, 2018. № 44 (1320). С. 42–49.

2. Довбиш А. С., Москаленко В.В., П'ятаченко В.Ю. Інформаційно-екстремальне машинне навчання системи керування протезом руки / А. С. Довбиш, В. В. Москаленко, В. Ю. П'ятаченко // Радіoeлектронні і комп'ютерні системи, 2017. – №4(84).– С. 40 – 49.

3. C. Sommer and D. Gerlich, "Machine learning in cell biology – teaching computers to recognize phenotypes", Journal of Cell Science, vol. 126, no. 24, pp. 5529-5539, 2013, doi:10.1242/jcs.123604.

4. C. M. Bishop. Pattern Recognition and Machine Learning, Berlin, Heidelberg: Springer, 2011.

5. Benatti S. Analysis of robust implementation of an emg pattern recognition based control / S. Benatti, E. Farella, L. Benini, E. Gruppioni // Conference: International Conference on Bio-inspired Systems and Signal. – Angers, France : BIOSIGNALS. – 2014. – P. 45-54.

6. Farrell T. R. A comparison of the effects of electrode implantation and targeting on pattern classification accuracy for prosthesis control / T. R. Farrell, R.F. Weir // Biomedical Engineering. – IEEE Press, Vol. 55, N. 9. – 2008. – P. 2198-2211.

7. T. Zhang, L. Jiang and H. Liu, "Design and Functional Evaluation of a Dexterous Myoelectric Hand Prosthesis With Biomimetic Tactile Sensor," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 26, no. 7, pp.1391-1399, July 2018.

8. G. Liang, Y. Wang, D. Mei, K. Xi and Z. Chen, "Flexible Capacitive Tactile Sensor Array With Truncated Pyramids as Dielectric Layer for Three-Axis Force Measurement," in Journal of Microelectromechanical Systems, vol. 24, no. 5, pp. 1510-1519, Oct. 2015.

9. T. Zhang and L. Jiang, "Biomimetic Tactile Data Driven Closed-loop Control of Myoelectric Prosthetic Hand-," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 1738-1742.

10. Chowdhury R. H. Surface Electromyography Signal Processing and Classification Techniques / R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali, A. A. A. Bakar, K. Chellappan, T. G. Chang // Sensors. – Basel, Switzerland : MDPI. – 2013. – P. 12431-12466.

11. EP 2 813 194 A1 Control of limb device / Popovic D., Graimann B., Markovic M., Dosen S. // Applicants: Georg-August Universität Göttingen (DE), Otto Bock HealthCare GmbH (DE).– Date of filing: 12.06.2013, Application number: 13171671.4, Date of publication: 17.12.2014 Bulletin 2014/51.

12. Q. K. Al-Shayea, "Artificial neural networks in medical diagnosis," IJCSI International Journal of computer science, vol. 8, no. 2, pp. 150-154, 2011.

13. Y. Khan, P. H. Zope, and S. R. Suralkar, "Importance of artificial neural network in medical diagnosis disease like acute nephritis disease and heart disease", International journal of engineering science and innovative technology (IJESIT), vol. 2, no. 2. pp. 210-217, 2013.

14. Stango A. Spatial correlation of high density emg signals provides features robust to electrode number and shift in pattern recognition for myocontrol / A. Stango, F. Negro, D. Farina // Neural Systems and Rehabilitation Engineering. – IEEE Press. – 2015. – Vol. 23, N. 2. – P. 189-198.

15. M. Gheorghe, "A support vector machine approach for developing telemedicine solutions: medical diagnosis," *Network intelligence studies*, vol. III, no. 1(5), pp. 43-48, 2015.

16. Rossi M. Hybrid EMG classifier based on HMM and SVM for hand gesture recognition in prosthetics/ [M. Rossi](#), [S. Benatti](#); [E. Farella](#); [L. Benini](#)// [IEEE International Conference on Industrial Technology \(ICIT\)](#). – IEEE Press. – 2015. – P. 1700-1705.

17. Conradt J. Artificial prosthetic limbs Problems and solutions for connecting brains and robots. *Neuroscientific system theory*. – Munich, Germany : TUM. – 2015. – 39 p.

18. Subbotin S. The neuro-fuzzy network synthesis and simplification on precedents in problems of diagnosis and pattern recognition / S. Subbotin // *Optical Memory and Neural Networks* – 2013. – Vol. 22, № 2. – P. 97–103. DOI:10.3103/s1060992x13020082

19. Moskalenko V.V. Information-extreme algorithm of the system for recognition of objects on the terrain with optimization parameter feature extractor / V.V.Moskalenko, A.G. Korobov // *Radio Electronics, Computer Science, Control*. – Zaporizhzhya: Zaporizhzhya National Technical University, 2017. – № 2. – P. 61-69. DOI:[10.15588/1607-3274-2017-2-7](#).

20. Korobov, A. Moskalenko, V. Nahornyi, J. Zavgorodnya, O. Berest and V. Zhurba, "Parameters Optimization Method of the Information-Extreme Object Recognition System on the Terrain," 2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC), Kiev, 2018, pp. 1-5. doi:10.1109/SAIC.2018.8516771

21. Moskalenko, V., Moskalenko, A., Pimonenko, S., Korobov, A. Development of the method of features learning and training decision rules for the prediction of violation of service level agreement in a cloud-based environment [Text]. *Eastern-European Journal of Enterprise Technologies.*, 2017. – Vol. 5, No. 2 (89), pp. 26 – 33. DOI: 10.15587/1729-4061.2017.110073

22. Piotrowski, M. Napiorkowski, J. Napiorkowski, and

P. Rowinski, "Swarm Intelligence and Evolutionary Algorithms: Performance versus speed," *Information Sciences*, vol. 384, pp. 34-85, 2017. doi:[10.1016/j.ins.2016.12.028](https://doi.org/10.1016/j.ins.2016.12.028)

23. Dovbysh, A. S., Moskalenko, V. V., Rizhova, A. S. Learning decision making support system for control of nonstationary technological process/ *Journal of automation and information sciences*.— [New York: Begell House Inc.](#), 2016, v. 48. i. 6. pp. 39 – 48. DOI:10.1615/JAutomatInfScien.v48.i6.40

24. Dovbysh, A.S., Rudenko, M.S. Information-extreme learning algorithm for a system of recognition of morphological images in diagnosing oncological pathologies/ *Cybernetiks and Systems Analysis*, 2014, v. 50, No. 1, pp. 157-163. DOI:10.15587/1729-4061.2016.71930

25. M. Shamsin, N. Krilova, M. Bazhanova, V. Kazantsev, V. A. Makarov, and S. Lobov, "Supervised and unsupervised learning in processing myographic patterns," in *Journal of Physics: Conference Series*, Nov. 2018, vol. 1117, no. 1, doi: 10.1088/1742-6596/1117/1/012008.

26. A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, "Prosthetic Myoelectric Control Strategies: A Clinical Perspective," *Curr. Surg. Reports*, vol. 2, no. 3, 2014.

## Додаток А

Лістинг програми машинного навчання системи керування протезом руки

```
import numpy as np
import matplotlib
from matplotlib import pyplot as plt
from matplotlib import image as pimg
import copy
from sklearn import metrics
import scipy.spatial.distance as dst
import math
import pandas as pd
from typing import List, Dict, Sized
import os
import re
import glob
import time
import codecs, json
from IPython.display import clear_output
import numba as nb
from datetime import datetime as dt
```

```
class ArrayEncoder(json.JSONEncoder):
    def default(self, obj):
        if isinstance(obj, np.ndarray):
            return obj.tolist()
        # Let the base class default method raise the TypeError
        return json.JSONEncoder.default(self, obj)
```

Функції реалізації основних операторів процесу навчання:

```
@jit(nopython=True, cache=True, parallel=True, fastmath=True)
def heming(arrA, arrB):
    return np.bitwise_xor(arrA, arrB).sum()
```

```
@jit('i2[:,:](b1[:,], b1[:,,:])', nopython=True, cache=True, parallel=True, fastmath=True)
def center_reals(binearized_center, binary):
    result = np.zeros((binary.shape[0], binary.shape[1]), dtype=np.int16)
    for i in range(binary.shape[0]):
        for j in range(binary.shape[1]):
            result[i][j] = heming(binearized_center, binary[i][j])
    return result
```

```
@jit('i8(i8, i2[:])', nopython=True, cache=False, parallel=True, fastmath=True)
def k13(rds, h_cntr_rls):
    return np.sum(h_cntr_rls <= rds)
```

```

@jit('i8(i8, i2[:])', nopython=True, cache=False, parallel=True, fastmath=True)
def k24(rds, h_cntr_rls):
return np.sum(h_cntr_rls > rds)

```

```

@jit(nopython=True, cache=False, fastmath=True)
def kulbak(N,K2,K3):
K23 = K2 + K3;
return np.log2((2*N - K23 + 10.0**(-8)) / (K23 + 10.0**(-8))) * (N - K23) / N

```

```

@jit(nopython=True, cache=False, fastmath=True, parallel=True)
def kulbak_mean(radius, h_cntr_rls, z, N):
klbk_sum = 0.0
K2 = k24(radius, h_cntr_rls[z])
sumK3 = 0.0
K3 = 0
for i in prange(h_cntr_rls.shape[0]):
if(i != z):
K3 = k13(radius, h_cntr_rls[i])
klbk_sum += kulbak(N, K2, K3)
sumK3 += K3
nbrcnt = (h_cntr_rls.shape[0] - 1)

return (K2/N, sumK3/nbrcnt/N, klbk_sum/nbrcnt)

```

```

@jit(nopython=True, cache=True, parallel=True, fastmath=True)
def binearize (image, top, bottom):
return np.logical_and(np.greater_equal(top, image), np.greater_equal(image, bottom))

```

```

@jit(nb.types.UniTuple(nb.float64,4)(nb.int16[:,:], nb.int64, nb.int64, nb.int64),nopython=True, cache=False, fastmath=True)
def m_radius_opt(h_cntr_rls, r_max, z, N):
radius = 0
result = 0
best_kulbak = 0.0
best_alfa = 0.0
best_beta = 0.0
while radius < r_max :

alfa, beta, klbk = kulbak_mean(radius, h_cntr_rls, z, N)

if klbk >= best_kulbak:
best_kulbak = klbk
result = radius
best_alfa = alfa
best_beta = beta
radius += 1
return result, best_kulbak, best_alfa, best_beta

```

```

@jit(nb.float64[:,:](nb.int16[:,:], nb.int64, nb.int64, nb.int64),nopython=True, cache=False, fastmath=True)

```

```

def stat_radius_opt(h_cntr_rls, r_max, z, N):
    radius_stats = np.zeros((r_max,4))
    radius = 0
    result = 0
    best_kulbak = 0.0
    best_alfa = 0.0
    best_beta = 0.0
    s = 0
    while radius < r_max :
        radius_stats[s,0] = radius

        alfa, beta, klbk = kulbak_mean(radius, h_cntr_rls, z, N)

        radius_stats[s,1] = klbk
        radius_stats[s,2] = alfa
        radius_stats[s,3] = beta
        s += 1
        radius += 1
    return radius_stats

```

Формування вхідної навчальної матриці:

```
gstcds = [1, 2, 3, 6, 7, 8, 9]
```

```
df = pd.read_hdf(os.path.join(filtered_data_folder, "emg_gestures-03-repeats_short-2018-05-11-11-15-21-403_filtered.hdf5"))
```

```
N = 20
```

```
k = 10
```

```
norm = kulbak(N,0,0)
```

```
GestArrs = [df.drop(df[df.TRAJ_GT != gest].index)
```

```
                [df.columns[df.columns.str.contains('EMG')]] for gest in gstcds]
```

```
LearnArrs = np.array([gest.iloc[:N*k].to_numpy().reshape((N, k, -1)).transpose(0, 2, 1).re-
shape(N,-1) for gest in GestArrs])
```

```
TestArrs = np.array([gest.iloc[N*k:2*N*k].to_numpy().reshape((N, k, -1)).transpose(0, 2,
1).reshape(N,-1) for gest in GestArrs])
```

Алгоритми навчання:

Базовий:

```
LearnGestArrs = LearnArrs.copy()
```

```
gstcds = [1, 2, 3, 6, 7, 8, 9]
```

```
print(best_deltas)
```

```
delta = 20
```

```
print("Delta = " + str(delta))
```

```
slct = 0.5
```

```
M = LearnGestArrs.shape[0]
```

```
best_klbks = [0]*M
```

```

all_rds_stats = [[]]*M
best_bin_centers = [0]*M
best_radiuses = [0]*M

centers = LearnGestArrs.mean(axis=1)
# Цикл що по чергово формує правила для класів
for z in range(0,M):
    print("Start for class " + str(z))

    best_kulbak = 0.
    klbk = 0.
    best_radius = 0.

    best_bin_cntr_dst_sum = 0

    bst_bin_cntr = None
    n = 0

    binary = np.apply_along_axis(lambda x: binearize(x, centers[z] + delta, centers[z] -
delta), 2, LearnGestArrs)

    bin_centers = binary.astype(np.int16).mean(axis=1) > slct

    best_bin_centers[z] = bin_centers[z]

    bs_otrh_cntr_dst = np.apply_along_axis(lambda x: heming(bin_centers[z], x), 1,
np.delete(bin_centers, z, axis=0))

    bin_cntr_dst_sum = bs_otrh_cntr_dst.sum()

    bs_center_all_reals_dst = center_reals(bin_centers[z], binary)

    radius_stats = stat_radius_opt(bs_center_all_reals_dst, bs_otrh_cntr_dst.min(), z, N);
    all_rds_stats[z] = radius_stats
    del bs_center_all_reals_dst

    if (radius_stats.shape[0] > 0):
        rds_klbks = radius_stats[:,1]
        rdss = radius_stats[:,0]
        best_klbks[z] = rds_klbks.max()
        best_radiuses[z] = rdss[np.where(rds_klbks == best_klbks[z])[0][0]]

    print("Radius = " + str(best_radiuses[z]) + " Kulbak=" + str(best_klbks[z]))

    del binary

filename = "simple_" + str(M) + "_classes_k=" + str(k) + "_N=" + str(N) + "_" +
dt.now().strftime("%Y_%b_%d_%H-%M") + ".json"
with open(filename, 'w') as outfile:
    json.dump({'deltas': delta,
        'radiuses': best_radiuses,

```



```

    'best_klbks': best_klbks,
    'centers': best_bin_centers,
    'stats': all_rds_stats
}, outfile, cls=ArrayEncoder)

```

Базовий алгоритм із оптимізацією квантування за рівнем:

```

minimum = LearnArrs.min()
maximum = LearnArrs.max()

quant_field = maximum - minimum

quant_step = quant_field / 30

LearnGestArrs = LearnArrs.copy()

gstcds = [1, 2, 3, 6, 7, 8, 9]

quant_stats = []

level_params = []

delta = 20
print("Delta = " + str(delta))
slct = 0.5

M = LearnGestArrs.shape[0]

best_klbks = [0]*M
all_rds_stats = [[]]*M
best_bin_centers = [0]*M
best_radiuses = [0]*M
best_quants = [0]*M
level_kulbaks = [0]*M
quant_x = []
quant_y = []
workspace = []
# Цикл що по чергово формує правила для класів
quant = minimum

while quant < maximum:
    quant_x.append(quant)
    quant_y.append([0.]*M)
    workspace.append([[0.,0.]]*M)

# level_kulbaks = [0.]*M

quant_stats.append([quant])

LearnGestArrs[LearnGestArrs < quant] = 0
centers = LearnGestArrs.mean(axis=1)

```

```

for z in range(0,M):
    print("Start for class " + str(z))

    best_bin_cntr_dst_sum = 0

    bst_bin_cntr = None
    n = 0

    binary = np.apply_along_axis(lambda x: binearize(x, centers[z] + delta, centers[z] -
delta), 2, LearnGestArns)

    #     for f, slct in enumerate(C):
    bin_centers = binary.astype(np.int16).mean(axis=1) > slct

    best_bin_centers[z] = bin_centers[z]

    bs_otrh_cntr_dst = np.apply_along_axis(lambda x: heming(bin_centers[z], x), 1,
np.delete(bin_centers, z, axis=0))

    #     Розділяємо бінарні центри в просторі

    bin_cntr_dst_sum = bs_otrh_cntr_dst.sum()

    bs_center_all_reals_dst = center_reals(bin_centers[z], binary)
    del binary

    radius_stats = stat_radius_opt(bs_center_all_reals_dst, bs_otrh_cntr_dst.min(), z,
N);
    del bs_center_all_reals_dst

    if (radius_stats.shape[0] > 0):
        rds_klbks = radius_stats[:,1]
        klbk = rds_klbks.max()
        best_idx = np.where(rds_klbks == klbk)[0][0]
        radius = radius_stats[best_idx,0]
        alfa = radius_stats[best_idx,2]
        beta = radius_stats[best_idx,3]

        workspace[-1][z] = [alfa, beta]
        quant_y[-1][z] = klbk

    if klbk > best_klbks[z] or (klbk == best_klbks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
        if(klbk > 0.0 and klbk > best_klbks[z]):
            print("Selection = "+str(slct)+" Radius = "+ str(radius) +" Kulbak=" + str(klbk))
            print("Delta optimization " + str(delta))
            print(bs_otrh_cntr_dst)
            best_radiuses[z] = radius
            best_klbks[z] = klbk
            best_bin_centers[z] = bin_centers[z]
            best_bin_cntr_dst_sum = bin_cntr_dst_sum

```

```

        best_quants[z] = quant
        all_rds_stats[z] = radius_stats
        if klbk > level_kulbaks[z] or (klbk == level_kulbaks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
            level_kulbaks[z] = klbk
#         print("Radius = "+ str(best_radiuses[z]) + " Kulbak=" + str(best_klbks[z]))

#     best_bin_cntr_dst_sum = bin_cntr_dst_sum
#         best_selections[z] = slct
#         best_quants[z] = quant

quant_stats[-1].extend(level_kulbaks)
quant += quant_step

filename = "simple_quant_" + str(M) + "_classes_k=" + str(k) + "_N=" + str(N) + "_" +
dt.now().strftime("%Y_%b_%d_%H-%M") + ".json"
with open(filename, 'w') as outfile:
    json.dump({'deltas': delta,
              'radiuses': best_radiuses,
              'best_klbks': best_klbks,
              'centers': best_bin_centers,
              'quant_stats': quant_stats,
              'rds_stats': all_rds_stats,
              'x': quant_x,
              'y': quant_y,
              'wrkspc':workspace
              }, outfile, cls=ArrayEncoder)

```

Алгоритм із паралельною оптимізацією контрольних допусків:

```
LearnGestArrs = LearnArrs.copy()
```

```
gstcds = [1, 2, 3, 6, 7, 8, 9]
```

```
delta_constraint = 90
print(delta_constraint)
```

```
slct = 0.5
min_delta = 0
```

```
M = LearnGestArrs.shape[0]
best_quants = [0]*M
best_deltas = [0]*M
best_klbks = [0]*M
best_selections = [0]*M
best_bin_centers = [0]*M
best_radiuses = [0]*M
```

```
delta_x = []
delta_y = []
```

```

workspace = []

centers = LearnGestArrs.mean(axis=1)

for z in range(0,M):
    print("Start for class " + str(z))

    best_kulbak = 0.
    klbk = 0.
    best_radius = 0.
    best_delta = 0.
    bst_slct = 0.
    best_bin_cntr_dst_sum = 0
    delta = min_delta
    bst_bin_cntr = None

    delta_x.append([])
    delta_y.append([])
    workspace.append([])
    while delta < delta_constraint:

        binary = np.apply_along_axis(lambda x: binearize(x, centers[z] + delta, centers[z] -
delta), 2, LearnGestArrs)

        delta_x[z].append(delta)
        delta_y[z].append(0.)
        workspace[z].append([0.,0.])

        bin_centers = binary.astype(np.int16).mean(axis=1) > slct

        bs_othr_cntr_dst = np.apply_along_axis(lambda x: heming(bin_centers[z], x), 1,
np.delete(bin_centers, z, axis=0))

#     Розділяємо бінарні центри в просторі
    if np.any(bs_othr_cntr_dst == 0):
        delta += 0.5
        continue

    bin_cntr_dst_sum = bs_othr_cntr_dst.sum()

    bs_center_all_reals_dst = center_reals(bin_centers[z], binary)

    radius, klbk, alfa, beta = m_radius_opt(bs_center_all_reals_dst,
bs_othr_cntr_dst.min(), z, N);

    del bs_center_all_reals_dst
#     Логування
    workspace[z][-1] = [alfa, beta]
    delta_y[z][-1] = klbk

```

```

    if klbk > best_klbks[z] or (klbk == best_klbks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
        if(klbk > 0.0 and klbk > best_klbks[z]):
            print("Selection = "+str(slct)+" Radius = "+ str(radius) +" Kulbak=" + str(klbk))
            print("Delta optimization " + str(delta))
            print(bs_otrh_cntr_dst)
            best_deltas[z] = delta
            best_radiuses[z] = radius
            best_klbks[z] = klbk
            best_bin_centers[z] = bin_centers[z]
            best_bin_cntr_dst_sum = bin_cntr_dst_sum
#             best_selections[z] = slct
#             best_quants[z] = quant
        del binary
        delta += 0.5
filename = "base_" + str(M) + "_classes_k=" + str(k) + "_N=" + str(N) + "_" +
dt.now().strftime("%Y_%b_%d_%H-%M") + ".json"
with open(filename, 'w') as outfile:
    json.dump({'deltas': best_deltas,
              'radiuses': best_radiuses,
              'best_klbks': best_klbks,
              'centers': best_bin_centers,
              'x':delta_x,
              'y':delta_y,
              'wrkspc':workspace
              }, outfile, cls=ArrayEncoder)

```

Паралельний алгоритм із оптимізацією за рівнем селекції:

```
LearnGestArrs = LearnArrs.copy()
```

```
gstcds = [1, 2, 3, 6, 7, 8, 9]
```

```

delta_constraint = 90
print(delta_constraint)
C = [
    0.5, 0.6, 0.7, 0.8, 0.9
]

```

```
min_delta = 0
```

```

M = LearnGestArrs.shape[0]
best_quants = [0]*M
best_deltas = [0]*M
best_klbks = [0]*M
best_selections = [0]*M
best_bin_centers = [0]*M
best_radiuses = [0]*M

```

```
delta_x = []
```

```

delta_y = []
workspace = []
print(delta_x)
centers = LearnGestArrs.mean(axis=1)
# Цикл що по чергово формує правила для класів
for z in range(0,M):
    print("Start for class " + str(z))

    best_kulbak = 0.
    klbk = 0.
    best_radius = 0.
    best_delta = 0.
    bst_slct = 0.
    best_bin_cntr_dst_sum = 0
    delta = min_delta
    bst_bin_cntr = None

    delta_x.append([])
    delta_y.append([])
    workspace.append([])
    while delta < delta_constraint:

        binary = np.apply_along_axis(lambda x: binearize(x, centers[z] + delta, centers[z] -
delta), 2, LearnGestArrs)

        delta_x[z].append(delta)
        delta_y[z].append([0.] * len(C))
        workspace[z].append([[0.,0.]] * len(C))

        for f, slct in enumerate(C):
            bin_centers = binary.astype(np.int16).mean(axis=1) > slct

            bs_othr_cntr_dst = np.apply_along_axis(lambda x: heming(bin_centers[z], x), 1,
np.delete(bin_centers, z, axis=0))

            # Розділяємо бінарні центри в просторі
            if np.any(bs_othr_cntr_dst == 0):
                continue

            bin_cntr_dst_sum = bs_othr_cntr_dst.sum()

            bs_center_all_reals_dst = center_reals(bin_centers[z], binary)

            radius, klbk, alfa, beta = m_radius_opt(bs_center_all_reals_dst,
bs_othr_cntr_dst.min(), z, N);

            del bs_center_all_reals_dst
# Логування
workspace[z][-1][f] = [alfa, beta]
delta_y[z][-1][f] = klbk

```

```

    if klbk > best_klbks[z] or (klbk == best_klbks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
        if(klbk > 0.0 and klbk > best_klbks[z]):
            print("Selection = "+str(slct)+" Radius = "+ str(radius) +" Kulbak=" + str(klbk))
            print("Delta optimization " + str(delta))
            print(bs_othr_cntr_dst)
            best_deltas[z] = delta
            best_radiuses[z] = radius
            best_klbks[z] = klbk
            best_bin_centers[z] = bin_centers[z]
            best_bin_cntr_dst_sum = bin_cntr_dst_sum
            best_selections[z] = slct

del binary
delta += 0.5
filename = "base_select_" + str(M) + "_classes_k=" + str(k) + "_N=" + str(N) + "_" +
dt.now().strftime("%Y_%b_%d_%H-%M") + ".json"
with open(filename, 'w') as outfile:
    json.dump({'deltas': best_deltas,
              'radiuses': best_radiuses,
              'best_klbks': best_klbks,
              'centers': best_bin_centers,
              'best_slcts': best_selections,
              'x':delta_x,
              'y':delta_y,
              'wrkspc':workspace
              }, outfile, cls=ArrayEncoder)

```

Паралельний алгоритм із квантуванням за рівнем:¶

```

minimum = LearnArrs.min()
maximum = LearnArrs.max()

quant_field = maximum - minimum

quant_step = quant_field / 10

LearnGestArrs = LearnArrs.copy()

gstcds = [1, 2, 3, 6, 7, 8, 9]

quant_stats = []

level_params = []

delta_constraint = 90
print(delta_constraint)

slct = 0.5
min_delta = 0

```

```

M = LearnGestArrs.shape[0]
best_quants = [0]*M
best_deltas = [0]*M
best_klbks = [0]*M
qrnt_klbks = [0]*M
best_selections = [0]*M
best_bin_centers = [0]*M
best_radiuses = [0]*M

best_delta_x = [[]] * M
best_delta_y = [[]] * M
best_workspace = [[]] * M
quant = minimum
while quant < maximum:
    level_kulbaks = [0.]*M
    delta_x = []
    delta_y = []
    workspace = []
    quant_stats.append([quant])

    LearnGestArrs[LearnGestArrs < quant] = 0
    centers = LearnGestArrs.mean(axis=1)
    # Цикл що почергово формує правила для класів
    for z in range(0,LearnGestArrs.shape[0]):
        print("Start for class " + str(z))

        delta_x.append([])
        delta_y.append([])
        workspace.append([])

        best_kulbak = 0.2
        klbk = 0.
        best_radius = 0.
        best_delta = 0.
        bst_slct = 0.
        best_bin_cntr_dst_sum = 0
        delta = min_delta
        bst_bin_cntr = None
        n = 0
        while delta <= delta_constraint:

            binary = np.apply_along_axis(lambda x: binearize(x, centers[z] + delta, cen-
ters[z] - delta), 2, LearnGestArrs)

            delta_x[z].append(delta)
            delta_y[z].append(0.)
            workspace[z].append([0.,0.])

            bin_centers = binary.astype(np.int16).mean(axis=1) > slct

```



```

    bs_othr_cntr_dst = np.apply_along_axis(lambda x: heming(bin_centers[z], x), 1,
np.delete(bin_centers, z, axis=0))

#    Розділяємо бінарні центри в просторі
    if np.any(bs_othr_cntr_dst == 0):
        delta += 0.5
        continue

    bin_cntr_dst_sum = bs_othr_cntr_dst.sum()

    bs_center_all_reals_dst = center_reals(bin_centers[z], binary)

    radius, klbk, alfa, beta = m_radius_opt(bs_center_all_reals_dst,
bs_othr_cntr_dst.min(), z, N);

    del bs_center_all_reals_dst
#    Логування
    workspace[z][-1] = [alfa, beta]
    delta_y[z][-1] = klbk

    if klbk > best_klbks[z] or (klbk == best_klbks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
        if(klbk > 0.0 and klbk > best_klbks[z]):
            print("Selection = "+str(slct)+" Radius = "+ str(radius) +" Kulbak=" + str(klbk))
            print("Delta optimization " + str(delta))
            print(bs_othr_cntr_dst)
            best_deltas[z] = delta
            best_radiuses[z] = radius
            if alfa <= 0.5 and beta <= 0.5:
                best_klbks[z] = klbk
                best_bin_centers[z] = bin_centers[z]
                best_bin_cntr_dst_sum = bin_cntr_dst_sum
                best_selections[z] = slct
                best_quants[z] = quant
            if klbk > level_kulbaks[z] or (klbk == level_kulbaks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
                level_kulbaks[z] = klbk
            del binary
            delta += 0.5

        if(best_quants[z] == quant):
            best_delta_x[z] = delta_x[z]
            best_delta_y[z] = delta_y[z]
            best_workspace[z] = workspace[z]
        quant_stats[-1].extend(level_kulbaks)
        quant += quant_step

filename = "level_" + str(M) + "_classes_k=" + str(k) + "_N=" + str(N) + "_" +
dt.now().strftime("%Y_%b_%d_%H-%M") + ".json"
with open(filename, 'w') as outfile:
    json.dump({'deltas': best_deltas,

```

```

'radiuses': best_radiuses,
'best_klbks': best_klbks,
'centers': best_bin_centers,
'best_quants': best_quants,
'x':best_delta_x,
'y':best_delta_y,
'wrkspc':best_workspace,
'quant_stats': quant_stats
}, outfile, cls=ArrayEncoder)

```

Паралельний алгоритм із квантуванням за рівнем та оптимізацією рівня селекції:

```
minimum = LearnArrs.min()
```

```
maximum = LearnArrs.max()
```

```
quant_field = maximum - minimum
```

```
quant_step = quant_field / 10
```

```
quant_stats = []
```

```
LearnGestArrs = LearnArrs.copy()
```

```
gstcds = [1, 2, 3, 6, 7, 8, 9]
```

```
wrkspc_sqntl = []
```

```
level_params = []
```

```
hierarchy = []
```

```
delta_constraint = 90
```

```
print(delta_constraint)
```

```
C = [
```

```
    0.5, 0.6, 0.7, 0.8
```

```
]
```

```
# slct = 0.6
```

```
min_delta = 0
```

```
M = LearnGestArrs.shape[0]
```

```
best_quants = [0.]*M
```

```
best_deltas = [0.]*M
```

```
best_klbks = [0.]*M
```

```
best_selections = [0.]*M
```

```
best_bin_centers = [0]*M
```

```
best_radiuses = [0]*M
```

```
best_delta_x = [[]] * M
```

```
best_delta_y = [[]] * M
```

```
best_workspace = [[]] * M
```

```
quant = minimum
```

```
while quant < maximum:
```

```
    level_kulbaks = [0.]*M
```

```
    delta_x = []
```

```
    delta_y = []
```

```
    workspace = []
```

```

quant_stats.append([quant])

LearnGestArrs[LearnGestArrs < quant] = 0
centers = LearnGestArrs.mean(axis=1)
# Цикл що по чергово формує правила для класів
for z in range(0, LearnGestArrs.shape[0]):
    print("Start for class " + str(z))

    delta_x.append([])
    delta_y.append([])
    workspace.append([])

    best_kulbak = 0.
    klbk = 0.
    best_radius = 0.
    best_delta = 0.
    bst_slct = 0.
    best_bin_cntr_dst_sum = 0
    delta = min_delta
    bst_bin_cntr = None
    n = 0
    while delta <= delta_constraint:

        binary = np.apply_along_axis(lambda x: binearize(x, centers[z] + delta, cen-
ters[z] - delta), 2, LearnGestArrs)

        delta_x[z].append([delta] * len(C))
        delta_y[z].append([[0., 0.]] * len(C))
        workspace[z].append([[0., 0.]] * len(C))

        for f, slct in enumerate(C):
            bin_centers = binary.astype(np.int16).mean(axis=1) > slct

            bs_othr_cntr_dst = np.apply_along_axis(lambda x: heming(bin_centers[z], x),
1, np.delete(bin_centers, z, axis=0))

            # Розділяємо бінарні центри в просторі
            if np.any(bs_othr_cntr_dst == 0):
                continue

            bin_cntr_dst_sum = bs_othr_cntr_dst.sum()

            bs_center_all_reals_dst = center_reals(bin_centers[z], binary)

            radius, klbk, alfa, beta = m_radius_opt(bs_center_all_reals_dst,
bs_othr_cntr_dst.min(), z, N);

            del bs_center_all_reals_dst
        # Логування
        workspace[z][n][f] = [alfa, beta]
        delta_y[z][n][f] = [klbk, best_kulbak]

```

```

# Зберігай окремо максимальні значення критеріїв для кожного рівня квантування
if klbk > best_klbks[z] or (klbk == best_klbks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
    if(klbk > 0.0 and klbk > best_klbks[z]):
        print("Selection = "+str(slct)+" Radius = "+ str(radius) +" Kulbak=" +
str(klbk))
        print("Delta optimization " + str(delta))
        print(bs_othr_cntr_dst)
        best_deltas[z] = delta
        best_radiuses[z] = radius
        best_klbks[z] = klbk
        best_bin_centers[z] = bin_centers[z]
        best_bin_cntr_dst_sum = bin_cntr_dst_sum
        best_selections[z] = slct
        best_quants[z] = quant
    if klbk > level_kulbaks[z] or (klbk == level_kulbaks[z] and bin_cntr_dst_sum >
best_bin_cntr_dst_sum):
        level_kulbaks[z] = klbk
    del binary
    n += 1
    delta += 0.5
    if(best_quants[z] == quant):
        best_delta_x[z] = delta_x[z]
        best_delta_y[z] = delta_y[z]
        best_workspace[z] = workspace[z]
    quant_stats[-1].extend(level_kulbaks)
    quant += quant_step

filename = "level_select_" + str(M) + "_classes_k=" + str(k) + "_N=" + str(N) + "_" +
dt.now().strftime("%Y_%b_%d_%H-%M") + ".json"
with open(filename, 'w') as outfile:
    json.dump({'deltas': best_deltas,
        'radiuses': best_radiuses,
        'best_klbks': best_klbks,
        'centers': best_bin_centers,
        'best_selections': best_selections,
        'best_quants': best_quants,
        'x':best_delta_x,
        'y':best_delta_y,
        'wrkspc':best_workspace,
        'quant_stats': quant_stats
    }, outfile, cls=ArrayEncoder)

```